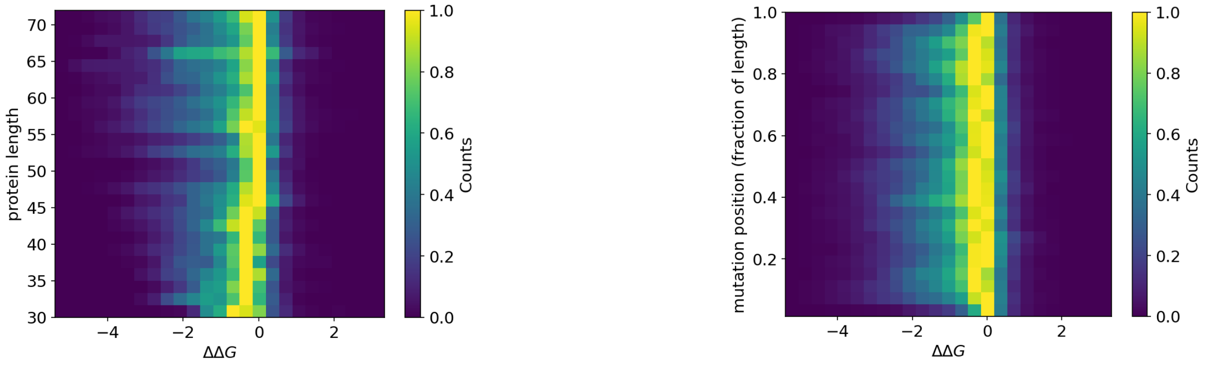
Prediction of single point protein mutation stability changes with the Mega-scale dataset.

Jeremy Ocampo

1 Problem analysis

Understanding the mechanisms in protein stability is important, since even a single amino acid mutation can be the cause of a disease. The effect of a mutation on a protein is usually assessed by the difference of the Gibbs free energy between the mutated and wild-type form protein, denoted $\Delta\Delta$G (denoted `score` in the dataset). Protein mutation experiments which measure $\Delta\Delta$G, are expensive and time-consuming, therefore predictive methods using simulation or machine learning have been developed. There are a few simulation methods, including the computation of force fields in molecular dynamics [4], but these methods are often computationally expensive for large datasets. Therefore, machine learning methods have been developed, including Graph neural networks which use atomic and bond information of the protein [6], dense neural networks which take as input the protein ID and mutation type [2], and anti-symmetric neural networks [3] which can make better predictions for reverse mutations. Here we will be using Long Short Term Memory (LSTM) networks [5], which are known to work well for sequence data.

Exploratory data analysis. We should explore the dataset first in order to find which variables have a high predictive power of $\Delta\Delta$G. There are 310 different wildtype proteins (`pdbid`) which isn't much, this means it would be harder for machine learning models to generalize to "unseen" protein structures. But there are 340k data points which should be enough for a neural network to perform well given we design it carefully. We also make a number of plots with descriptions, in Figure 1 we show the dependance of $\Delta\Delta$G with protein length and mutation position. Figure 2 shows the $\Delta\Delta$G dependance on the wildtype, and Figure 3 shows the dependance on the amino acid mutation.



(a) $\Delta\Delta$G vs protein length

(b) $\Delta\Delta$G vs mutation position

Figure 1: 2D histograms of $\Delta\Delta$G vs protein length (number of amino acids in `aa_seq`) or mutation position (as a fraction of the protein length). For fair comparison at each length, we've normalized each row of the histogram by the max. It seems like there is not much correlation with length, only slightly less energy for lower lengths. Presumably, this is because there are less electrostatic interactions for shorter proteins, but because they can fold in all sorts of ways, there is a spread in $\Delta\Delta$G. There's not much correlation with mutation position either, except at the ends of the protein which have mostly no change in $\Delta\Delta$G, presumably because there are less atoms and bonds around the ends of the protein.

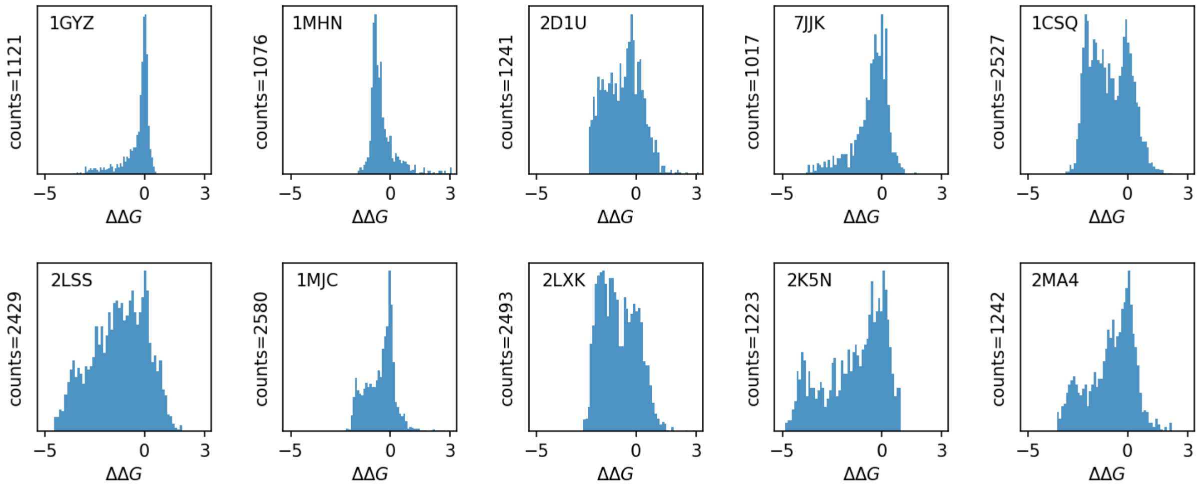


Figure 2: ΔΔG histograms, for 10 wildtypes (`pdbid`). This clearly shows that ΔΔG is dependant on the wildtype which makes sense since it should depend on the structure.

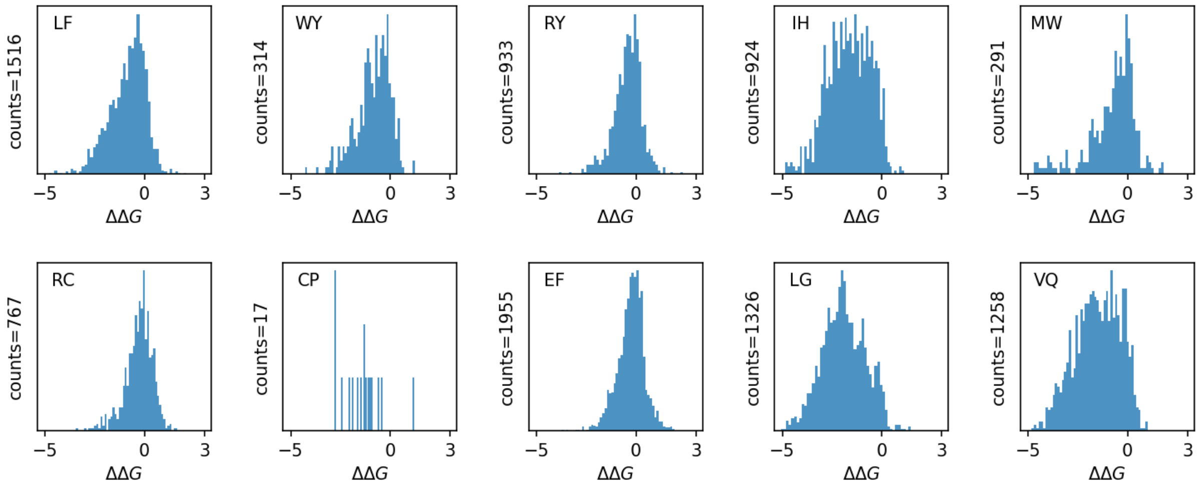


Figure 3: ΔΔG histograms, for 10 different mutation types (wildtype amino acid → mutant amino acid). The distributions seem to be slightly different for each mutation since the energy should depend on the amino acid substitution.

2 Method explanation

Data. The inputs to our model should be the protein sequence itself and the mutation since these provide information for predicting ΔΔG. To test that the models can generalize to unseen proteins, the training and test are split such that they contain completely different protein wild types (or `pdbid`). We will also use the same test set when predicting on the reverse mutations which have energy change $-\Delta\Delta G$. We've chosen a training:test split of 0.8:0.2, which is enough to train the neural network while being able to test for robustness. There are a few ways to encode the amino acids, for simplicity we will be using one-hot encodings, which is a size 20 vector of zeros (for 20 amino acids) and a 1 at the index of the amino acid, see Figure 4. To represent the wild type to mutant amino acid transition, we also include a -1 at the index of the wild type amino acid, which is an anti-symmetric representation.

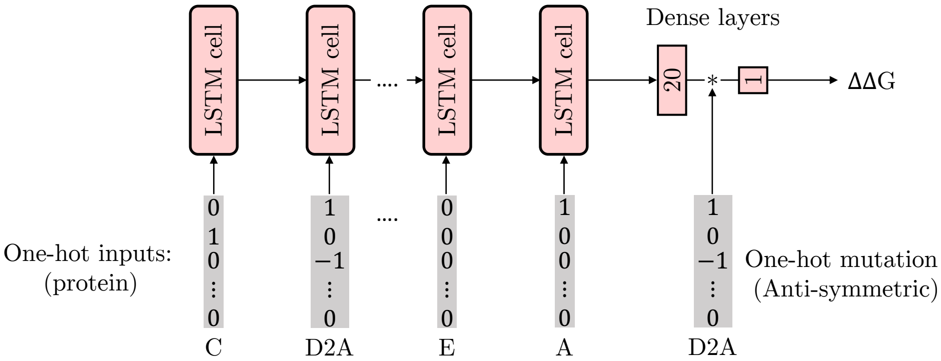


Figure 4: LSTM model used for predicintg ΔΔG. The inputs are one-hot encodings of the amino acid sequence, with a -1 inserted for the wild type amino acid. This makes the one-hot mutation anti-symmetric, i.e the swapping operation S on the one-hot mutation x gives $Sx = -x$.

Model. We will be using Long Short Term Memory (LSTM) neural networks, which have been proven to work well at encoding and making predictions on sequence data (see [1] for an overview). A LSTM can remember information over a wide range of intervals throughout the sequence, this is important since the protein structure (and therefore ΔΔG) depends on short and long range electrostatic interactions.

To start off, we should use simple models to minimize the number of assumptions, thereby improving generalizability. We will be using 1 recurrent LSTM cell from which the final output is applied onto 2 Dense layers which results in the final predicted ΔΔG, see Figure 4. In total, there are 50k parameters in the model. To aid with anti-symmetry, we do an elementwise multiplication of the mutant amino acid with the output of the first dense layer (with no bias). We may also use anti-symmetric activation functions (e.g. tanh) and no bias on the LSTM cells to further aid anti-symmetry, but for some reason, this leads to a bug which trains the models a lot slower, so we ended up not doing that.

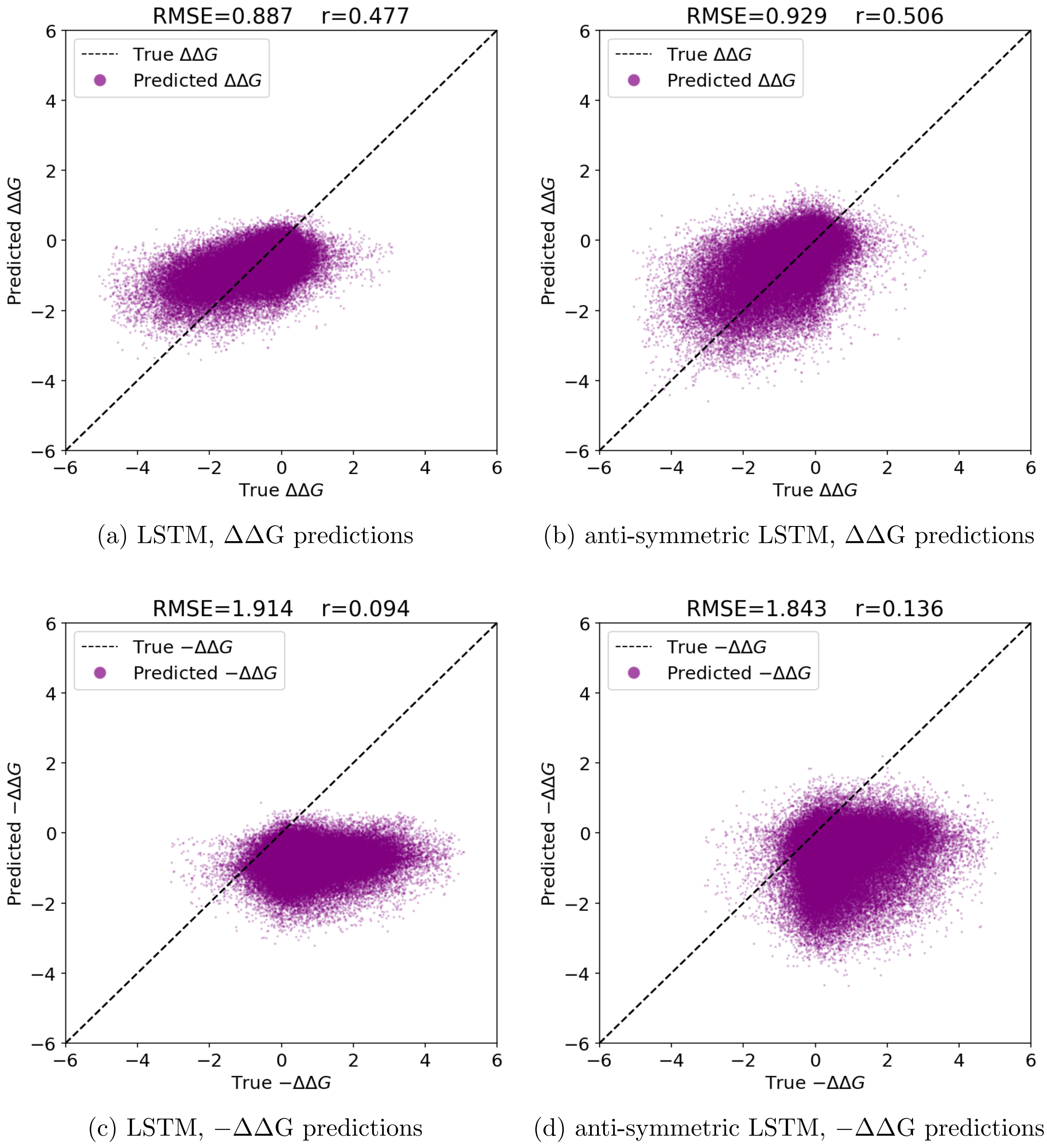
3 Experiment description

To correctly test our model's generalizability to unseen protein structures, the training and test datasets contain entirely different protein structures (`pdbid`). As a further generalizability test, we also predict the energy change for the test set when the wildtype (W) and mutant (M) amino acid is reversed which should give ΔΔG(W→M)=−ΔΔG(M→W). We test the model with and without the elementwise multiplication in Figure 4, called LSTM and anti-symmetric LSTM, both trained separately.

A commonly used metric to measured the performance of models for predicting ΔΔG is the root mean square error (RMSE) in combination with the pearson correlation coefficient, r. Together they provide a better metric to evaluate the performance for a wide range of ΔΔG.

4 Result Analysis

Figure 5a-5b shows the results for the (non)anti-symmetric models. We get a (RMSE, r) of (0.89, 0.48) and (0.93, 0.51). For comparison, the graph-NN from [6] has (1.23, 0.61) but trained on a smaller dataset, so our model isn't that bad! But for predicting the reverse energies, Figure 5c-5d shows a (RMSE, r) of (1.9, 0.1) and (1.8, 0.14), the anti-symmetric model does slightly better but they both seem to be pretty bad at generalizing to reverse energies, which is most likely due to the non-anti-symmetric LSTM cells. In comparison, [6] has (1.27,0.48) for predicting reverse energies.



(a) LSTM, ΔΔG predictions

(b) anti-symmetric LSTM, ΔΔG predictions

(c) LSTM, −ΔΔG predictions

(d) anti-symmetric LSTM, −ΔΔG predictions

Figure 5: True vs predicted ±ΔΔG using the (non-)anti-symmetric LSTM models.

References

[1] Long short-term memory. https://en.wikipedia.org/wiki/Long_short-term_memory.

[2] O. Alvarez-Machancoses, E. J. De Andres-Galiana, J. L. Fernandez-Martinez, and A. Kloczkowski. Robust Prediction of Single and Multiple Point Protein Mutations Stability Changes. *Biomolecules*, 10(1), 2019. URL https://doi.org/10.3390/biom10010067.

[3] S Benevenuta, C Pancotti, P Fariselli, G Birolo, and T Sanavia. An antisymmetric neural network to predict free energy changes in protein variants. *Journal of Physics D: Applied Physics*, 54(24), 2021. URL https://dx.doi.org/10.1088/1361-6463/abedfb.

[4] B. R. Brooks, C. L. Brooks, A. D. 3rd, Mackerell, L. Jr, Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, and M. Karplus. CHARMM: the biomolecular simulation program. Journal of computational chemistry. *Journal of computational chemistry*, 30(10), 2009.

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

[6] Shuyu Wang, Hongzhou Tang, Peng Shan, and Lei Zuo. ProS-GNN: Predicting effects of mutations on protein stability using graph neural networks. *bioRxiv*, 2021. doi: 10.1101/2021.10.25.465658. URL https://www.biorxiv.org/content/early/2021/10/26/2021.10.25.465658.