Predicting Functional Protein Domains from the HT-recruit Dataset

Oussama Fadil
Stanford University
Stanford, CA
fadil@stanford.edu

Cynthia Hao
Stanford University
Stanford, CA
chao16@stanford.edu

Yuxi Ke
Stanford University
Stanford, CA
kyx@stanford.edu

Yiheng Li
Stanford University
Stanford, CA
yyhhli@stanford.edu

1. Abstract

Despite the importance of transcriptional activation and repression to cellular activity and the many studies that have focused on transcriptional effectors, we still do not have a comprehensive understanding of the mechanisms of transcriptional effectors or a way to predict transcriptional effects of different proteins based on sequence. In this work, we present two deep learning models that predict transcriptional repression scores for 80 aa-long protein fragments. Both models were trained on experimentally measured repression scores of 80 aa-long protein fragments of Pfam domains and 80 aa-long protein fragments tiling human nuclear proteins from the HT-recruit dataset [1].

The first model, a fully-connected network trained on UniRep sequence embeddings, achieved a mean absolute error of 0.380 on our test set [2]. The second model was built with a convolutional neural network architecture and performed slightly worse, with a mean absolute error of 0.452 on the test set.

Since the fully-connected model performed better on the test dataset, we applied it to annotate repression scores of 80 aa tiles over all human protein sequences and validated its performance. The proteins containing highly scoring tiles were enriched for DNA binding and transcription repressor activity. In addition, the proteins containing the top 3 tiles were transcription factors. We have developed a repression prediction model that can be used to systematically discover new natural sequences with repressor activity, clarify the function of unknown effector domains, and aid in the design of synthetic transcriptional repressors.

2. Introduction

Transcriptional repression and activation are processes central to protein expression and cellular function. These functions are carried out by proteins that bind to DNA and act to either block or recruit the transcription-initiation complex. Each such protein contains at least one DNA-binding domain and a separate functional effector domain to either repress or activate transcription. While DNA-binding domains are fairly well-characterized and well-understood due to their high sequence homology, we do not have a unified understanding of the physical and molecular mechanisms behind transcriptional effector function. Neither do we have a comprehensive list of all human proteins with silencing or activating function. High-throughput studies to measure eukaryotic activation and repression function have been performed in both yeast and human cells. These studies typically use empirical methods to track expression of a reporter gene with a known promoter in response to a pool of synthesized protein fragments. One such study utilizes the HT-recruit magnetic separation assay to simultaneously characterize repression and activation function in thousands of human nuclear protein domains, as well as protein fragments tiling known human repressors [1]. However, the capacity of even these high-throughput pooled screens is limited, and it is challenging to obtain accurate activity measurements for protein fragments that are not well-expressed or that behave differently in the context of the whole protein.

For this reason, we are developing a deep learning model to predict the repression ability of protein fragments based only on their amino acid sequences. Using a computational model to predict repression function would enable rapid and convenient screening of many protein sequences for repressor activity without needing to perform an experiment. This would not only allow the discovery and annotation of repressive domains in natural genomes, but also aid in the design of synthetic transcriptional repressors. In addition, the model could be continuously validated by design and experimental screening of a new protein fragment library based on its predictions. While similar computational models such as ADpred have been created to predict transcriptional activation ability or other protein functions, we are not aware of any models that predict transcriptional repression [3]. Our goal in this paper is to develop and train a model from the HT-recruit repression dataset to predict repression scores for 80 aa-long protein fragments. Using that model, we will also annotate all 80 aa-long tiles across the human proteome with repression scores to discover new po-

tential repression domains and proteins.

3. Methods

3.1. HT-recruit dataset

We used the HT-recruit dataset to train our model [1]. In this series of experiments, repression and activation measurements were obtained by expressing different libraries of synthetic 88 aa-long protein fragments with potential transcriptional effector activity, with each fragment fused to a known doxycycline-inducible DNA-binding domain. The expression of a magnetic cell surface marker reporter under a promoter driven by the known DNA-binding domain was measured for each mixed population of cells by magnetic cell separation, and each domain fragment was assigned a repression or activation score based on the number of cells with that fragment expressing the cell surface marker after 5 days of doxycycline induction.

This dataset consists of several main parts: repression measurements for 80 aa-long nuclear protein Pfam domains; activation measurements for the same set of Pfam domains; repression measurements for 80 aa-long protein fragments tiling known repressors; and repression scores for all possible single, double, and triple amino acid mutants of the well-characterized KRAB repressor protein. We utilized both the Pfam domain and the repressor tiling repression score dataset to build both of our models. While these datasets included repression scores at multiple time points after doxycycline induction of DNA-binding activity, we used only the day 5 time point, averaged across biological replicates, as our training labels for consistent scoring. Each 80 aa fragment was marked as either highly expressed or not well expressed, so we filtered out the fragments that were not well expressed and used only highly expressed protein fragments for our predictions. We also filtered out fragments with high baseline repression scores at day 0 (before doxycycline induction), indicating leaky DNA-binding or repression activity.

3.2. Data pre-processing and encoding

Our dataset was split 8:1:1 into separate training, validation, and test datasets. We kept all fragments from the same gene in a single sub-dataset to avoid data leakage between the test, training, and validation datasets. This resulted in 2792 80 aa-long Pfam domain fragments in the training set, 262 in the validation set, and 239 in the test set. We split the repressor tiling dataset so that there were 8613 repressor tiles in the training set, 790 in the validation set, and 1135 in the test set. Once we had our datasets split, we were able to use pre-processing methods to clean up our data before model training.

We pre-processed our data in two different ways. To compare different methods of encoding the amino acid sequence of each tile, we tried one-hot encoding as well as using UniRep embeddings for each protein sequence [2]. Each embedding method has unique benefits–the one-hot encoding method results in a lower dimensional vector that retains spatial information about the original sequence. However, the UniRep encoding method is able to capture biologically and physically relevant information from the sequence and is a constant length (1900) no matter what the input sequence length is.

Layer (type)	Output Shape	Param #
dense_8 (Dense)	(None, 900)	1710900
dropout_8 (Dropout)	(None, 900)	0
dense_9 (Dense)	(None, 300)	270300
dropout_9 (Dropout)	(None, 300)	0
dense_10 (Dense)	(None, 100)	30100
dropout_10 (Dropout)	(None, 100)	0
dense_11 (Dense)	(None, 1)	101
dropout_11 (Dropout)	(None, 1)	0

Total params: 2,011,401
Trainable params: 2,011,401
Non-trainable params: 0

Figure 1: Final Dense Model Architecture

3.3. Fully-connected model

To explore different approaches to the problem and take full advantage of these different embedding structures, we trained two separate models with different architectures and compared their performance against each other. To fully exploit the density of information and learned features in the UniRep protein fragment embeddings, we used a fully connected, dense model on data pre-processed this way (Figure 1). The specific architecture of the first model was a fully connected neural network with different hidden layers of size 900, 300, 100, and 1, respectively. The activation function for each of the layers was ReLU. We trained using an Adam optimizer with mean squared error as our loss function. To choose the hyperparameters for our final model, we tested several different learning rates, as well as whether dropout and regularization after each layer would improve performance after 50 training epochs. We also compared the performance of a model trained using only the Pfam domain repression score data against a model trained using the combination of the two datasets to determine the optimal training strategy.

3.4. Convolutional model

For the model trained on one-hot amino acid embeddings, we wanted to leverage the spatial information still present in the data, so we used a convolutional neural network (CNN) model to search for motifs in the amino acid sequences of the fragments. The architecture we selected consisted of two conv-relu layers and maxpool, followed

by a fully connected layer (Figure 2). The first block is built with 40 filters, while the second block is built with 24 filters. We tested different 2D filter sizes of 10aa and 40aa in both blocks. We added padding to get activations of length 80 to match the length of our 80 aa input sequences.

We also experimented with regularization by adding dropout and batch normalization after each of the two convolutional blocks, right before the max pool layers, and after the fully connected layer. We set our dropout rate rate to 40%. However, we eventually settled on a non-regularized model given that regularization worsened performance on our validation set. For this model architecture, we also compared a model trained on the Pfam domain data only against a model trained on both training datasets.

3.5. Human proteome annotations

To annotate the human proteome with predicted repression scores, we used a list of human proteome sequences paired with their corresponding Ensembl IDs. First, we filtered out any sequences shorter than 80 aa in length, and we cleaned the sequence data to remove extraneous stop codons. We then split the human proteome up into 80 aa-long tiles with a 40 amino acid sliding window between the start of each tile. To fully tile proteins with a length that was not an exact multiple of 80, we included the last 80 amino acids of each protein as a tile. We filtered out human proteome sequences that were too close to our data in UniRep space (including test and validation samples). The smallest pairwise distance between two UniRep vectors in the Pfam dataset was 0.09, while the smallest pairwise distance within the tiling dataset was 0.13. So we used 0.15 as a threshold and excluded all human proteome tile UniRep embeddings that were within that threshold of any point in the HT-recruit dataset. We ended up excluding 2,389 tiles out of 263,193. The remaining 80aa tiles were then input into our final optimized model and the repression scores were predicted.

3.6. Metrics

We used mean absolute error (MAE) as our primary benchmark. Using MAE provides better interpretability for the loss, because the error is on the same scale as the repression and activation measurements. MAE also enables direct comparisons between models trained on different training data. In addition to MAE, we included other metrics to better report the overall performance of our models. In this study, we also used mean squared error (MSE), plots of training and validation loss over the training epochs, and comparison of predicted and true repression scores for our validation set to evaluate and improve the performance of both of our models.

To evaluate performance for our convolutional models considering the uncertainty in the measurement of the data,

Layer (type)	Output Shape	Param #
conv2d_103 (Conv2D)	(None, 1, 80, 40)	8040
conv2d_104 (Conv2D)	(None, 1, 80, 24)	9624
dropout_51 (Dropout)	(None, 1, 80, 24)	0
max_pooling2d_57 (MaxPooling	(None, 1, 40, 24)	0
flatten_55 (Flatten)	(None, 960)	0
dense_55 (Dense)	(None, 1)	961

Total params: 18,625
Trainable params: 18,625
Non-trainable params: 0

Figure 2: Final Convolutional Model Architecture

we also performed permutation tests on both biological replicates of the data and predicted repression levels to yield a z-score for each prediction. The bootstrapped z-score measures how likely the mean squared error or mean absolute error of the prediction is, under the null hypothesis that the predicted repression level is drawn from the same distribution as the observed repression levels. The higher the magnitude of the z-score, the further the prediction is from the null hypothesis and the less accurate our model is. A positive z-score implies a higher MAE; while a negative z-score, especially for the training set, means an artificially low MAE and hence indicates over-fitting.

4. Results

4.1. UniRep embedding with fully-connected model

The first experiment that we did tested different learning rates on our fully-connected model, trained only on the Pfam domain repression score data. We tried a wide range of learning rates from 10^{-9} to 10^{-1}, and we found the learning rate with the highest performance to be 3.16e-5. The resulting MAE on the validation set after 50 epochs with the learning rate 3.16e-5 was 0.464. The training loss and validation loss curves for this learning rate over the 50 epochs are reported in Figure 3. In this plot, the training loss drops quickly and remains slightly lower than the validation loss, which signifies that our model is over-fitting the training set.

We observed evidence of over-fitting in many of our learning rate experiments and attempted to reduce this problem with regularization. We added two dropout layers after the first and second hidden layers to regularize the model. The dropout rates we tested were $0.1, 0.2, 0.4, 0.5$, and 0.6. After 50 epochs of training, the validation MAEs corresponding to each dropout rate were $0.435, 0.436, 0.408, 0.440$, and 0.509, respectively. Plots for the training and validation loss curves of the model with the best dropout rate (0.4), as well as scatter plots of the true and predicted labels on the validation set, are included in Figure 4. For our optimal dropout rate of 0.4, we were able to reduce evidence of over-fitting, since the validation

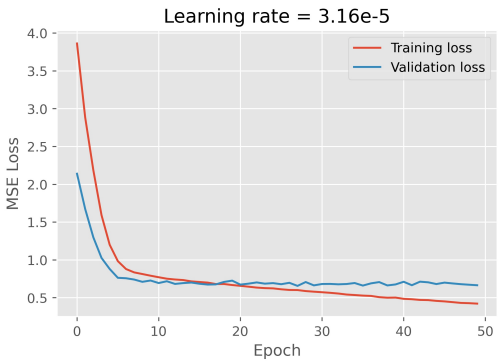


Figure 3: Fully-connected model training mean squared error loss (red) and validation loss (blue) for the optimal learning rate in our learning rate experiment.

loss stayed slightly below the training loss over 50 epochs (Figure 4). However, the training loss still exhibited a decreasing trend at the end of training. Extreme values for the dropout rate, such as 0.5 and 0.6, negatively impacted the model's performance and fit.

To try to further decrease over-fitting, we added regularizers to the model with the best-performing dropout rate (0.4). Three models with "L1-L2" regularizer parameters of 'L1=1e-4, L2=1e-2', 'L1=1e-2, L2=1e-3' and 'L1=1e-4, L2=1e-3' were used. The validation MAEs of these models after 50 epochs of training are 0.422, 0.539, and 0.430. The first 'L1=1e-4, L2=1e-2' regularization setting performed the best on the validation set. Although these MAEs were all higher than the original model with dropout and without regularization, we saw over-fitting (diverging training and validation losses) with the original model when training for 2000 epochs, so we used the "L1=1e-4, L2=1e-2" regularizer in our final fully-connected model (Figure 5).

We selected the best fully-connected model, which uses 0.4 dropout and an "L1-L2" regularizer, trained for 2000 epochs on only the Pfam domain data, and reached an MAE of .430 on the validation set and .498 on the test set. Although several methods were taken to prevent over-fitting, the model still suffered from over-fitting issues. Thus, the validation and test set performance were not as good as the training set performance.

Using these hyperparameters, we tried to determine whether using the larger repressor tiling dataset in combination with the Pfam dataset for training would improve our fully-connected model's ability to generalize. When we trained the model on the combination of the repressor tiling dataset and the Pfam domain fragments for 2000 epochs, the model achieved an MAE score of 0.400 on the combined validation set, 0.547 on the Pfam domain validation dataset alone, and 0.351 on the tiling validation dataset alone (Figure 5). The model performed better on the repressor tiling dataset than the Pfam domain dataset, probably due to the tiling dataset's larger size. Training on the combined dataset improved performance on the corresponding validation set from our original validation MAE of 0.429, so we used this combined model as our final model to improve generalizability to many different types of protein fragment sequences. Training this final model on the combined dataset for 2000 epochs resulted in an MAE of 0.380 on the combined test set. Given how close the test set MAE (0.380) was to the validation set MAE (0.400), we were confident that our performance gain was not due to overfitting.

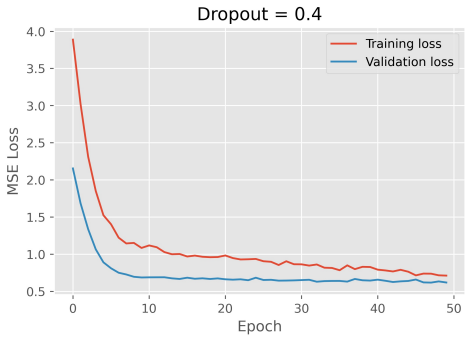
4.2. One-hot encoding with convolutional model

Next, we built and evaluated convolutional models with one-hot encoded amino-acid data of shape $(1, 80, 20)$. We initially started with an architecture with 6 convolutional layers. Surprisingly, we found simple one-layer and two-layer convolutional networks perform comparably to deeper models on this dataset. We tested different values for several hyperparameters, including the number of filters in each layer, dropout rate, and early stopping patience. Our convolutional model with the best-performing hyperparameter combination had 40 filters in the first layer and 24 in the second layer, a filter size of 10, a dropout rate of 0.4, and early stopping patience 2 as shown in Figure 6. The learning rate for the final model was $5e-4$. Our best-performing two-layer convolutional model achieves a mean squared error of 0.712 and mean absolute error of 0.614 on the validation set. The MAE on the test set is comparable at 0.591, but remains much higher than the MAE of the best FC model on the test set.

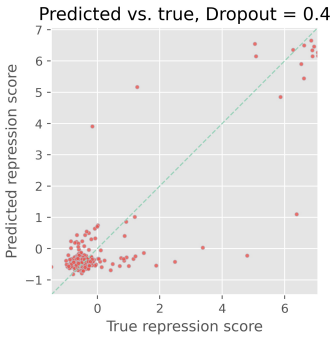
The permutation test on MSE yields high z-scores for the training set (30.27) and validation set (8.23), indicating that we avoided over-fitting because of our early stopping criteria (Figure 7). In contrast, an over-fitted model with 6 convolutional layers has an extremely negative z-score on the training set and comparable score on the validation data.

We visualized the filters in the first layer and colored by the chemical properties of the amino acids (Figure 8) but did not observe any clear motif patterns since many of the filters had 4 or 5 amino acids weighted heavily in each sequence position. Assuming that the motifs may be different across different protein families or domain groupings but similar within these families, we attempted to cluster the motifs by K-means clustering and plot the centroids. However, there were still no discernible patterns or clear motifs in the cluster centroid visualizations (Figure 9).

For the CNN model, we also tried training on a combination of the repressor tiling dataset and the Pfam dataset using the hyperparameters we determined above for 50 epochs. In this combined model, we changed the filter size to 40 aa rather than 10 aa to achieve better performance and account for proteins with longer sequence motifs. The con-



(a) Training loss curve for model with dropout = 0.4



(b) Predicted vs. true values with dropout = 0.4

Figure 4: Fully-connected model training curve and predicted vs. true repression scores for the model we tested with the best dropout rate.

volutional model achieved an MAE score of 0.441 on the combined validation set, 0.568 on the Pfam domain validation dataset alone, and 0.399 on the tiling validation dataset alone. The model MAE on the combined test dataset was 0.452. The training loss curves and scatter plots of the training data showed less evidence of over-fitting, at the expense of performance on both individual datasets but especially the Pfam domain dataset (Figure 10). Due to over-fitting and worse performance demonstrated by the CNN model in most cases we tested, we chose to use the fully-connected UniRep model for our final predictions tiling human protein fragments.

4.3. Annotating the human proteome

We applied our final fully-connected model trained on UniRep embeddings of both Pfam domain and repressor tiling datasets to predict repression scores of 80 aa-long tiles along the human proteome. The short length of the tiles allows us to replicate our training data as closely as possible and get higher sub-protein-level resolution for our repression scores. The distribution of predicted scores is shown in Figure 11. We observed a skewed distribution with many protein fragments with no repressor activity and a smaller subset of protein fragments with high repressor activity, which is what we would expect to occur.

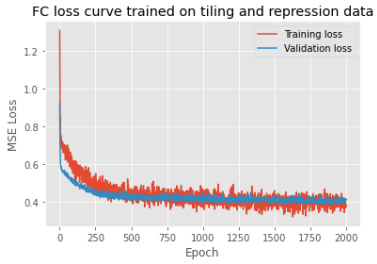
To validate the predictions of our model, we created a list of all of the genes that included fragments with scores higher than a threshold of 4, and then performed a Panther Overrepresentation test on gene ontology (GO) molecular function for these genes. The top two GO categories that these genes with highly scoring fragments were enriched in were "RNA polymerase II cis-regulatory region sequence-specific DNA binding" ($p = 7.90e - 72$), with a 12.76-fold enrichment, and "DNA-binding transcription repressor activity, RNA polymerase II-specific" ($p = 1.18e - 12$), with a 10.42-fold enrichment. This suggests that the model is correctly predicting high scores for the repression domains inside repressor and transcription factor proteins.

In addition to our Panther search, we searched for the functions of the proteins containing the 3 highest scoring fragments and the 3 lowest scoring fragments. The top 3 fragments all came from zinc finger proteins (ZNF267, ZNF732, and ZNF718), and the bottom 3 fragments were from different chains of human collagen (COL22A1, COL6A6, and COL11A2). While the top-scoring fragments are likely accurate predictions due to their zinc finger (transcription factor) categorization. We observed that some low-scoring fragments also belonged to proteins that were putatively associated with transcription repression. It is possible that our model could be incorrectly underestimating the repression scores of these tiles, which we have seen in previously on our training and validation data. An alternative explanation is that the model predictions are accurate and these low-scoring fragments simply do not contain the repressor domain inside the repressor protein. Collaborating with the authors of the HT-recruit paper to test selected protein fragments empirically will allow us to further validate our model's predictions.

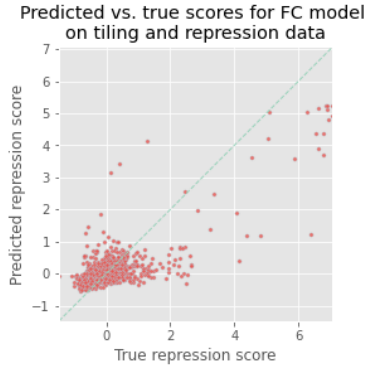
5. Discussion

5.1. Model Blind-spots

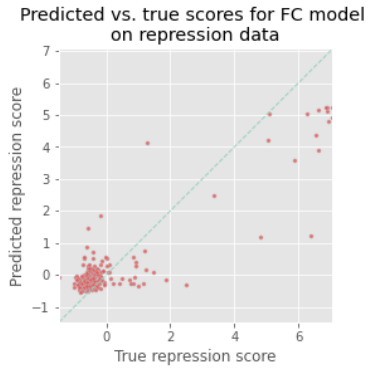
When taking a close look at our results, we noticed that both models predict low repression levels for a subset of protein domains that actually have moderate repression scores. We noticed this effect especially for the validation and test datasets. We were able to replicate these errors on the training set through different forms of regularization, such as early stopping and dropout. This under-prediction may be due to high sequence similarity between domains that have moderate repression activity and domains that have no repression activity at all, which the HT-recruit



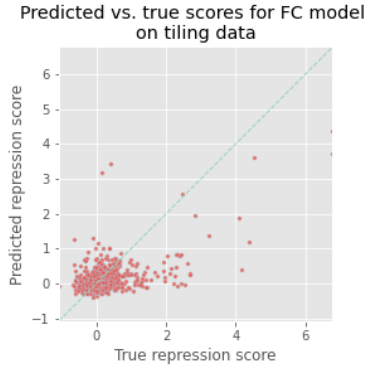
(a) Training and validation loss curves on the combined training data over 2000 epochs.



(b) Scatter plot of true versus predicted scores for the combined model on the combined Pfam domain and repressor tiling validation data.



(c) Scatter plot of true versus predicted scores for the combined model on the Pfam domain validation dataset only.



(d) Scatter plot of true versus predicted scores for the combined model on the repressor tiling validation dataset only.

Figure 5: Testing performance of fully-connected model trained on multiple datasets.

NumConvLayer	NumFilter	DropOut	Patience	TrainMSE	ValMSE	TrainZScore	ValZScore
1	(4)	0.4	0	0.730039389	0.958182954	15.96292033	6.258037262
1	(4)	0.2	0	0.585838903	1.001780061	15.49469132	6.587554565
1	(4)	0.1	0	0.667164605	1.022977562	16.37588593	6.330544525
1	(4)	0.6	0	0.923740109	1.197317143	17.71095405	6.16254915
1	(8)	0.4	0	0.606633951	0.950966906	15.76180599	6.258642964
1	(16)	0.4	0	0.644770336	1.092156458	15.71389075	6.29846794
1	(32)	0.4	0	0.488796521	1.047365322	14.80754101	6.351565158
2	(32, 4)	0.4	0	0.631780943	0.977677916	15.79291169	5.966024812
2	(32, 32)	0.4	0	0.390476627	0.809346546	11.6631952	5.780920094
2	(32, 32)	0.4	1	0.358189661	0.940064375	12.01716808	5.691759624
2	(32, 32)	0.3	0	0.362280813	0.885436613	11.61967515	6.208392432
2	(40, 32)	0.4	0	0.352477078	0.847621773	10.92671669	5.545266527
2	(48, 32)	0.4	0	0.337299997	0.919058404	10.56391877	5.88076111
2	(40, 36)	0.4	0	0.342455035	0.952735101	10.58496391	6.546478169
2	(40, 24)	0.4	0	0.353583259	0.918620319	11.20675856	5.500367504
2	(40, 16)	0.4	0	0.425431222	0.885875786	12.03394757	6.220438142
2	(40, 28)	0.4	0	0.326975518	0.800106642	10.47455581	5.596970408
3	(40, 24, 12)	0.4	0	0.511906071	0.879025053	13.16508681	5.876530811
2	(42, 28)	0.4	0	0.330083463	0.835299655	10.32601651	5.63176302
2	(40, 26)	0.4	0	0.208213469	0.867116701	6.685749602	6.081422407
2	(40, 24)	0.2	0	0.30772359	0.895423928	9.893200427	6.116650112
2	(40, 24)	0.3	0	0.245242249	0.785824316	8.331064361	5.545808292
2	(40, 24)	0.3	1	0.396739744	0.882344548	11.74829019	5.407735228
2	(40, 24)	0.3	2	0.17239426	0.801237572	5.851782241	6.132334303
2	(40, 24)	0.4	2	0.226991708	0.753656596	7.313218722	6.339958365

Figure 6: Hyperparameter search of the 2 layer CNN. We eventually chose the last row of hyperparameters for our final CNN model.

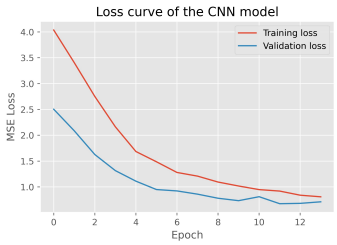
authors had previously observed [1].

5.2. Over-fitting on Training and Validation Data

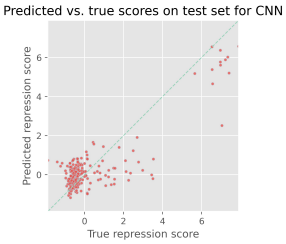
The question of over-fitting comes up when comparing the fully connected model to the convolutional neural network. While both models exhibit a gap between training and validation loss, over-fitting is more of a concern for the convolutional model and we see this in several ways. First, from an architectural standpoint, the convolutional model is more expressive and we are able to get an almost perfect fit on the training data. Second, the fully-connected model relies on UniRep embeddings which act as a form of regularizer. The convolutional network can exploit one-off variation in a single amino acid, while it is more challenging to exploit variations in embeddings that tend to be similar within a protein family. Our over-fitting concerns are apparent when looking at motifs learned by CNN filters as they do not seem to pick out clear patterns (Figure 9). Furthermore, our observations are supported by the lower test set MAE for the final fully connected model (0.380) relative to the final convolutional model (0.452). Due to overfitting, the convolutional model ultimately performed worse than the fully-connected model and resulted in the fully-connected model being chosen to make our final predictions across the human proteome.

6. Conclusions and Future Directions

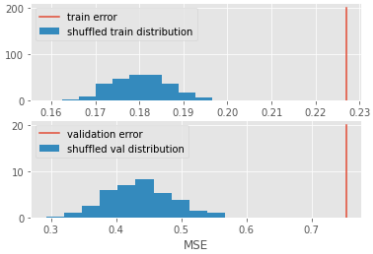
We have developed and tested two models that predict the repression ability of 80 aa-long protein sequences. In a comparison of the two models, the fully connected network based on UniRep embeddings outperformed the convolutional neural network for one-hot encoded sequences. We have identified several areas in which the model accuracy could be improved: by expanding our training dataset, and by modeling several other features. Due to the lack of solved structure data for many of our protein fragment sequences, we were unable to incorporate structure into either of our models. However, it is possible to utilize secondary



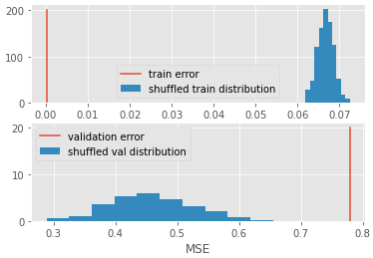
(a) Training and validation loss curves over the final CNN model training epochs.



(b) Scatter plot of predicted versus true scores of the final CNN model on the test set.



(c) Permutation test of the final 2 layer model. Histogram of the MSE in blue, Final MSE of the model in red.



(d) Permutation test of an over-fitted model without early stopping.

Figure 7: Testing performance of our tuned convolutional model trained on only the Pfam domain dataset.

structure predictions from structure prediction servers as features in a future version. In this work, we also limited our training set to only highly expressed protein fragments. We may be able to utilize data from protein fragments that were not well-expressed and further improve accuracy if we incorporate some measure or prediction of fragment expression into our feature set. To avoid overfitting, we can also train on experimental datasets containing tiling sequences that are more similar to each other to train our model to

Figure 8: Sample motifs learned by the CNN. Weight matrices are drawn from the 40 convolutional filters in the first layer. Each subplot is one learned filter. The x axis is the position in the weight matrix, y the weight in the corresponding filter. We used cutoff at one standard deviation of all weights in plotting, so as to reduce noisy characters. Amino acids are colored according to chemistry.

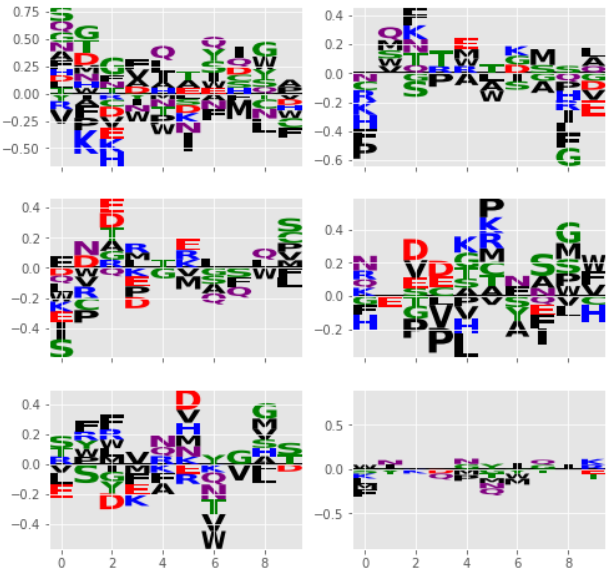
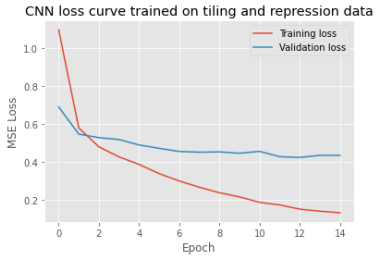
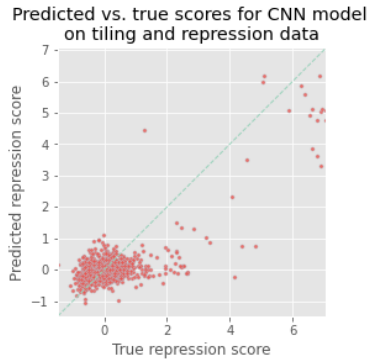


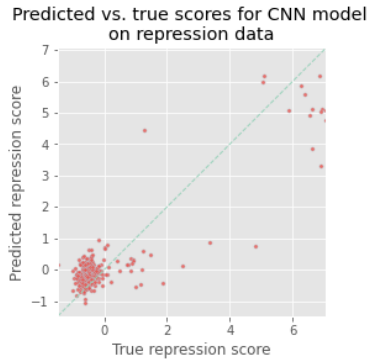
Figure 9: Motifs of cluster centroids. Weight matrices are drawn from the 6 K-means clustering centroids over the 40 filters in the first layer. Each subplot is one centroid. The x axis is the position in the weight matrix, y the weight in the corresponding filter. We used cutoff at one standard deviation of all weights in plotting, so as to reduce noisy characters. Amino acids are colored according to chemistry.



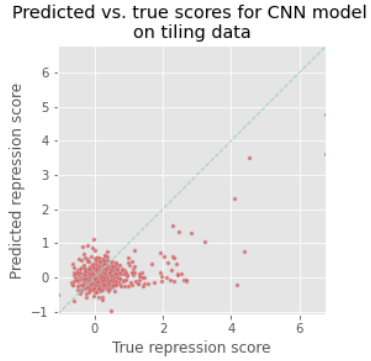
(a) Training and validation loss curves on the combined training data over 50 epochs.



(b) Scatter plot of predicted versus true scores for the combined model on the combined Pfam domain and repressor tiling validation data.



(c) Scatter plot of predicted versus true scores for the combined model on the Pfam domain validation dataset only.



(d) Scatter plot of predicted versus true scores for the combined model on the repressor tiling validation dataset only.

Figure 10: Testing performance of convolutional model with filter size 40 trained on multiple datasets.

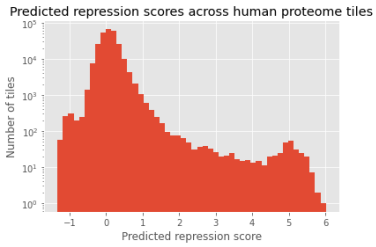


Figure 11: Histogram of the predicted repression scores across 80 aa tiles of the human proteome.

pick up subtle differences between sequences. We would like to augment our data to include more positive examples of domains with moderate and high repressor activity to make it more balanced. We have developed two models for repression score prediction that lay the groundwork for future advances in this space. The better-performing fully-connected model was applied to sequences tiling the human proteome, and these predictions were validated by bioinformatics analyses. Eventually, such a model would be applied to any arbitrary synthetic amino acid sequence. This will enable discovery of previously unknown repressor domains and facilitate design and engineering of new ones.

7. Code

All project code and selected data files can be found in the following public GitHub repository: https://github.com/o-fadil/cs273b

References

[1] Josh Tycko, Nicole DelRosso, Gaelen T. Hess, Aradhana, Abhimanyu Banerjee, Aditya Mukund, Mike V. Van, Braeden K. Ego, David Yao, Kaitlyn Spees, Peter Suzuki, Georgi K. Marinov, Anshul Kundaje, Michael C. Bassik, and Lacramioara Bintu. High-throughput discovery and characterization of human transcriptional effectors. *bioRxiv*, 2020.

[2] Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M. Church. Unified rational protein engineering with sequence-only deep representation learning. *bioRxiv*, 2019.

[3] Ariel Erijman, Lukasz Kozlowski, Salma Sohrabi-Jahromi, James Fishburn, Linda Warfield, Jacob Schreiber, William S. Noble, Johannes Söding, and Steven Hahn. A high-throughput screen for transcription activation domains reveals their sequence features and permits prediction by deep learning. *Molecular Cell*, 78(5):890 – 902.e6, 2020.