Modeling DNA Sequence Determinants of Regulatory Activity

Mateusz Borowski
Kacper Krzywicki
Katarzyna Łyczek
Michał Piasecki

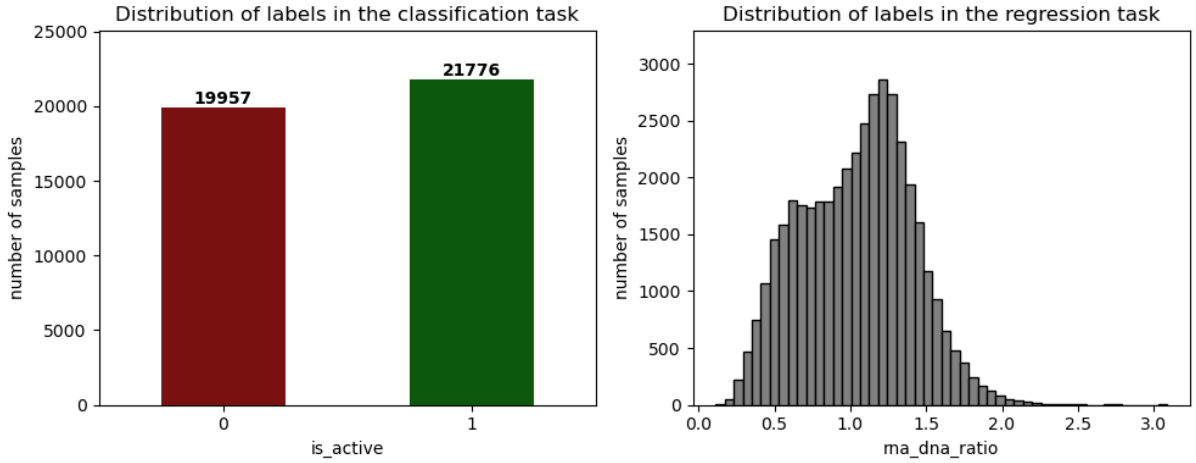
April 23, 2025

1 Data exploration

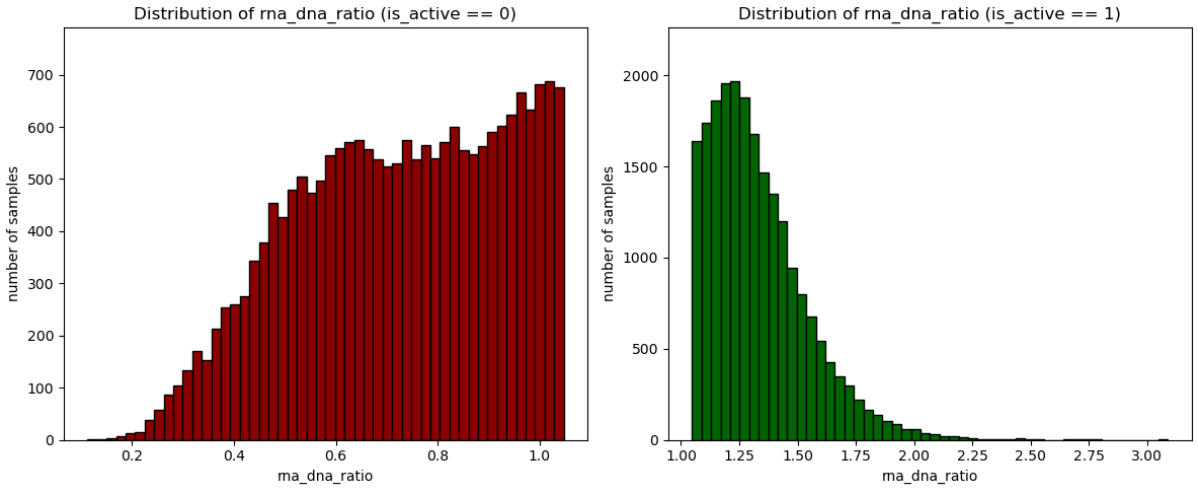
The dataset consists of 41733 samples. In view of the binary classification task, the dataset is well-balanced. In view of the regression task, the response variable assumes values between min = 0.11 and max = 3.09, is centered close to value 1 with standard deviation around 0.37.

statistic	value
mean	1.0412
std	0.3668
min	0.1129
25%	0.7525
50%	1.0702
75%	1.2997
max	3.0903

Table 1: Descriptive statistics of the `rna_dna_ratio` variable.



We observed that all sequences with `is_active` = 0 have `rna_dna_ratio` ≤ 1.0472, whereas all sequences with `is_active` = 1 have `rna_dna_ratio` ≥ 1.0474. Hence, assuming we had a well-trained regressor, we could build a classifier using solely the regression prediction together with the threshold of ≈ 1.04.



2 Methodology

Input sequences are long and the number of provided examples is relatively small, making training models from scratch a very difficult task. That is why we decided to test two different architectures that have already demonstrated strong performance in similar tasks: ESM (Evolutionary Scale Modeling) and DeePromoter.

2.1 DeePromoter

We copied the architecture of DeePromoter from official repository. Since original problem contained sequences with similar lenghts to our problem (300 in original DeePromoter vs 271 in our problem), we decided to train it with default parameters.

2.2 ESM

Standard ESMForSequenceClassification class from the transformers module was used for the classification problem. For the regression problem, however, we used a very similar architecture but with an additional linear layer at the very end.

Since we noticed that fine-tuning all parameters of ESM quickly leads to overfitting, we decided to then experiment with freezing some of the model's parameters during fine-tuning.

2.3 Validation

We use accuracy as the evaluation metric for the classification task and mean squared error (MSE) for the regression task.

Ideally, cross-validation would be performed to provide a more robust evaluation of model performance and to ensure the generalizability of results across different data partitions. However, due to computational and time limitations, we were unable to conduct cross-validation. Instead, we assess the models using a train-test split, with 70% of the dataset used for training and the remaining 30% for testing. A fixed random seed is used across all experiments for better comparability.

3 Results

	train	**test**
ESM without freezing	0.9693	0.6937
ESM with freezing	0.6524	0.6517
DeePromoter without freezing	0.811	0.7153

Table 2: Classification accuracy on training and test datasets after 20 epochs.

Accuracy of around 70% was achieved already after two epochs, after that the train accuracy continued to improve, whereas test accuracy stayed at the same level - signalling strong overfitting to the training set.

Since we did not notice significant differences in the performance between the two architectures, we decided to focus on just one of them and chose ESM.

	train	**test**
ESM without freezing	0.004	0.030*
ESM with freezing	0.114	0.114
DeePromoter without freezing	0.135	0.190

Table 3: MSE error on training and test datasets after 20 epochs.

Evaluation metric chosen for this task was Mean Squared Error. Even though ESM without freezing was evaluated only on a subset of test data, we decided it achieved the best results out of all the experiments.

4 Discussion

4.1 Limitations

- The models considered here are quite large and expressive, while our dataset for finetuning is rather small.
- The models also use their own tokenizers, which makes the learning and evaluation processes more computationally expensive.

4.2 Future directions

- Adding more complex layers as a regression head may lead to better results and should be explored.
- Understanding how RNA-to-DNA ratio depends on the genetic sequence or when it is labeled as active and including this knowledge in the model may also prove beneficial to the results.
- Our regression model might be overfitted since it has a very low error on the training set. More experiments or less training epochs might improve the results.
- Frozen models were improving very slowly during training. This might be due to the complexity of the dataset, but increasing learning rate or significantly increasing the number of epochs might be worth testing.

References

[1] Meta AI Research. *ESM: Evolutionary Scale Modeling*. Available at: `https://github.com/facebookresearch/esm`, Accessed: April 23, 2025.

[2] Chao Ego. *DeePromoter: PyTorch Implementation of Robust Promoter Predictor Using Deep Learning*. Available at: `https://github.com/egochao/DeePromoter`, Accessed: April 23, 2025.