SQUEEZE AND LEARN: COMPRESSING LONG SEQUENCES WITH FOURIER TRANSFORMERS FOR GENE EXPRESSION PREDICTION

Vittorio Pipoli[♣] *Giuseppe Attanasio*[♡] *Marta Lovino*[♣] *Elisa Ficarra*[♣]

[♣] University of Modena and Reggio Emilia, Modena, Italy
[♡] Bocconi University, Milan, Italy

ABSTRACT

Genes regulate fundamental processes in living cells, such as the synthesis of proteins or other functional molecules. Studying gene expression is hence crucial for both diagnostic and therapeutic purposes. State-of-the-art Deep Learning techniques such as Xpresso have proposed to predict gene expression from raw DNA sequences. However, DNA sequences challenge computational approaches because of their length, typically in the order of the thousands, and sparsity, requiring models to capture both short- and long-range dependencies. Indeed, the application of recent techniques like transformers is prohibitive with common hardware resources. This paper proposes FNETCOMPRESSION, a novel gene-expression prediction method. Crucially, FNETCOMPRESSION combines Convolutional encoders and memory-efficient Transformers to compress the sequence up to 95% with minimal performance tradeoff.

Experiments on the *Xpresso* dataset show that FNETCOMPRESSION outscores our baselines and the margin is statistically significant. Moreover, FNETCOMPRESSION is 88% faster than a classical transformer-based architecture with minimal performance tradeoff.[1]

***Index Terms*—** Deep Learning, DNA sequences, Fourier compression, gene-expression, transformers,

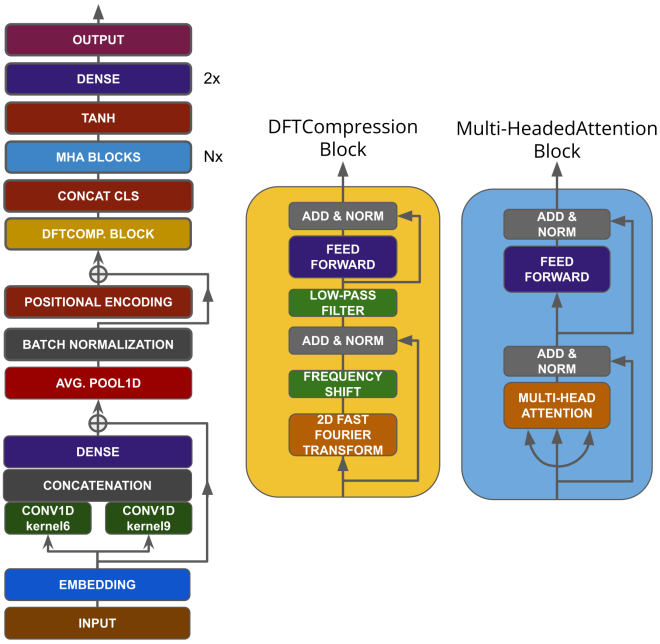

Fig. 1. FNETCOMPRESSION overview (left). After sequence embedding and pooling, DFTCompression (center) and MHA (right) layers compress and route information from the input sequence. Similar functional blocks share background colors.

1. INTRODUCTION

Gene Expression [1] regulates the existence of every living organism. It consists in the fundamental mechanisms the cells exploit to gather information from the deoxyribonucleic acid (DNA) and synthesize functional molecules (e.g., proteins) according to inherent regulatory mechanisms. Recent work has proposed to use Deep Learning (DL) models to predict gene expression directly from raw DNA sequences sampled and sequenced from living organisms, e.g., human tissues [2]. However, DNA sequences often count thousands of elements, and the signal within it is sparse: functional coding regions alternate with long non-coding parts. Length and sparsity make such sequences impractical for modern DL models, motivating increasing interest in compression for efficiency and noise reduction. Therefore, recent methods encode the sequence of original base pairs (bp) into shorter sequences, where each new token "represents" several bps. 1D Convolution layers [3], Long Short-Term Memory [4], and Transformer-based networks [5] have been adopted for the task [2, 6, 7, 8].

The nature of such DNA sequences requires gene expression prediction algorithms to learn from both local- and long-range interactions. For example, recent evidence found interactions among DNA elements at several kilo base-pairs (kbp) of distance [9]. Transformers models [5] provide a suitable method to learn from both short- and long-range dependen-

cies: the Multi-Headed Attention (MHA) mechanism. A typical MHA layer connects every input item with every other item and learns how to weigh every pair. By contrast, Convolutional Neural Networks (CNNs) [10] need a deep structure with many layers to enlarge the receptive field to distant elements. However, MHA's memory footprint grows quadratically with the sequence length, motivating recent research efforts on efficient transformers [11, 12]. FNet [13] is a prominent example: the network substitutes MHA with a Discrete Fourier Transform (DFT), a non-parametric, linearithmic token mixing strategy.

Contributions. This work introduces FNETCOMPRESSION, a novel approach to gene expression prediction from long DNA sequences. FNETCOMPRESSION uses convolution kernels, DFT-based transformers, and low-pass filters to compress input sequences and a final MHA layer for improved information routing. Results on gene-expression datasets [2] show that FNETCOMPRESSION significantly outperforms the baseline solution, reaching up to the 93% of performances of less efficient standard transformers despite compressing inputs by 95% of their length. Moreover, we conducted a qualitative analysis on FNETCOMPRESSION and discovered that i) attention weights are stronger on low-frequency components of the sequence, and ii) all elements contribute to the prediction.

2. RELATED WORK

Recent advances in sequence modeling and compression have motivated new neural gene expression models learning from raw DNA sequences. *Xpresso* [2] (Agarwal et al., 2020) is a state-of-the-art Deep Convolutional Neural Network[10] in the field of gene expression. The network predicts the steady-state gene expression levels in human and mouse organisms, exploiting DNA sequences and features associated with mRNA stability. The authors claim that *Xpresso* explains 59% of variation (measured with R^2) in steady-state mRNA levels in humans. Xpresso handles sequences of several thousand base pairs. The best-reported range is 7,000 bp and 3,500 bps, respectively, upstream and downstream Transcription Start Site (TSS). Note that the information around the TSS is an important proxy for gene-expression[14]. We build on Xpresso and use initial Convolutional layers for input summarization. However, we differ on the embedding of the nitrogenous basis, the type of pooling layers, and the transformer encoder. Expecto [6] (Zhou et al., 2018) is a Convolutional Deep Neural Network for predicting tissue-specific gene expression levels in humans. Unlike Xpresso, it requires additional biological information related to chromatin, defining different experimental conditions. Enformer [7] (Avsec et al., 2021) is a state-of-the-art transformer-based architecture for encoding even longer DNA sequences. Although Enformer and FNETCOMPRESSION share several architectural parts, e.g., pooling and transformer blocks, the former was devised to predict sequences of biological tracks. FNet [13] (Lee-Thorp et al., 2021) replaces the self-attention sublayers, which pay a quadratic complexity, with a standard, non-parametrized and linearithmic two-dimensional Fast Fourier Transform achieving 92-97% of the accuracy of BERT[15], but training 80% faster on GPU and 70% faster on TPU. We build on FNet to introduce FNETCOMPRESSION and add further compression layers to enhance efficiency.

3. DATASETS

Aiming for a fair comparison, we test FNETCOMPRESSION in existing gene-expression prediction setups. Specifically, we use the dataset of sequences introduced in Xpresso [2], which counts 18,377 genes. For each gene, Xpresso releases: 1. the DNA sequence (20,000 bp long); 2. the half-life features (that estimate the time required for degrading 50% of the existing mRNA molecules [2]) which are embedded in a vector of 8 real numbers for each gene; 3. the expression value, which is the label to be predicted. Moreover, both the validation and test set are obtained by sampling at random 1000 genes from the train set. These DNA sequences are arrays of nitrogenous bases extracted from the human reference genome. Moreover, the neighborhood of the Transcription Start Site (TSS) contains the most useful information for the prediction of gene expression [14]. Therefore, all the sequences are extracted and centered with respect to the TSS and contain the 10kbp upstream and downstream of it. The locations of about 15k TSS have been downloaded from the FANTOM5 consortium's UCSC data hub (Lizio et al.) [16]. For the remaining genes, Xpresso considered as TSS, among all the transcripts for each gene, the start coordinate of the one with the longest Open Reading Frame [17], followed by the longest 5' Untranslated Gene Region, followed by the longest 3' Untranslated Gene Region was selected [18]. The gene expression values were retrieved from the Epigenomics Roadmap Consortium [19]. In particular, the values were retrieved in a tabular format of normalized expression values for protein-coding genes across 56 tissues and cell lines obtained by RNA-seq data [2].

In addition to Xpresso's dataset, we perform experiments on a Controlled Test Bench (CTB) that removes the half-life features but relies on longer sequences, i.e., 65,536 bp. The TSS locations are downloaded by the FANTOM5 consortium's UCSC data hub (Lizio et al.) [16], and for the genes that are not covered, we decided to take the start coordinate of the longest transcript. CTB uses the same Xpresso target labels but different splits, built as follows. Chromosomes 8 and 10 were used for the test and validation splits, respectively, and the remaining chromosomes were used for the

[2]The preprocessing foresees the averaging among the tissues, ending up with one expression value per gene. After the aggregation, steps the values are then processed with a log-transformation ($\hat{y} = \log_{10}(y + \text{pseudocount})$, pseudocount=0.1) to reduce the right skew of the labels' distribution.

training set.[3] The resulting CTB training, validation, and test sets count 16,832, 683, and 618 sequences, respectively.

4. PROPOSED METHOD

This paper presents FNETCOMPRESSION, a novel method for gene expression level prediction. FNETCOMPRESSION uses a convolutional sequence embedding and a transformer encoder. The latter is composed of a non-parametric 2D Discrete Fourier Transform [20, 13], a subsequent low-pass filter to reduce the sequence length up to 95%, and an MHA layer for optimizing the final information routing.[4] The model takes as input DNA sequences tens of thousands of nitrogenous bases long (and optionally the half-life features vector, concatenated after the tanh pooler) and gives as output a real number that quantifies the gene expression level.

4.1. Sequence Embedding

As standard transformers handle sequences shorter than a thousand items [5], we first need to embed the input into a shorter sequence.

Unlike prior work using one-hot encoding [2, 7], we use an initial embedding layer to represent DNA basis as dense vectors. Next, two 1D convolutional layers with different kernel sizes (kernel1=6, kernel2=9) transform the sequence. Note that different kernels capture different local patterns from the sequence. Convolutional outputs are concatenated and projected via a dense layer to recombine the information, and a skip-connection [21] is used to facilitate gradient backpropagation. Next, we apply a 1D Average Pooling, i.e., the first compression step, Batch Normalization [22], and sum absolute sinewave Positional Encodings [5]. Note that empirical experiments revealed Batch Normalization to be crucial for sequence embedding. We hypothesize this layer improves numeric stabilization before the addition of positional information, ensuring proper weighing of semantic and positional information.

4.2. DFTCompression

The output of the sequence embedding stage is fed to the *DFTCompression* block, which learns long-range patterns and further compresses the input sequences.

First, we apply a 2D DFT and retain only the real part [13]. By a first approximation, the resulting sequence represents the same signal in the "frequency" domain. Using this time-frequency intuition, we apply a low-pass filter—i.e., the second and most prominent compression step—as follows. We shift the zero-frequency component of the sequence to the center of the sequence and cut out symmetrically the outermost positions. Our results have shown that we can push this compression to remove up to the 95% of the sequence while retaining most of the prediction accuracy. The output of the compression block is prepended with a special token and fed to a MHA.[5] The final part of the network consists of two dense layers with a ReLU activation function each and a final neuron that represents the output of our regression model.

5. RESULTS

We evaluated the learning capability of FNETCOMPRESSION compared to *Xpresso*'s model. Then, we tested generalization to longer sequences by reducing the pooling size on Xpresso's dataset and using the long sequences of our CTB. We compared FNETCOMPRESSION to four different baseline configurations: 1) the sequence embedder without any transformer encoder block, 2) *FNet_1_0* which has one DFT block and no MHA blocks, 3) *FNet_1_1* which has one DFT block and one MHA (i.e., with no compression or special tokens, similar to [13]), and 4) a Transformer with two encoder blocks. All these models are obtained by removing the DFTCompression block from the backbone depicted on the left in Figure4 and modifying the blocks of the totem pole that follow the concatenation of the special token. Moreover, we provide the study of the computational complexity paid by the models, the attention maps, and gradient x input analysis.

Confidence intervals have been computed with 14 runs per experiment, a confidence level 0.95, the unbiased standard deviation estimator, and t-student distribution.

5.1. Training details

All the methodologies have been fitted employing the Adam optimizer[23] exploiting a warm-up step scheduler[5]. The loss metric adopted is Mean Squared Error (MSE) and the test metric is R^2. The compression rate of FNETCOMPRESSION is always set to 95%. All the MHA blocks have four heads. Refer to our github for the rest of the hyperparameters. We adopted Google's Tesla T4 and TPU as hardware resources.

5.2. Performances on Xpresso Dataset and CTB

Here, we compare FNETCOMPRESSION (§4.2) with *Xpresso's model* [2] on their dataset. *Xpresso*'s gene prediction values have been the authors' code [24]. As shown in Table 1, FNETCOMPRESSION and *FNet_1_1* provide the best results even if FNETCOMPRESSION reduces the input sequences length of 95%. Experiments on CTB dataset show that FNETCOMPRESSION outperforms *FNet_1_1* with sequences long three times Xpresso's ones.

[3] Genes have different lengths, and extracting a fixed-size window of base pairs can result in extracting the information of multiple genes. Stratifying on chromosomes prevents any overlap between training and test sequences..

[4] Code and data are available at `https://github.com/vittoriopipoli/FNetCompression`.

[5] Using starting special tokens is commonplace in Computer Vision and Natural Language Processing. The token is often used to summarize the sequence.

Dataset	Method	Low_CI	Mean_CI	Upp_CI
Xpresso	Xpresso	0.5593	0.5668	0.5743
	Seq. Emb.	0.5343	0.5422	0.5501
	FNet_1_0	0.5567	0.5604	0.5641
	FNet_1_1	0.6121	0.6183	0.6245
	FNetComp.	0.6076	0.6133	0.6190
CTB	FNET_1_1	0.5786	0.5859	0.5931
	FNetComp.	**0.5944**	**0.6006**	**0.6068**

Table 1. Gene expression R^2 on the test set of the Xpresso's dataset and CTB (0.95 confidence levels).

Method	Pool Size	2DFFT O(nlogn)	MHA O(n^2)	Relative perf.	Time per batch [s]	Speed up
Transformer	128	-	156x2	-	36	-
FNet_1_0		156	0	85%	32	+11%
FNet_1_1		156	156	93%	34	+6%
FNetComp.		156	8	**93%**	32	**+11%**
Transformer	32	-	626x2	-	60	-
FNet_1_1		626	626	89%	48	+25%
FNetComp.		626	34	**93%**	32	**+87.5%**

Table 2. Speed up and performance comparisons of FNETCOMPRESSION and *FNet_1_1* with a classic Transformer architecture on Xpresso's dataset using a Tesla T4 GPU.

5.3. Computational Complexity

We studied the computational complexity of the tested models. Table 2 reports the results. FNETCOMPRESSION's speed-up over *FNet_1_1* increases with the input sequence length, as an expected result of our compression stages. Moreover, FNETCOMPRESSION performance remains stable unlike *FNet_1_1*.

We do not report the comparison of execution times on the CTB dataset due to out-of-memory errors in the testing environment. Preliminary tests on TPU hardware proved FNETCOMPRESSION as the fastest model but by a smaller margin.

5.4. Attention and Gradient x Input Analysis

Attention plots can reveal some interesting patterns in transformer architectures' data modeling. In particular, the attention patterns of the Multi-Headed Attention block that follows the *DFTCompression block* of our model can be examined. As we can see in Figure 2, it is possible to spot vertical patterns in the middle of the matrix. Vertical patterns occur when all the elements of a sequence are paying attention to the same location. Therefore, most of the elements are paying attention to the regions that embed the lowest frequencies.

When FNETCOMPRESSION applies a compression factor of 95%, only the 5% of the sequences in processed by the subsequent Multi-Headed Attention layer. Hence, we computed the Gradient x Input to prove that all the elements of the original sequence take part in the loss contribution. Results are

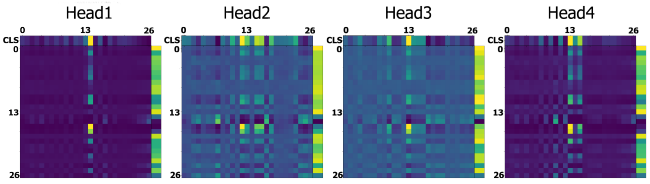

Fig. 2. Attention weights in FNETCOMPRESSION trained on CTB. Attention values expressed (first row) and received (last column) by the special token are magnified and min-max normalized.

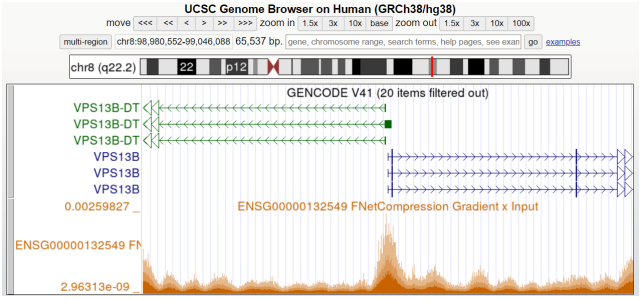

Fig. 3. Gradient x Input attribution obtained by feeding FNETCOMPRESSION with the CTB test gene VPS13B.

shown in Figure 3, and it is possible to notice that all the nitrogenous bases have a significant contribution and the signal follows a sinusoidal pattern.

6. CONCLUSION

This work presented a transformer-based[5] model, called FNETCOMPRESSION, for predicting gene expression levels from raw DNA sequences exploiting a crucial sequence compression. The main challenge of this work is to deal with the quadratic complexity of the attention mechanism by designing a transformer-based architecture that exploits a 2D DFT that can analyze and compress long DNA sequences even with few computational resources. Results proved that FNETCOMPRESSION [4.2] outperforms *Xpresso* on their dataset. Hence, *Xpresso*'s authors claim to explain up to the 59% of the variation of gene expression levels, while FNETCOMPRESSION explains up to 62%. The comparison between FNETCOMPRESSION and *FNet_1_1* shows that FNETCOMPRESSION is capturing all the useful information even if it is discarding the 95% of the sequences. On the other hand, *FNet_1_1* become unstable when its input length grows. Finally, FNETCOMPRESSION is the fastest algorithm of these experiments. For future works, we suggest finding better ways for exploiting the 2D DFT and compression strategies.

7. ACKNOWLEDGMENT

This study was funded by the European Union's Horizon 2020 research and innovation programme DECIDER under Grant Agreement 965193 and by Fondazione Cariplo (grant No. 2020-4288, MONICA).

8. REFERENCES

[1] Jill U. Adams, *Differential Control of Transcription and Translation Underlies Changes in Cell Function*, MA: NPG Education, Cambridge, 2010.

[2] Vikram Agarwal and Jay Shendure, "Predicting mrna abundance directly from genomic sequence using deep convolutional neural networks," *Cell reports*, vol. 31, no. 7, pp. 107663, 2020.

[3] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman, "1d convolutional neural networks and applications: A survey," 2019.

[4] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin., "Attention is all you need doi," *arXiv 1706.03762*, 2017.

[6] Jian Zhou, Chandra L Theesfeld, Kevin Yao, Kathleen M Chen, Aaron K Wong, and Olga G Troyanskaya, "Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk," *Nature genetics*, vol. 50, no. 8, pp. 1171–1179, 2018.

[7] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley, "Effective gene expression prediction from sequence by integrating long-range interactions," *Nature methods*, vol. 18, no. 10, pp. 1196–1203, 2021.

[8] David R Kelley, Yakir A Reshef, Maxwell Bileschi, David Belanger, Cory Y McLean, and Jasper Snoek, "Sequential regulatory activity prediction across chromosomes with convolutional neural networks," *Genome research*, vol. 28, no. 5, pp. 739–750, 2018.

[9] Amartya Sanyal, Bryan R Lajoie, Gaurav Jain, and Job Dekker, "The long-range interaction landscape of gene promoters," *Nature*, vol. 489, no. 7414, pp. 109–113, Sept. 2012.

[10] Yann LeCun and Yoshua Bengio, *Convolutional Networks for Images, Speech, and Time Series*, p. 255–258, MIT Press, Cambridge, MA, USA, 1998.

[11] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al., "Rethinking attention with performers," *arXiv preprint arXiv:2009.14794*, 2020.

[12] Iz Beltagy, Matthew E Peters, and Arman Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[13] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon, "Fnet: Mixing tokens with fourier transforms," 2021.

[14] Philipp Kapranov, "From transcription start site to cell biology," *Genome Biology*, vol. 10, no. 4, pp. 217, 2009.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

[16] Marina Lizio, Jayson Harshbarger, Hisashi Shimoji, Jessica Severin, Takeya Kasukawa, Serkan Sahin, Imad Abugessaisa, Shiro Fukuda, Fumi Hori, Sachi Ishikawa-Kato, et al., "Gateways to the fantom5 promoter level mammalian expression atlas," *Genome biology*, vol. 16, pp. 1–14, 2015.

[17] Schuster Stefan Sieber Patricia, Platzer Matthias, "The definition of open reading frame revisited. [PMID]," *PubMed*, vol. 34, no. 3, pp. 167–170, 2018.

[18] Lucy W Barrett, Sue Fletcher, and Steve D Wilton, "Untranslated gene regions and other non-coding elements," 2013, Springer. ISBN 978-3-0348-0679-4.

[19] Bradley E Bernstein and John A Stamatoyannopoulos, "The NIH roadmap epigenomics mapping consortium," *Nature Biotechnology*, vol. 28, no. 10, pp. 1045–1048, Oct. 2010.

[20] David H. Bailey and Paul N. Swarztrauber, "A fast method for the numerical evaluation of continuous fourier and laplace transforms," *SIAM Journal on Scientific Computing*, vol. 15, no. 5, pp. 1105–1110, 1994.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," 2015.

[22] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, Francis Bach and David Blei, Eds., Lille, France, 7 2015, vol. 37 of *Proceedings of Machine Learning Research*, pp. 448–456, PMLR.

[23] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] Shendure J Agarwal V, "Predicting mrna abundance directly from genomic sequence using deep convolutional neural networks," Xpresso Colab.