Screening Fist

James Engleback

June 23, 2022

Contents

1 Abstract

Screening Fist is a screening operation used to train a machine learning model to predict the likelihood of a given enzyme binding to a given small molecule from sequence. The model is pre-trained on a large and general dataset of protein and small molecule binding pairs scraped from several online sources before retraining on the screening data. The screening data is generated in-house with a high-throughput, custom developed Cytochrome P450-small molecule binding assay, with which five P450 BM3 mutants were each screened for binding against 980 drug-like molecules.

The retrained model can be used for virtual screening of new enzyme sequences against a specific target molecule, or to query a single sequence against many prospective small molecule binding partners. It can also be used to design subsequent rounds of screening based on the expected information gain of an experimental design.

2 Introduction

2.1 Engineering Problem and Context

2.2 Similar Work

2.3 Technologies Used

2.3.1 Transfer Learning

Transfer learning is a phenomenon where a model trained on one task, can be re-trained on a different domain-related task sample-efficiently compared to an untrained model. For example an object detection model trained from from images of vehicles can transfer efficiently to identification of cell phenotypes from microscopy images. This is because some learned features generalize well enough to be useful in other tasks, reducing the number is samples required to reach a baseline performance level.

In the domain of protein sequence-based machine learning, thoroughly pre-trained models are available for generating a neural embedding of a given protein that can improve sample efficiency in downstream learning tasks. Generally, these models are large attention-based models trained *unsupervised* on a large corpus of protein sequences, like the *TrEMBL* collection of *Uniprot*. In this case, unsupervised training often entails reconstruction of a distorted or masked protein sequence and is run on hardware far beyond the budget of this project.

2.3.2 The TAPE Benchmark

Tasks Assessing Protein Embeddings **tape2019** is a benchmark for comparing numerical representations of protein sequence (learned or otherwise) on a set of biological learning tasks from different domains of protein science. It currently contains five tasks:

1. **Secondary Structure Prediction Task:**
2. **Structural Contact Prediction Task:**
3. **Remote Homology Detection:**
4. **Fluorescent Protein Landscape Prediction:**
5. **Protein Stability Landscape Prediction:**

Tasks 4 and 5 are most applicable to protein engineering, since they involve metric prediction from a set of largely similar protein sequences. The leader boards for performance on these two tasks as of 5 Jun 2022 are:

Fluorescence:

Stability:

Table 1: Fluorescence TAPE benchmark leader boards

Ranking	Model	Spearman's rhoSize
1.	Transformer	0.68
2.	LSTM	0.67
2.	Unirep	0.67
4.	Bepler	0.33
5.	ResNet	0.21
6.	One Hot	0.14

Table 2: Stability TAPE benchmark leader boards

Ranking	Model	Spearman's rhoSize
1.	Transformer	0.73
1.	Unirep	0.73
1.	ResNet	0.73
4.	LSTM	0.69
5.	Bepler	0.64
6.	One Hot	0.19

| Ranking | Model | Spearman's rho | |:-:|:-:|:-:|

Facebook AI Research - Evolutionary-Scale Modelling

2.4 Overview of This Work

3 Methods and Development

3.1 Assay Development

In order to generate a P450 BM3-specific dataset on which a model could be re-trained to make binding likelihood predictions on drug-like molecules, a high throughput P450 binding detection assay was developed. The assay is based on traditional UV-Visible light spectroscopy-based techniques for detection of P450-ligand binding, miniaturized into a 384 well format. It relies heavily on automation and a throughput of 980 compounds per day is demonstrated.

3.1.1 Aim

The initial aims of this development work were:

- Develop a high throughput P450-ligand binding assay based on established biophysical characterisation techniques.
- Develop necessary software for design and analysis of each assay.

- Compare the precision and accuracy of the assay to existing techniques.

3.1.2 Basis: UV-Visible Spectroscopy for Monitoring Cytochrome P450-Ligand Binding

The assay is based on a technique for quantifying P450-ligand interactions based on UV-visible photospectroscopy. The technique consists of the purified Cytochrome P450 heme domain in question in a neutral buffer at around 5-10 μ M in a optically clear cuvette. Since only the heme-containing domain of the P450 is used, no chemical reactions are expected to take place which removes time-sensitivity from the assay.

The UV-visible light absorbance of the sample is typically measured for all wavelengths between 200 and 800 nm, which for a P450 without a ligand bound in the active site should show a large and defined absorbance peak at around 420 nm.

After an initial absorbance measurement of the ligind-free P450, the compound of interest can be titrated into the sample. On binding to the ligand, the absorbance profile of the P450 changes such that the absorbance at 420 nm (A_{420}) decreases and absorbance at 390 nm (A_{390}) increases.

The change in A_{420} and A_{390} in response to change in ligand concentration can be quantified and used to derive metrics that indicate affinity between the ligand and P450 using Michaelis-Menten kinetics models.

The original Michaelis-Menten model of enzyme kinetics states:

$$v = V_{max} \frac{[S]}{[S] + K_M}$$

where v is the reaction velocity - the rate of an enzymatic reaction. V_{max} is the maximum possible v for this particular enzyme-substrate pair, $[S]$ is the concentration of the substrate and K_M is the $[S]$ at which $v = V_{max}$.

V_{max} and K_M are useful metrics for quantifying the binding interaction between enzyme and substrate, where low K_M indicates a tight binding interaction and a high V_{max} indicates a large magnitude of response.

Important assumptions in the Michaelis-Menten model of kinetics are:

1. The concentration of enzyme is $< K_d$
2. The rate of reaction is directly proportional to the concentration of the substituents
3. The reaction is at chemical equilibrium at the time of measurement
4. The interaction is reversible

A variant of this model is applied to Cytochrome P450 photospectroscopy assays. v is substituted for $\Delta A_{390} - \Delta A_{420}$ - the magnitude of the P450 response and K_M is substituted for K_d - the dissociation constant between the enzyme and ligand. This yields the formula:

$$\Delta A_{390} - \Delta A_{420} = V_{max} \frac{[S]}{[S] + K_d}$$

or

$$Response = V_{max} \frac{[S]}{[S] + K_d}$$

This style of assay was miniaturized into a 384-well format for the purpose of this project. The 384-well format permits high throughput screening of compounds for binding with a given P450 provided it is sufficiently stable to last the duration of the experiment without degrading and interfering with measurement.

3.2 Assay Protocol

This is a assay protocol for detecting binding interactions between a Cytochrome P450 and multiple small molecule compounds. The assay has demonstrated scale to a library of 980 compounds and five P450 mutants and with some small adjustments could be improved in scale and precision.

It works in 384 well microplates and uses a microplate reader to capture absorbance profiles from 220-800 nm wavelengths, from which a pattern associated with a P450-small molecule binding interactions can be detected and quantified.

It was designed for profiling and modelling the effect of mutations on a P450's substrate binding preferences. This was tested with five P450 mutants against 980 drug-like compounds. It requires purified P450 protein which limits the rate of data generation, though can scale to more compounds.

3.2.1 Requirements

Essential:

- **Hardware:**
    - **Microplate reader:** Able to read absorbance for all wavelengths between 220 and 800 nm. Used here: *BMG ClarioStar and FluoStar* microplate readers.

  - **Labcyte Echo [500 550]:** Acoustic liquid handlers for precise compound dispensing.
  - **Consumables:** In absence of a high precision liquid handling machine, serial dilution of compounds would probably be fine.
  - **Enzyme:** A purified Cytochrome P450 - used here were mutants of P450 BM3 at 800 μM. Note that BM3 is fairly stable at room temperature which facilitates queuing large batches of plates to the plate reader. You could run the assay at a low temperature if you use a solvent other than DMSO, which freezes at 19ºC, which interferes with measurement.
  - **Compound Library:** A large number of compounds in solvent (e.g. DMSO) in a microplate format. Used here was a 980 compound library, dissolved at 10 mM in DMSO in 96 well format.
  - **Buffer:** must be optically clear the protein must be stable in it. Must not contain potential ligands. Used here was 100 mM Potassium Phosphate at pH 7.0 - chosen based on traditional wisdom.
  - **384 well microplates - clear bottom:** Assay plates with at least 30 μl working volume. Some treated surfaces may be more suitable for unstable proteins. Ideally have minimal absorbance in the 390-420 nm region, but this can be corrected for with controls. Used here: *ThermoFisher Nunc 384 Well Plates* - **384 well *Labcyte Echo* source plates:** for dispensing compounds to assay plates. Used here were the *Low Dead Volume (LDV)* variety, which have a working volumne of 2.5-12.5 μl, which limits compound waste compared to the standard *Echo* plates (12-65 μl i think?).

Optional:

- **Hardware:**
  - **Bulk liquid dispensing:** can be far more accurate than a multichannel pipette when dispensing protein or buffer into wells. During development, both a *ThermodropMulti* peristaltic dispenser and a *Hamilton Star* liquid handling robot. Both work well.
  - **Microplate reader plate loader:** Autoloading plates into the reader increases throughput capacity significantly. I used a **BMG ClarioStar** plate reader with a stacker module.
- **Consumables:**
  - **BSA:** in assay buffer may have a stabilizing effect on the enzyme - which would improve time stability and reduce errors. Time stability is important for scalability.

- **384 well Labcyte Echo DMSO Trays:** for control for DMSO concentration in assay wells by topping up each assay plate to a fixed concentration. Around 5% is ok with BM3.

3.2.2 Procedure

Summary:

1. **Design *Echo* picklists**
2. **Dispense compounds into *Echo* source plates**
3. **Dispense compounds from *Echo* source plates to empty assay plates**
4. **Stopping point**
5. **Thaw purified P450 and make stock of 10 μM in a neutral buffer, enough for 15.36+ ml per plate (40 μl per well)**
6. **Dispense the diluted protein into the assay plates, centrifuge etc**
7. **Capture UV-Visible light absorbance data between 220 and 800 nm from plates in a microplate reader at room temperature within 3 hours**
8. **Data analysis**

1. **Design *Echo* picklists:**
   - An *Echo* can accept a `csv` file with column headers: `Transfer Volume`, `Volume`, `Destination Well`, `Source Well` and optionally: `Destination Plate Name` and `Source Plate Name`. The Volume must be in nano litres and a multiple of 2.5 and the Source and Destination wells must be in the format `[A-Z][0-9]+` and exist in the plate layout specified to the *Echo* client at runtime.
   - The picklist(s) can be generated in a spreadsheet exported to `.csv` or programmatically. Documentation for the `python` tools used are [documented here.](picklists.md)
2. **Dispense compounds into *Echo* source plates** This can be done with a multichannel pipette, and requires one tip per compound. If the total volume of each compound required is greater than 60 μl then a standard polypropylene *Echo* plate should be used, otherwise a low dead volume plate may be economical If not, or for valuable compounds, Low Dead Volume *Echo* may plates should be used. These have a working volume of 2.5-12.5 μl, outside of which the *Echo* will refuse to dispense. You may need to dispense the same compounds into multiple source wells and the picklists must be designed accordingly.
3. **Dispense compounds from *Echo* source plates to empty assay plates**

   (a) Transfer the picklist `.csv` to the *Echo* host computer.

   (b) Launch the *Echo Plate Reformat* client there:

   (c) Create New Protocol

   (d) Select Custom Region Definitions

   (e) `File` > `Import Regions` and select your picklist `.csv`

   (f) **Optional:** Specify the log output in `Options`, simulate with `Run` > `Simulate`

   (g) Save and run, optionally simulating the run first. Multiple copies of a set of destination plates can be specified if the source plates contain sufficient compound volume.

4. **Stopping point:** Length of pause depends on rate of DMSO evaporation from destination/assay plates and the stability of the compounds at the plate storage conditions. Plates stored in a stack should limit evaporation rate to an extent, though specialised lids for *Echo* plates that limit DMSO evaporation are available. Up to 24 hours seems ok.

5. **Thaw purified P450, make stock of 10 μ M in a neutral buffer, enough for 15.36+ ml per plate (40 μl per well)** I heard that thawing fast limits ice crystal formation, which could destroy some protein. Optionally, in a microcetrifuge, pre-cooled to 4C, spin the protein at 14,000 rpm and carefully transferr to fresh tubes to remove unfolded protein.

   (a) Measure the stock concentration of the protein in a UV-Vis spectrometer by taking an absorbance trace from 200-800 nm, diluted in the destination buffer. There should be a peak at 420 nm, the height of which can be used to calculate the protein concentration with the following equation:

   $$[P450] = ae$$

   where a is absorbance and e is the extiction coefficient - 95 for P450 BM3 heme domain. Use the measured stock concentration of P450 to create a working stock of around 10 μ M. 10μ M was chosen because it yeilds a reasonably strong signal when in plates. Varying the protein concentration doesn't have a big effect on measurements, so err towards using more.

   (b) Dilute in neutral buffer to the target working concentration. Filtration through a 22 um syringe filter can remove some precipitates. Vacuum filtration can work too but in practice, the protein can pick up binding partners from the filtration equipment contaminants, which can ruin downstream measurements.

6. **Dispense the diluted protein into the assay plates, centrifuge**

   An electric multichannel pipette works but accuracy is more limited than with automated dispensing. 38 μl of protein working stock needs to be dispensed into each well, which brings the total well volume to 40 μl in cases where the volume of compounds in DMSO in each well is 2 ul. If the volume of DMSO in destination wells is not a constant 2 ul, then default to 38 μl of the protein stock. The variation in total volume can be corrected for in compound concentration calculations, though the path lenght will vary which affects precision.

   Better than that is a precise bulk liquid handling device. I used a *ThermodropMulti* for a while which was fast and accurate. Occasionally a nozzle would become blocked either with DMSO ice or precipitates, though the protein still dispensed into the correct well. Blockages can be cleared by disassembling the pump head, coupling a syringe of water to the nozzle and flushing.

   It may be necessary to dispense some control plates, with everything but protein. This is useful to correct for the intrinsic absorbance of the plate and buffer, as well as the compounds themselves which sometimes have absorbance at the measurement wavelengths. A control set of plates for every protein screen may be unnecessary and expensive. One good one should be ok.

   Centrifuge the plates for 2-5 mins at around 2000 rpm to push the well contents to the bottom. This step can also ensure that meniscuses are flat and uniform and remove bubbles. If possible, centrifuge at room temperature to avoid DMSO ice formation.

7. **Capture UV-Visible light absorbance data between 220 and 800 nm from plates in a microplate reader at room temperature within 3 hours:**

   The protein is fairly stable over the course of 3 hours. On a BMG platereader, measurements take about 15 minutes per plate including the data transfer from device to host machine. Using an automated plate loader is recommended, for example a BMG Stacker module. In that case, put an empty or waste plate on the top of the stack to limit evaporation from the top assay plate. The BMG ClarioStar can be instructed not to read the last plate.

   The stacker occasionally jams due to a solenoid error, which can be due to a misaligned stack of plates. It is advisable to un-stack and restack the plates using the stacker to check for this kind of issue prior to measurement.

8. **Data analysis overview** More info [here](data.md).

   (a) Export the plate measurement data to a workable format, like

‘.csv‘. In the BMG Mars software, the operation is simple but on all host machines I’ve tried it on have been unreasonably slow to open the data files prior to export.

(b) Index the files to their experiments. I used a ‘config.yml‘ file to track this.

(c) **Analysis**

i. Match compounds to plate well data.

ii. Match the *Echo* exceptions report to wells to find the actual compound volume in each well.

iii. From each trace, subtract its own absorbance A_{800} at 800 nm. This accounts for baseline drift which can be caused by light scattering from precipitates.

iv. If correcting for compound absorbance with control plates, then subtract the absorbance of each test well from each control well. If the actual compound volumes of the test and control don’t match up, it can be an issue if the compound interferes with the absorbance in the 390-420 nm region. If the compound absorbance changes predictably then it can be interpolated.

v. Curves can be smoothed with Gaussian smoothing using `scipy.ndimage.gaussian_filter` if necessary. Sources of a jagged curve can be precipitates, which can interfere with downstream analysis.

vi. At this point, changes in the P450 absorbance trace can be identified. Common categories of trace are:

- Clean absorbance trace, no shift.
- Clean absorbance trace, peak shift from 420 to 390 nm.
- Clean absorbance trace, peak shift from 420 to 424 nm.
- Compound interference in absorbance trace.
- Light scattering in absorbance trace.

vii. For clean traces with a peak shift from 420 to 390 or 424 nm, biding response can be calculated using the $|\Delta A390| - |\Delta A_{420}|$ or $|\Delta A420| - |\Delta A_{420}|$ for each compound concentration. With a set of concentration-response data points, the binding dissociation constant K_d can be calculated using the Michaelis-Menten equation for enzyme kinetics:

$$Response = \frac{V_{max} \times [S]}{K_d + [S]}$$

Table 3: BM3 mutants used in screening

ID	**Mutations**	**PDB**
WT		1BU7
A82F	A82F	4KEW
DM	A82F/F87V	4KEY
1YQO	T268A	1YQO
1YQP	T268N	1YQP

$$Response = |\Delta A_{390}| - |\Delta A_{420}|$$

Where $[S]$ is a given substrate concentration and V_{max} is the maximum response magnitude possible from the P450 being tested from this compound. The metrics K_d and V_{max} can be derived by fitting $|\Delta A390| - |\Delta A_{420}| = \frac{V_{max} \times [S]}{K_d + [S]}$ can be fit to the P450 substrate concentration-response data points using a curve fit algorithm like `scipy.optimize.curve_fit`.

Useful additional metrics for each compound are R^2 score of the curve fit, a data quality metric.

An ideal end output of this analysis as a table of compounds, P450s and a qualification or quantification of their binding interactions.

Documentation on how I implemented this is [here](data.md)

3.3 Enzyme Production

3.3.1 Summary

This page contains the methods for producing the enzymes used in this screening program. The enzymes are variants of the Cytochrome P450 BM3:

The page shows the method used to create the mutant BM3 expression plasmid DNA, expression of the mutants in *E. coli* and their purification.

3.3.2 Aims

- Create expression plasmids containing the target mutants from an in-house starting point - `bm3-wt.gb`.
- Sequence the plasmids to confirm they carry the mutations
- Express the mutants in *E. coli* using those plasmids.
- Purify the mutant protein from the *E. coli* harvest.

3.3.3 DNA

3.3.4 Starting Material

An heirloom BM3 Wild-type (heme domain) expression plasmid, [bm3-wt.gb](), was inherited and used as the basis for DNA work in this project. The plasmid is a **?** pET15(?) expression vector where the BM3 gene has a 6xHis purification tag at the N-terminus, flanked by a T7 promoter and terminator which leads to high yields in strains of *E. coli* containing the T7 RNA polymerase. The plasmid also encodes ampicillin resistance and a ? replication origin which leads to a low copy number.

3.3.5 Primer design and Acquisition

Mutations were introduced to the wild-type sequence via Polymerase Chain Reaction (PCR)-based site-directed mutagenesis. Two methods were considered for this task based on commercially available kits, where each imposes different constraints on primer design. Efforts were made to automate primer design as far as possible with scalability in mind.

The PCR kits used were:

1. *New England Biolabs (NEB) Q5 mutagenesis kit* - which requires that primers facilitate cloning of a linear DNA strand from the circular template plasmid and mutation payloads are carried in the tail of one primer. The kit includes a cocktail of the DNAse *DPN1*, which disassembles template plasmid methylated in *E. coli* and a kinase and ligase that work to join the ends of the linear DNA into a circular plasmid. The reaction is restricted to one payload.
2. *Agilent Quickchange mutagenesis kit* - which requires a pair of overlapping primers that carry the mutation payload in the mid-section. This cloning method produces circular DNA carrying the targeted changes. It has the advantage of allowing multiple payloads carried by multiple primer sets.

Two important considerations based on the template sequence are:

1. Adenine-Thymine (AT) richness of the template sequence. Compared to cytosine and guanine (C and G), A and T bind to their complimentary bases weakly. This results in weak primer binding to the template sequence, measurable by a low primer *melting temperature* T_m. To compensate, primers must be longer than they otherwise would be for a sequence richer in CG, which increases their cost and their chance of self-binding. The template sequence used here is AT-rich - at x%
2. Repetitions and palindromic regions of the template sequence. If the sequence surrounding a mutation target area contains these fea-

tures, then the likelihood of *mis-priming* by binding to an off-target sequence area is high, so too is the likelihood of a non-functional, self-binding primer.

3.3.6 PCR and Work Up

3.3.7 Sequencing

Purified plasmid DNA ostensibly conataining the target mutations, having been harvested and purified from DH5a *E. coli* cells, was shipped to *Eurofins Genomics* for sequencing using their *TubeSeq* service, which uses a variant of Sanger Sequencing. Sequencing primers for this matched the T7 promoter and terminator and provided coverage of the targetted region.

3.4 Expression

Having been sequenced and confirmed to carry the target mutations, the mutant plasmids were used to produce the mutant protein *en masse* via a *BL21 DE3 E. coli* strain, which contains a T7 RNA polymerase under the control of a *lac* promoter.

3.4.1 Materials

- Expression plasmid encoding mutant P450 BM3
- *BL21 DE3 E. coli* - NEB. This domesticated *E.coli* strain is shipped in a transformation buffer.
- Auto-induction *Terrific Broth* (TB) media, which contains glucose and a lactose analog. The lactose analog triggers expression of T7 RNA-polymerase in *BL21 DE3 E. coli* and the subsequent expression of the target protein between the T7 promoter and terminator regions. The glucose inhibits this until it is consumed by the cells, which allows them to multiply to sufficient numbers before diverting energy to production of the target protein.
- Ampicillin - the antibiotic for which resistance is encoded in the target plasmid, ensuring that all cells in the growth media contain this resistance. Assuming no ampicillin-resistant contaminants, all cells should be *BL21 DE3 E. coli* containing the target plasmid.
- Δ Amino-Levulnic acid (Δ-ALA) - a precursor to heme, ensuring heme availability for the large amount of BM3.

3.4.2 Method

3.4.3 Purification

3.5 Screening

3.6 Model Design and Construction

$$P_{binding} = fn(sequence, smiles)$$

This page describes the deep learning model constructed for this project. The model is designed to estimate the likelihood of a binding interaction between a given Cytochrome P450 sequence and a ligand SMILES code. The intended end uses of the model are:

1. Virtually screening sequences for potential activity with a given compound. 2. Optimally design an enzyme-ligand screening experiment.

3.6.1 Approach: Recommender Systems

Abstractly, the problem of predicting the binding likelihood between a one of m proteins and one of n small molecules can be likened to filling empty the values of an $n \times m$ matrix, where rows and columns refer to proteins and small molecules and values are the probability of a binding interaction between the two:

$$\begin{array}{ccccc} & compound_i & compound_{i+1} & ... & compound_{i+n} \\ sequence_i & P_{binding_{i,j}} & & & \\ sequence_{i+1} & & \ddots & & \\ \vdots & & & \ddots & \\ sequence_{i+m} & & & & \ddots \end{array}$$

Some $P_{binding}$ values are known, which in the perspective of $n \times m$ possible values where n and m approach infinity, coverage is sparse.

This type of problem has been addressed in recommender systems, which in the context of streaming services translates to a matrix of n users and m paces of content. Known values are likes and engagement metrics and are similarly sparse, and blanks can be filled with the probability of a successful recommendation.

Machine learning models can be trained to predict the unknown values based on a numerical representation of the user and content. The prediction can be cast as a classification problem. To overcome the lack of negative data points, presumed negative data can be generated by sampling a random user: content pair, which should be treated with caution.

In this work, a machine learning model classifies pairs of protein sequence and small molecules as binding or not. Negative samples are generated by randomly sampling a sequence and small molecule, which given the vastness of sequence and chemical space may be reasonable in a large number of cases, though this assumption is treated with caution. A Binary Cross Entropy loss function is employed here where true positives and synthesized negatives are weighted evenly.

3.6.2 Data

Small molecules are represented as SMILES codes in the dataset, which are parsed using `rdkit` and then into 2048 bit fingerprint vectors using the `RdkitFingerprint`. Molecular fingerprints are generated by hashing functions based on an input molecule such that similar molecules are assigned similar fingerprints, which makes them useful in featurizing small molecules for machine learning tasks. The fingerprints are represented as a $b \times 2048$ tensor where b is batch size.

Sequences are represented in the dataset as strings where each character c_i is a single letter amino acid code:

$$c_i \subset ACDEFGHIKLMNPQRSTVWY$$

Roughly, characters are encoded as tensors of the integers that index their position in the list $ACDEFGHIKLMNPQRSTVWY$, with extra positions to represent null values, start of frame and end of frame characters.

3.6.3 Pre-Training Data

3.6.4 Training Data

3.6.5 Model Architecture

The model aims to predict:

$$P_{binding} = fn(sequence, smiles)$$

Where fn is a model that takes an input of a protein $sequence$ and a prospective ligands' SMILES code - $smiles$ and outputs $P_{binding}$ - an estimate of the probability that the given $sequence$ and $smiles$ bind to one another.

Given their prior success in chemical and protein sequence learning, a neural network model was chosen to build fn. The network can be split into three parts:

- **Sequence Embedding:** For a given protein $sequence$, output a tensor encoding a neural embedding $z_{sequence}$.

- **Chemical Embedding:** For a given chemical $smiles$ encoding, outputs an embedding z_{smiles}.
- **Prediction Head:** For the embeddings z_{smiles} and $z_{sequence}$, output a prediction $P_{binding}$.

3.6.6 Sequence Embedding

Although this model is the smallest of the ESM collection, on the single *NVIDIA Quadro RTX 6000* used it still occupied most of the 24 GB of available memory and most of the processing capability, which lead to long training times and difficulty in training more than one model in parallel on the same machine.

This could be remedied, however since in the complete `o3f.csv` dataset and the screening dataset there are 2947 unique sequences, so it was economical to pre-compute the embeddings and save them to disk. This resulted in a roughly $4x$ speedup in training time and massively reduced the memory requirements, allowing several models to be trained in parallel on a single GPU. This also saved costs significantly.

3.6.7 Chemical Embedding

As mentioned, chemical SMILES were hashed into chemical fingerprints using the `rdkit` `RDKFingerprint` method as a means of representation, yielding a 2048-bit vector for each compound. The vectors were converted to tensors and served as an input to a residual neural network that output an embedding that would later be used to form a combined representation of both compound and sequence for binding likelihood prediction.

3.6.8 Prediction Head

The combined sequence and compound embeddings served as input to the prediction head, which output a single number that indicated a binding likelihood prediction for the two inputs.

Both residual neural networks and transformers were compared as prediction head architectures. Each consisted of 2-6 stacked layers of either residual or transformer layers with a fixed hidden layer size for convenience of automated assembly. The final layer in both cases was a single linear layer with a single output and a sigmoid function to output a number between 0 and 1 representing binding probability.

3.6.9 Pre-Training, Training and Evaluation

Training was done in two stages, each with a performance evaluation, during which several models with varying architectures and hyper-parameters

Table 4: BM3 mutants used in screening

Item	**Specifications**	**Number**	**Size**
CPU	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	8	
RAM	?	N/A	30 GB
Disk	?	1	630 GB
GPU	NVIDIA Quadro RTX 6000	1	20 GB VRAM

were trained and compared. All training was done on a *Linode* `g1-gpu-rtx6000-1` which cost $1.50 per hour and was equipped with the following hardware specifications:

1. **Pre-Training:** This was done with the larger, more general 'o3f' dataset, which was randomly split into training and validation partitions, the latter of which was used sparingly to avoid model bias. Pre-training lasted up to 64 epochs with a batch size up to 64. For each sample, a random sequence and SMILES pair were sampled as a presumed negative sample. The loss function used was binary cross entropy used with an Adam (Adaptive momentum) optimizer. Loss was tracked live using the *Weights and Biases* API which was useful to evaluate models as they trained and terminate them where necessary. Model weights were saved in each epoch and after training the model was evaluated for precision and accuracy on a subset of the training data. The metrics gathered were:

   (a) Mean binary cross entropy loss over evaluation.

   (b) Mean precision

   (c) A confusion matrix

   (d) A receiver operator curve (ROC)

   (e) A precision recall curve

   (f) A detection error trade-off (DET) curve

2. **Training:** This was done with the manually annotated screening dataset. An issue with the data was the class imbalance in that there were very few positive examples relative to negative. This was addressed by using *Synthetic Minority Oversampling* (SMOTE) whereby the rarer positive data were re-sampled until they number that of the negative data. The total size of the re-sampled data was 6666 points, which were then split 3:1 into training and validation sets of size 4999 and 1667 respectively. A model pre-trained on the larger `o3f` dataset was

re-trained on this set and evaluated for performance in the same manner as with the `o3f` data, visualised in the following section.

3.6.10 Evaluation

4 Results

4.1 Data Analysis

4.2 Model Training

4.3 Model Application

5 Discussion and Future Work