CS230

Generating Human Antibodies Using Language Models

Mira Partha mpartha@stanford.edu
Felipe Calero Forero fcalero@stanford.edu
Aaron Behr abehr@stanford.edu

1 Motivation

As the current pandemic has demonstrated, it is vital to be able to design synthetic antibodies very quickly. This process is costly and time-consuming in vitro, but can be significantly expedited with in silico techniques. Antibodies (and proteins in general) can be modelled as chains, or sequences, of amino acid residues. Each sequence encodes a unique three-dimensional structure that determines the antibody's ability to bind to target antigens, like SARS-CoV-2. Therefore, to rapidly respond to a pandemic, we need an effective way to generate antibody sequences that are promising binding candidates.

Recent advances in NLP have produced novel neural architectures particularly well-suited to modeling sequential information. This concept of *language modeling*, wherein a network learns probability distributions over text-like sequences, enables powerful embedded representations capable of exploiting complex positional information within protein structure. We surmise, therefore, that a language model may be a very effective method for generating promising candidate antibody sequences.

2 Problem Formulation

Our problem is composed of two distinct elements. We begin with a language model trained on protein sequences (note: antibodies are a specific class of proteins, but our language model is trained on general protein sequences). We expect that this model will tend to produce sequences encoding viable proteins, that is, proteins with structures that can properly fold. (This entails plausibly arranged polar and non-polar residues, no glycosylation motifs, etc.)

Amino acid residues that are far apart in sequence (2D) space may end up directly interacting - even forming hydrogen or disulfide bonds - in structural (3D) space, because of the complexity with which proteins fold into secondary and tertiary structures. Simple recurrent neural networks would not be adequate to handle the full-range dependencies, even with LSTMs/GRUs to address the problem of vanishing gradients. As noted by Rives et al., "Since self-attention explicitly constructs pairwise interactions between all positions in the sequence, the Transformer architecture directly represents residue-residue interactions." [7]. The complexity of protein structure demands the use of transformer architectures, which are epitomized by BERT [3]. Recently released protein language models, such as the one developed by Rives et al., provide an excellent starting point for our work.

The second part of our workflow is a regression model, which we train to predict the free energy change for binding to the SARS-CoV-2 receptor-binding domain, using sequence embeddings. We are not fine-tuning the language model explicitly for this downstream task; however, among the 10^40 possible antibodies considered for binding to SARS-CoV-2, we expect that viable proteins lie on a much lower-dimensional subspace - that is, most random sequence permutations do not encode foldable proteins. Strongly binding antibodies will lie on a further subspace of viable proteins; so, we use the metric of binding free energy as a proxy for protein sequence quality.

CS230: Deep Learning, Autumn 2020, Stanford University, CA.

Our pipeline is as follows:

1. The trained language model is used to generate embeddings for our labeled dataset of antibody sequences with calculated binding free energies.
2. The regression model is trained to predict free energy estimates given sequence embeddings as input.
3. We generate sets of candidate antibody sequences (mutants of the m396 antibody that binds to SARS-CoV-1), using three different methods:
   - Sequences are generated by allowing random mutations (sampling from a uniform distribution over amino acids) at a previously chosen set of residue locations.
   - Sequences are generated by mutating a previously chosen set of residue locations, this time sampling from non-uniform distributions determined by the previous (template) amino acid at each residue. This *substitution matrix*-based technique is the most commonly used method today for generating candidate antibodies for *in vitro* experimentation.
   - Sequences are generated by applying the trained language model to the task of *masked token prediction* - mutations are sampled from softmax distributions over possible amino acids at specified masked residues.
4. The trained language model is used to generate embeddings for three candidate sequence sets.
5. The trained regression model is used to predict binding free energies for the three candidate sequence sets, which are finally compared.

Though the task of masked token prediction has been explored, the task of predicting multiple (non-consecutive) masked tokens is less studied. This task is central, however, to our problem of generating antibody sequences. Below is an illustrative example:

> Consider the task of predicting the missing words in the following sentence: 'She wore a _____ because it was _____.' One would expect word pairs like (jacket, cold) and (skirt, hot) to co-occur; conversely, pairs like (jacket, hot) are unlikely. Marginal distributions can be computed for both missing words; but once one word is sampled, the distribution for the other missing word changes to the conditional distribution.

Therefore, the order in which we choose residues to mutate, and the information we provide (either masking or unmasking the other mutable residues) when calculating output distributions, may have a significant impact on the quality of generated sequences. Therefore, separate from comparing our model-generated sequences to those generated randomly or by use of a substitution matrix, we explore multiple methods for the model-based sequence generation.

3 Data

The language model we use has been pre-trained by Rives et al. on a high-diversity sparse dataset (UR50/S) of protein sequences from UniRef[7]. For our regression model, we use a set of approximately 85,000 candidates for a SARS-CoV-2-binding antibody, generated by Desautels et al.[2]. The sequences were obtained by mutating up to thirty-one specified residues (chosen by computational chemists for maximal proximity to the SARS-CoV-2 receptor binding domain) from the m396 antibody, an antibody known to effectively bind to and target SARS-CoV-1. The labeled dataset consists of amino acid sequences (concatenating heavy and light chains), along with predicted change in free energy for binding to the SARS-CoV-2 receptor-binding domain as evaluated using a variety of computational tools. For our purpose, we conduct supervised multitask learning to predict the FoldX free energy estimates for both the entire antibody-antigen complex, as well as the complex interface. In addition, the hundred best candidate sequences among these were identified by further evaluation using Rosetta and molecular dynamics simulations. Both we and Desautels et al. restrict ourselves to considering only point mutations (not insertions or deletions); therefore, all sequences are 458 residues long.

4 Methods, Model Architectures, and Hyperparameters

We use the BERT-based[3] transformer model developed and pre-trained by Rives et al.[7] to perform language modeling. The 34-layer model contains a total of 670M trainable parameters, and has a per-token embedding dimension of 1280. The perplexity of this model after pre-training was 8.54. The model was trained using the standard masked language modeling loss[3]:

$$\mathcal{L}_{\text{MLM}} = \mathbb{E}_{x \sim X} \, \mathbb{E}_M \sum_{i \in M} -\log p(x_i | x_{/M})$$

Figure 1: BERT's Masked Language Model Loss

For our regression task, we constructed a model with 474M trainable parameters. The model, which has a total input dimension of 585,782 (sequences are 458 tokens long, with an embedding dimension of 1279 per token), consists of 8 dense hidden layers of 800 units each, followed by 1 dense layer of 700 units, 1 dense layer of 400 units, and lastly, an output layer with 2 units (corresponding to the FoldX entire complex and interface-only free energies, respectively). The hidden layers have ReLU activation functions; and dropout (.2) and batch normalization (.99) were used for all hidden layers. The regression model was trained on a set of 80,000 sequences using mini-batch gradient descent with the Adam optimizer; weights were initialized using He initialization, and embeddings were normalized in each batch. Sets of 1,000 sequences were used for validation and test respectively. The model was trained using a standard mean-squared error loss function, shown below.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

Figure 2: Mean Squared Error Loss for the Regression Model

As mentioned in section 2, we also considered multiple methods for generating candidate sequences using our model. These different methods are described below:

- **Method 1**: We feed in the template sequence with all 31 mutable residues masked, compute softmax probability distributions over each residue, and then sample from these marginal distributions simultaneously. In this method, the sampled residues do not affect each other.
- **Method 2**: A number, n, is chosen uniformly at random from 1 to 31. We then repeat the following procedure n times: mask a residue chosen uniformly at random; compute the softmax distribution over possible amino acids at that residue; and lastly sample from the output distribution.
- **Method 3**: Begin with all 31 mutable residues masked. Proceed as follows: select uniformly at random among the remaining masked residues; compute the softmax distribution for the chosen residue; sample from the output distribution. This procedure is repeated without replacement, thirty-one times (until no masked tokens remain). This method is the most expensive and time-consuming.
- **Method 4**: Choose a subset size n uniformly at random from the range 1 to 31. Randomly select a particular residue subset of size n (chosen from $\binom{31}{n}$ possible subsets). Begin with this subset of residues masked. Then repeat the one-at-a-time unmasking (softmax and sampling) procedure from method 3, until no masked tokens remain.

5 Results

Below are our regression model's predicted FoldX entire-model binding free energy estimates, for: the m396 SARS-CoV-1 antibody (our starter template), the best antibody candidates generated by Desautels et al.[2], and sequence sets from our three generators - a random mutant generator, a substitution matrix-based generator (using the BLOSUM80 substitution matrix), and our language model-based generators, using the four different methods for multiple masked token prediction.

Sequence Generation Method	Mean	Std Dev	Min (best)	Max (worst)
SARS-CoV-1 antibody	-0.17			
Random mutation	$\mathbf{13.55}$	2.69	6.89	18.66
Substitution matrix	$\mathbf{6.19}$	2.48	2.36	12.88
Language model, method 1	8.89	1.11	7.10	10.42
Language model, method 2	3.66	3.67	-0.05	11.01
Language model, method 3	8.81	3.16	4.06	15.42
Language model, method 4	$\mathbf{2.89}$	2.09	-0.19	7.20
Best predictions from Desautels et al.	$\mathbf{-2.63}$	0.52	-4.15	-1.63

Overall, we found that the model-generated sequences significantly outperformed the substitution matrix-generated sequences, which significantly outperformed the randomly generated sequences, in terms of predicted FoldX binding free energies to SARS-CoV-2.

6 Analysis and Discussion

For our purpose, our regression model does not need to accurately predict objective binding energies; its only purpose is to accurately predict *relative* binding energies, in order to compare different generation methods. Therefore, we evaluated the model by comparing its generated sequences against the following:

- Randomly generated mutations
- Mutations generated using the BLOSUM80 substitution matrix
- The best hundred sequences from Desautels et al.[2]
- The m396 antibody template for SARS-CoV-1

We expect that randomly mutated sequences would be the worst, as random mutations (with a uniform distribution over amino acids) are most likely to render an antibody structurally infeasible (i.e. unable to fold); so such antibodies would not be able to effectively bind to the target antigen. Substitution matrix-based generation is the current standard for designing antibodies. It attempts to preserve antibody structure and function by making it more likely to replace residues with similar amino acids (polar replace polar, hydrophobic replace hydrophobic, etc.); this is less likely to disrupt the molecular conformation of the protein. However, no correlations between residues are accounted for; this is in contrast to reality, wherein certain motifs of residues are more or less likely to co-occur. Our transformer-based network *is*, in fact, able to model short- and long-range dependencies between residues. Therefore, we expect it to outperform both the random and the substitution matrix-based generators.

Our regression model successfully and reliably reproduces this distribution. In fact, given the relatively low overlap of predicted free energy distributions of the various generators, the model can perhaps even be used to classify how a given sequence was generated (via random mutation, substitution matrix, or by Desautels et al.'s method). Given that Desautels et al.'s method is fine-tuned on the task of minimizing binding free energy, we did not expect the language model's generated sequences to perform as well on this task; and unsurprisingly, this was borne out in the results. However, the model's generated sequences did perform substantially better than sequences generated by random mutations, as well as those generated using the substitution matrix (when using the better two methods for multiple masked token prediction). The best generated sequence from our language model has a similar energy to the m396 antibody for SARS-CoV-1, demonstrating that the language model accurately learned the sequential properties that characterize a viable human protein.

Overall, the superiority of the model-generated sequences validates the use of language models as a method for generating antibody sequences. Antibody function (i.e. binding efficacy) is determined by three-dimensional structure, which is uniquely encoded by amino acid sequence. The structural conformation of an antibody is a result of highly complex folding processes. Spatially proximal (and thus, significantly interacting) residues may appear far from each other in sequence; so the relationship between sequence and structure cannot be easily captured. Only an attention-based

model is capable of capturing interactions between residues that are both proximal and distant in sequence space.

We also find that the procedure of multiple (non-consecutive) masked token prediction, a problem which is not significantly addressed in current literature on language modeling (at least, to our knowledge), has tremendous impact on the quality of the generated sequences. We can understand these impacts by comparing the traits of the generation methods to those of their biological analogue (somatic hypermutation). (See Section 4 for descriptions of the individual model generation methods.)

Among the methods we tried, the first method performed most poorly, as expected. It is the most naive approach, and fails to capture inter-dependencies between the multiple masked tokens. Our third method performed similarly poorly. We hypothesize that this is because the method of generation was not sufficiently stochastic. Each residue was allowed to mutate exactly once, which is highly dissimilar from the naturally occurring hypermutation of human immune cells. With this method, so many residues are masked initially, that the first residue predictions are relatively uninformed; as a result, generated sequences may not sufficiently resemble the template sequence.

Our second method proved a substantial improvement on the first and third methods. It allows for multiple mutations of the same residue, which more closely resembles natural somatic hypermutation. However, because residues are predicted in isolation, this method still fails to address the issue of co-occurring residue motifs (as described in Section 2, in the (jacket, cold) vs. (skirt, hot) example). Our fourth method, which was designed to better handle residue correlations, addresses many of the issues from the previous three methods; and is therefore, unsurprisingly, the best performing generation method.

7 Final Thoughts and Lessons Learned

We encountered a number of issues while training and tuning our regression model, from which we have learned. One difficulty was that we used the per-token embeddings generated by the language model, instead of the mean-sequence embeddings. We initially decided to do so because we assumed the mean-sequence embeddings would lose too much information by collapsing the total sequence embedding dimension, and we surmised that this would make it difficult to train an accurate regression model. However, due to the size of the token embeddings and the length of our antibody sequences, this made the inputs to our regression model of dimension 585,782, which significantly limited the depth and the size of the remaining layers our model. This led to a dramatic collapse in dimension from the input layer to the first hidden layer, which we found to be sub-optimal. It also imposed serious constraints on our ability to tune hyperparameters for the model, due to the limited compute (time and memory) resources available. We were able to tune the regression model about five times, only allowing the models to train on a very small subset of our data before modifying them. As a result, our final regression model was far from ideal, though it worked reasonably well given our computational constraints. In retrospect, it would have been better to simply use the mean-sequence embeddings. We also should have chosen a larger AMI initially, with more GPUs to train a model with larger initial layers.

We also found significant differences between loss reported when using model.fit() versus model.evaluate(), while measuring our final training and test set loss. Our final training set error was 7.9, while our test set error was 38.2. While we expected test set error to be higher due to the model's limitations (as discussed earlier), we discovered that this extreme discrepancy is actually a result of a known issue with BatchNorm in Keras. Specifically, this problem arises due to differences in how Keras handles regularization (how the BatchNorm hyperparameters are applied) during training (model.fit()) versus inference (model.evaluate()). To circumvent this problem, after training the model, we set model.trainable equal to False, freezing all of the model's weights, and then used model.fit() to obtain the final loss. We duly note that the test set error fails to be a useful metric for evaluating model accuracy; therefore, we used other methods for the primary evaluation of our work.

8 Contributions

All team members made equal contributions to all parts of the project.

References

[1] Tileli Amimeur, Jeremy M. Shaver, Randal R. Ketchem, J. Alex Taylor, Rutilio H. Clark, Josh Smith, Danielle Van Citters, Christine C. Siska, Pauline Smidt, Megan Sprague, Bruce A. Kerwin, and Dean Pettit. Designing feature-controlled humanoid antibody discovery libraries using generative adversarial networks. *bioRxiv*, page 2020.04.12.024844, 01 2020.

[2] Thomas Desautels, Adam Zemla, Edmond Lau, Magdalena Franco, and Daniel Faissol. Rapid *in silico* design of antibodies targeting sars-cov-2 using machine learning and supercomputing. *bioRxiv*, page 2020.04.03.024885, 01 2020.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, 2019.

[4] Aleksandr Kovaltsuk, Jinwoo Leem, Sebastian Kelm, James Snowden, Charlotte M. Deane, and Konrad Krawczyk. Observed antibody space: A resource for data mining next-generation sequencing of antibody repertoires. *The Journal of Immunology*, 201(8):2502, 10 2018.

[5] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R. Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation. *bioRxiv*, page 2020.03.07.982272, 01 2020.

[6] Alexander Mirsky, Linda Kazandjian, and Maria Anisimova. Antibody-specific model of amino acid substitution for immunological inferences from alignments of antibody sequences. *Molecular Biology and Evolution*, 32(3):806–819, 12 2014.

[7] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, 2020.