Predicting gene and protein expression levels from DNA and protein sequences exploiting Transformer-based architectures.

Angelo Casolaro
University of Naples Parthenope
and University of Pisa

Nicola Dall'Asen
University of Trento
and University of Pisa

Martin Menabue
University of Modena and Reggio Emilia

Francesca Miccolis
University of Modena and Reggio Emilia

1 Introduction

Proteins are functional units encoded by DNA, involved in many biological processes, such as cell signaling, regulation of gene expression, immune response, enzymatic reactions, transport, and fundamental for the structure, function, and regulation of cells and tissues. The quantification of protein expression levels could address many scientific questions in various applications, bringing benefits in clinical and biological research and fields such as agriculture, the food industry, biofuel production, and environmental science.

The mRNA expression level has been investigated for decades because of the strong correlation with gene function and regulation to discover biological insights, correlation to disease evolution, drug development, and other research tasks.

Several models have been developed to predict mRNA expression levels, often using CNN or LSTM [1]. These architectures present limitations when applied to long sequences such as DNA ones to model far-field interactions. For this reason, since 2017, the attention mechanism has become the standard for processing long sequences because of the possibility of capturing these long-range interactions.

This project is based on the [2]: in this work, a Perceiver architecture is applied to DNA sequences to predict mRNA and protein expression levels. The input is a series of letters representing respectively the nucleobases of the DNA and the aminoacids of the proteins. The output of the model is a number representing the genic or protein expression.

2 Motivations

We explore three possible areas that we can modify to improve the performance of the original architecture, and we use the same color to highlight the area in Figure 1:

(a) **Latent dimension choice:** We employ an intrinsic dimensionality estimator [3] to provide an educated guess to the latent array dimension, differently from the original fixed size of 128.

(b) **Sparsity:** We deem the original consideration to use a dense $1D$ convolution on the sparse input of the DNA to be suboptimal. We propose a dense representation of the Input DNA through an embedding.

(c) **Transformation of input mRNA half-life features:** We learn a non-linear transformation of the mRNA half-life features instead of directly concatenating this information as in the

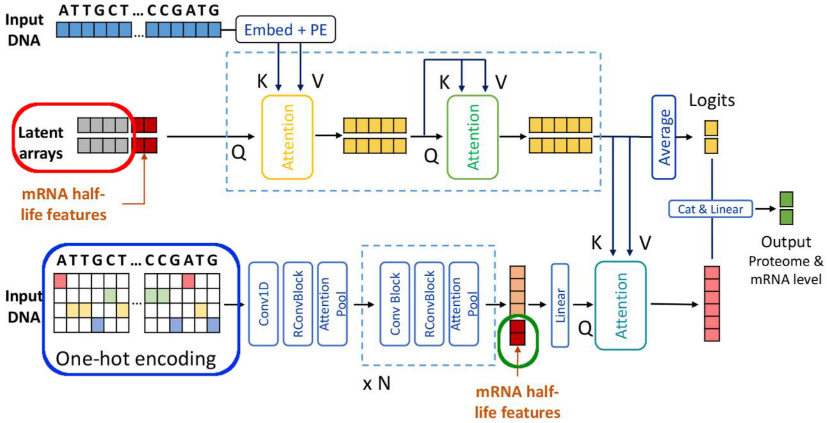

Figure 1: Original architecture [2] with the modified areas highlighted.

original architecture. In this way, we could learn how to employ the RNA information end-to-end.

3 Experiments

Starting from the reasons mentioned earlier, we devise experiments driven by practical considerations to refine the initial architecture. We approach this in two ways: first, we work on making the training process more efficient, and second, we tweak how the network handles incoming data. For practical reasons, we perform the proposed experiments on one of the original architectures, particularly the one that predicts protein expression level from gene and protein expression. However, these modifications can be applied to the other two models.

3.1 Training parameters

We exploit the underlying cluster architecture to propose practical modifications to the training hyperparameters to make the training loop more efficient.

- **Batch size:** Following best practices for the NVIDIA A100 GPUs available for the project, we change the original batch of 96 to the closest multiple of 64, 128.
- **Parallelism:** We adapt the original code to work in multi-GPU and multi-node settings using Accelerate [4], an open-source PyTorch wrapper to perform distributed training.
- **Training precision:** We study the effect of using mixed precision (bf16) *w.r.t.* full precision (fp32).
- **Efficient attention:** the original architecture [2] relies on Percevier [5] to reason on the input, which builds on top of the Transformer model [6]. Given the full-attention nature of the network, we optimize this complex operation by employing xFormers [7].

3.2 Architecture

We propose several modifications to the input data transformation, in particular:

- **Latent dimension**: An important modification of the original architecture made in this project is related to the *intrinsic dimension* (ID) defined for the input DNA sequence. In detail, the idea is to approximate the number of latent units considered in the first cross-attention layer, performing several intrinsic dimension estimators, e.g., *Grassberger-Procaccia* algorithm [3], to model adequately the correlations between latent variables and the input DNA sequence. The ID estimated is equal to 16, computed by the mean of all intrinsic dimension estimators studied.
- **Sparse input**: Instead of the original one-hot encoding of the Input DNA, we take the original indices from the dataloader representing the nucleobases, and we use an embedding

layer to obtain learnable vectors corresponding to each base. The dimensionality of the embedding is chosen to be compatible with the rest of the architecture, *i.e.* we choose 128.

- **mRNA input**: We learn a non-linear transformation of the mRNA half-life features by using an MLP with a non-linearity (GELU) between the two layers. The dimensionality is chosen to be the same as the original architecture, *i.e.* 8.

3.3 Results

We report the results of the experiments in Table 1, and we can divide the analysis into two parts:

- **Training parameters:** we can see that the adoption of mixed precision (Baseline (bf16)) brings a significant reduction in terms of training time per epoch (-9%) with a neglectable reduction in terms of R^2. By changing the other parameters (batch size, use of xFormers, and multi-node training), we can achieve the best results, which surpass the reproduction of the original training.
- **Architecture:** Reducing the latent arrays dimension (Baseline (16-dim)) decreases by 30 percent the number of parameters of the baseline model, maintaining a comparable accuracy for all provided experiments. On the other hand, increasing it (Baseline (512-dim)) hinders the performance, and we deem it to be overfitting since this modification introduces many parameters (5.18 million).
  The half-life projection and switching to a dense representation of the DNA do not improve the model's performance and introduce an overfitting phenomenon due to the number of parameters.

Table 1: Quantitive results and comparison of the proposed modifications. We compare the number of parameters, the number of GPUs used, batch size, latent array dimension, PyTorch dtype, and the use of efficient Attention against the seconds necessary to complete one training epoch and the final R^2.

Name	**Params**	**GPUs**	**Batch Size**	**Latent dim**	**Precision**	**xFormers**	**Time per epoch** ↓	$\mathbf{R^2}$ ↑
Baseline	1.04M	4	96	128	fp32	✗	3.22s	0.1323
Baseline (bf16)	1.04M	4	96	128	bf16	✗	2.93s	0.1308
Baseline (bs128)	1.04M	4	128	128	bf16	✗	2.97s	0.1359
Baseline (xFormers)	1.04M	4	128	128	bf16	✓	2.76s	0.1347
Baseline (multinode)	1.04M	32	128	128	bf16	✓	**2.47s**	**0.1383**
Baseline (16-dim)	0.72M	4	128	16	bf16	✗	2.81s	0.1330
Baseline (64-dim)	0.81M	32	128	64	bf16	✓	2.61s	0.1267
Baseline (512-dim)	5.18M	32	128	512	bf16	✓	3.95s	0.1271
Ours (half-life projection)	1.04M	4	128	128	bf16	✗	2.93s	0.1352
Ours (dense input)	3.47M	32	128	128	bf16	✓	3.37s	0.0989

References

[1] Vikram Agarwal and Jay Shendure. "Predicting mRNA abundance directly from genomic sequence using deep convolutional neural networks". In: *Cell reports* 31.7 (2020).

[2] Matteo Stefanini et al. "Predicting gene and protein expression levels from DNA and protein sequences with Perceiver." In: *Computer Methods and Programs in Biomedicine* 234 (2023).

[3] Peter Grassberger and Itamar Procaccia. "Measuring the strangeness of strange attractors". In: *Physica D: nonlinear phenomena* 9.1-2 (1983), pp. 189–208.

[4] Gugger Sylvain et al. *Accelerate: Training and inference at scale made simple, efficient and adaptable.* `https://github.com/huggingface/accelerate`. 2022.

[5] Andrew Jaegle et al. "Perceiver: General perception with iterative attention". In: *International conference on machine learning*. PMLR. 2021, pp. 4651–4664.

[6] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[7] Benjamin Lefaudeux et al. *xFormers: A modular and hackable Transformer modelling library.* `https://github.com/facebookresearch/xformers`. 2022.