This is a pre print version of the following article:

Predicting gene expression levels from DNA sequences and post-transcriptional information with transformers / Pipoli, Vittorio; Cappelli, Mattia; Palladini, Alessandro; Peluso, Carlo; Lovino, Marta; Ficarra, Elisa. - In: COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE. - ISSN 0169-2607. - 225:(2022), pp. 107035-107044. [10.1016/j.cmpb.2022.107035]

Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

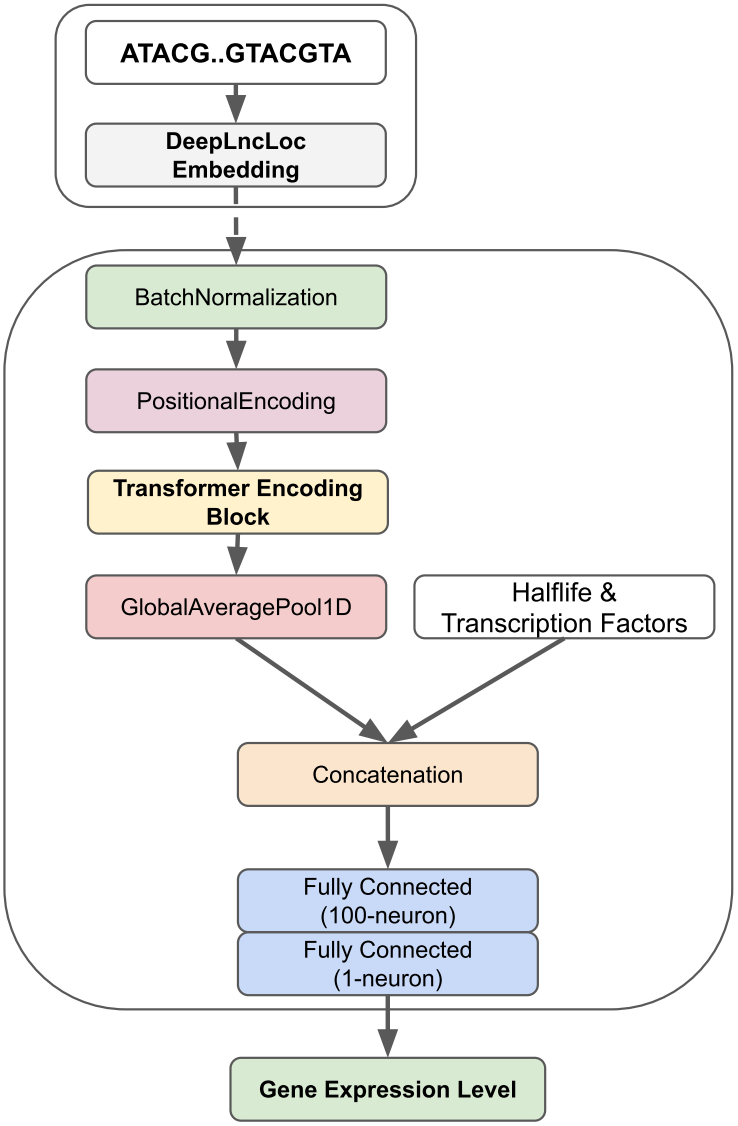
20/04/2024 13:27

(Article begins on next page)

Graphical Abstract

Predicting gene expression levels from DNA sequences and post-transcriptional information with transformers

Vittorio Pipoli, Mattia Cappelli, Alessandro Palladini, Carlo Peluso, Marta Lovino, Elisa Ficarra



Highlights

Predicting gene expression levels from DNA sequences and post-transcriptional information with transformers

Vittorio Pipoli, Mattia Cappelli, Alessandro Palladini, Carlo Peluso, Marta Lovino, Elisa Ficarra

- Predicting gene expression levels is crucial due to its clinic applications.
- Post-transcriptional processes are essential in understanding the gene expression regulatory mechanisms.
- Previous models do not include post-transcriptional information.
- We present Transformer DeepLncLoc (a transformer-based architecture) to predict gene expression levels from DNA sequences and transcription factors post transcriptional regulation.
- Transformer DeepLncLoc reached 0.76 of the R2 evaluation metric, outperforming existing methods.
- Transcription factor post-transcriptional regulation resulted in a massive performance boost.

Predicting gene expression levels from DNA sequences and post-transcriptional information with transformers

Vittorio Pipoli[a], Mattia Cappelli[a], Alessandro Palladini[a], Carlo Peluso[a], Marta Lovino[b], Elisa Ficarra[b]

[a]*Department of Control and Computer Engineering, Corso Duca degli Abruzzi, 24, Turin, 10129, Piedmont, Italy*
[b]*Enzo Ferrari Engineering Department, University of Modena and Reggio Emilia, Via P. Vivarelli, 10, Modena, 41125, Emilia Romagna, Italy*

Abstract

Background and Objectives: In the latest years, the prediction of gene expression levels has been crucial due to its potential applications in the clinics. In this context, Xpresso and others methods based on Convolutional Neural Networks and Transformers were firstly proposed to this aim. However, all these methods embed data with a standard one-hot encoding algorithm, resulting in impressively sparse matrices. In addition, post-transcriptional regulation processes, which are of uttermost importance in the gene expression process, are not considered in the model.
Methods: This paper presents Transformer DeepLncLoc, a novel method to predict the abundance of the mRNA (i.e., gene expression levels) by processing gene promoter sequences, managing the problem as a regression task. The model exploits a transformer-based architecture, introducing the DeepLncLoc method to perform the data embedding. Since DeepLncloc is based on word2vec algorithm, it avoids the sparse matrices problem.
Results: Post-transcriptional information related to mRNA stability and transcription factors is included in the model, leading to significantly improved performances compared to the state-of-the-art works. Transformer DeepLncLoc reached 0.76 of R^2 evaluation metric compared to 0.74 of Xpresso.
Conclusion: The Multi-Headed Attention mechanisms which characterizes the transformer methodology is suitable for modeling the interactions between DNA's locations, overcoming the recurrent models. Finally, the integration of the transcription factors data in the pipeline leads to impressive gains in predictive power.

Keywords: attention, DNA, gene-expression, prediction, transcription-factors, transformers

1. Introduction

Gene expression is the process of producing a functional product from the instructions stored in the DNA. Predicting the abundance levels of these products - so, predicting the gene expression levels - is crucial for several applications, from drug discovery to pathway enrichment analysis.
Several studies proposed Machine Learning approaches to predict gene expression. This challenge can be addressed by exploiting sophisticated Deep Learning architectures on DNA reference sequences [1, 2, 3, 4, 11].
In detail, Convolutional Neural Networks (CNNs) were extensively adopted to address specific tasks, ranging from predicting tissue-specific expression from long promoter-proximal sequences (ExPecto, Zouh et al., 2018 [1]) to predicting the gene expression raw counts from CAGE and ChIP-seq experiments (Basenji, Kelley et al., 2018 [2]).
ExPecto [1] predicts tissue-specific expression from a wide regulatory region of 40-kbp promoter-proximal sequences and which genes are mutated. On the other hand, Basenji [2] is based on dilated convolutional filters that spot longer relationships in the inputs concerning standard convolutions. However, the limited receptive field of the Convolutional Networks (CNN) can not compete with the MultiHeadedAttention layer of a Transformer architecture [6], even using dilated filters. In addition, its main limitation consists in the use of cell line data that are more homogeneous than data from human tissues.
Regarding the solutions for predicting gene expression levels directly from the DNA sequence, Xpresso (Agarwal and Shendure [11], 2018) is the most complete, accessible, and reproducible project. Xpresso's [11] architecture is based on CNNs and, the hyperparameters were optimized using a meta-heuristic approach. Indeed, a small change in one of the hyperparameters can substantially decrease the performance and lower the stability and robustness of the architecture.
Cutting-edge deep architectures were then proposed bringing further improvements. The Enformer network (Avsec et al., 2021 [4]) takes steps forward compared to Basenji by introducing a Transformer architecture [6] to integrate long-range interactions in the genome.

A common limitation of all the cited models is the embedding of the input sequences. Every model uses a one-hot encoding, which leads to sparse matrices, which are not very informative.
Here, we propose some alternatives based on Word2Vec embeddings [9] and a domain-specific embedding method called DeepLncLoc [8].
In addition, we present an innovative pipeline called **Transformer DeepLncLoc**. Such a method relies on the transformer's [6] capability of modeling long-range dependencies and a task-aware embedding, overcoming the classical CNN-based solutions. As its name suggests, it is built on top of the DeepLncLoc embedding [8] and a vanilla Transformer Encoder Block [6].
Moreover, we propose two additional reference models, used as baselines for further evaluation.

The baseline architectures are:

- **LSTM DeepLncLoc** is an LSTM-based network fed with DeepLncLoc embedded data, used as a baseline for evaluating the DeepLncLoc embedding method [8].
- **DivideEtImpera** is an experimental model whose aim is to find a more stable Convolutional-based solution, which will be compared directly with Xpresso [11].

Furthermore, in this paper, transcription factors data are integrated with the model, leading to a significant improvement in gene expression prediction.
Transcription factors are proteins that regulate the transcription rate of genetic information from DNA to messenger RNA. Eukaryotic transcription factors work by binding to their target DNA site, located near their target genes, to recruit or block the transcription machinery onto the promoter region of the gene of interest. Their function relies on the ability to find their target site quickly and selectively [16, 17].
The rest of the paper is structured as follows. Data refers to the data used for the training phase of the models and their main characteristics. Afterwards we present the Methods and their Results. At the end, a Discussion and Conclusion part is provided.

Data

The dataset is obtained from the Xpresso paper[11], and contains about 18000 gene sequences with their expression values already processed and easily usable.
Xpresso's authors refers to these gene's sequences with the name of promoters, that are sequences of DNA located upstream the Transcription Start Site (TSS)[14], usually 100–1000 base pairs long, containing specific DNA sequences that provide a secure initial binding site for RNA polymerase and proteins called transcription factors recruiting RNA polymerase. Nevertheless, the actual dataset's sequences contains other DNA regions with respect to the promoters such as the neighborood of the TSS and the codifying part. Indeed, in Xpresso, gene sequences contain 20000bp for each gene (10000bp upstream and 10000 downstream the TSS) and not the promoter part only. Furthermore, Xpresso performs a fine-tuning of the promoter region, identifying 7000bp upstream and 3500bp downstream the TSS as the best interval to predict gene expression.
Xpresso model, in addition to the gene sequences, exploits for each gene some extra information, named mRNA half-life features, to predict the gene expression levels.
The half-life of mRNAs is "the time required for degrading 50% of the existing mRNA molecules" [12]. Knowledge of the half-life of mRNA could potentially provide information about the stability of different types of mRNA.
The half-life of mRNA is challenging to be determined experimentally because an mRNA molecule is short-lived (between 3 and 8 minutes). However, equations describing the decay of mRNA and the growth of cells can be used to estimate the mRNA half-life. Indeed, the information collected in the Xpresso paper refer to 8 values that could explain the variability of mRNA half-lives [11], such as: *coding exon density*, *5' UTR G/C content*, *3' UTR G/C content*, *ORF G/C content*, *5' UTR length*, *3' UTR length*, *ORF length*, *intron length*. In molecular biology and genetics, GC-content (or guanine-cytosine content) is the percentage of nitrogenous bases in a DNA or RNA molecule that are either guanine (G) or cytosine (C)[19]. Within two years of their discovery in 1977, introns were found to affect gene expression positively. Indeed, distributions of the length and matching rate of optimally matched intron segments are consistent with sequence features of miRNA and siRNA[20]. These results indicate that the interaction between intron sequences and mRNA sequences is a kind of functional RNA-RNA

interaction[20]. In molecular genetics, an open reading frame (ORF) is the part of a reading frame that can be translated. An ORF is a continuous stretch of codons that may[21] begin with a start codon (usually AUG) and ends at a stop codon (usually UAA, UAG, or UGA)[22]. In addition, an ATG codon (AUG in terms of RNA) within the ORF (not necessarily the first) may indicate where translation starts. One common use of open reading frames (ORFs) is evidence to assist in gene prediction. Long ORFs are often used, along with other evidence, to initially identify candidate protein-coding regions or functional RNA-coding regions in a DNA sequence [23]. However, the presence of an ORF does not necessarily mean that the region is always translated. For example, in a randomly generated DNA sequence with an equal percentage of each nucleotide, a stop-codon would be expected once every 21 codons [23].

Trascription factors

As previously mentioned, a Transcription Factor (TF) is a protein that controls the rate of transcription of genetic information from DNA to messenger RNA by binding to a specific DNA sequence [16, 17]. Hence, such information is related to the gene expression [18].
Therefore, TFs are exploited in the proposed method and integrated into the DL architecture.
Transcription factors information were retrieved from the ENCODE Transcription Factor Targets dataset [15], which provides the associations between 22449 distinct genes and their related transcription factors. Unfortunately, the ENCODE database contains only 181 distinct transcription factors (concerning the 1600 transcription factors present in the human genome) [29].
In this work, a presence-absence matrix is created leveraging the ENCODE database [15], containing the transcription factor information. An example of the TF matrix is reported in Table 1, where the rows correspond to the 22449 genes and the columns to the 181 transcription factors. The value in position x_{ij} is equal to 1 if gene i is targeted by transcription factor j, 0 otherwise.

Experimental conditions

In this work, three experimental conditions were considered to predict gene expression levels:

- using gene sequence **promoter** information only;

	TF-1	TF-2	TF-3	....	TF-M
GENE-1	0	1	1	....	1
GENE-2	1	1	0	....	1
....	....	....	....	....	.....
GENE-N	0	0	0	....	0

Table 1: TFs table integrated in the proposed model. The rows correspond to the 22449 genes and the columns to the 181 transcription factors. The value in position x_{ij} is equal to 1 if gene i is targeted by transcription factor j, 0 otherwise.

- using gene sequence **promoters** and **half-life features** information;
- using gene sequence **promoters**, **half-life features** and **transcription factors** information.

Hence, the Results section refers to the performances of Transformer DeepLncLoc, LSTM DeepLncLoc, and DivideEtImpera architectures in these three conditions.

Methods

In this section, the base encoding DeepLncLoc is presented. Then, the proposed method Transformer DeepLncLoc and the two baseline architectures are described.

DeepLncLoc

DeepLncLoc [8] is a domain-specific embedding used to synthesize information of long sequences of nucleotides in a compact fashion. It was initially proposed in the paper "DeepLncLoc: a deep learning framework for long non-coding RNA subcellular localization prediction based on subsequence embedding" [8].
The embedding process is defined as follows.

1. The sequences are divided in k-mer[10] of 3 to create a vocabulary. The total number of distinct 3-mer for ATCG nucleotides is 64.
   This vocabulary is then processed by the word2vec algorithm [9] which associates each different 3-mer to a vector of length embedding size. The result is an embedding matrix of shape (3-mer, embedding size), which in our specific case is 64x64. Those values were found via hyperparameter tuning.

2. The data is cleverly reshaped to preserve the order of the sequences. Initially, the data are divided into consecutive slices of an arbitrary length L, which is a hyperparameter: the values $[50, 100, 105, 140, 150, 210]$ were evaluated on the validation set, and the best value resulted in being 210. Hence, the initial sequence of dimension 10500 will be embedded into a matrix of 210 slices, and each slice will be 50 elements long. Afterward, each subsequence is converted in a k-mer form, and for each triplet, the respective embedding vector is associated. We can represent each subsequence with a matrix, concatenating the vectors calculated by word2vec [9] for each triplet.
3. As proposed in the original work, the mean of these vectors is taken to represent each subsequence with a vector. Then, the vectors related to the subsequences are concatenated to obtain the embedding of the whole sequence. The final dimension of the embedding is a matrix 210x64, where 210 is the number of slices and 64 are the features for each slice.

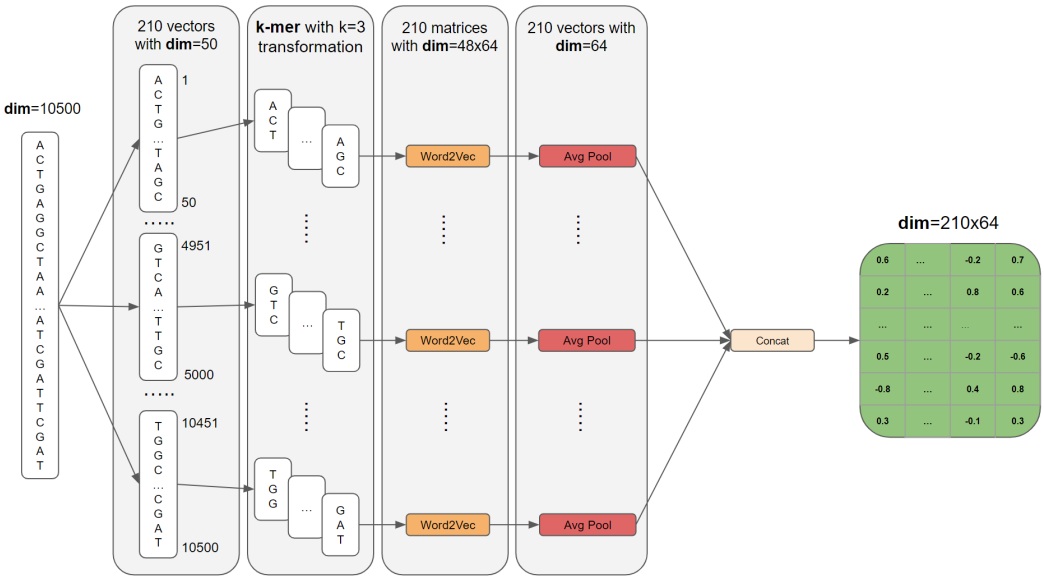


Figure 1: DeepLncLoc

The main advantages of the DeepLncLoc [8] embedding are to avoid the sparse matrix representation (typical of the one-hot encoding) thanks to word2vec[9] and to compress the data exploiting a domain-aware approach

making use of k-mer. In addition, it allows the usage of sequence processing models that would not be possible to exploit on the raw sequences. This dimensionality reduction is crucial in order to train complex many-to-one sequence models. For instance, LSTM-based networks can employ many resources and can be time-consuming: instead, feeding the networks with a reduced feature matrix makes the training phase lighter and faster.

Transformer DeepLncLoc

The Transformer[6] is one of the newest Deep Learning models, state of the art in the field of Natural Language Processing. The main pillars of this architecture are: Embedding of the tokens (word2vec), [9], Positional encoding (sinusoidal functions) [6], MultiHeadedAttention [6]. This paper presents Transformer DeepLncLoc, a transformer-based architecture combining the DeepLncLoc embedding advantages [8] with the transformers' [6] capability in finding complex and long-range dependencies. The transformers [6] build themselves the embedding given the sequences with a word2vec[9] approach, and then they add a positional encoding for keeping track of the position of the words. On the other hand, DeepLncLoc[8] is based on word2vec[9] too, but it is an offline procedure. Therefore, the integration of the DeepLncLoc embedding with the transformer architecture is performed using a BatchNormalization[32] layer just after the input layer to solve numerical issues. Then, the classical transformer's Positional Encoder is applied. Subsequently, there is a Transformer Encoding Block, the One-Dimensional Global Average Pooling, the concatenation with the Half-life features and the last two dense layers to accomplish the regression task. After the hyperparameter tuning, the best optimizer is Adam, and the best loss is mean square error (MSE)[30].

LSTM DeepLncLoc

LSTM DeepLncLoc provides a baseline for the Transformer DeepLncLoc [8] architecture. It consists of a Long Short term Memory (LSTM) [7] based model fed with DeepLncLoc embedded data. Indeed, it is the simplest type of processing that can be applied to a sequence. In detail, the model is composed of a 100 units LSTM[7] feature extractor and fully connected layer in order to allow the regression task. After the hyperparameter tuning, the best optimizer is Adam, and the best loss is MSE[30].

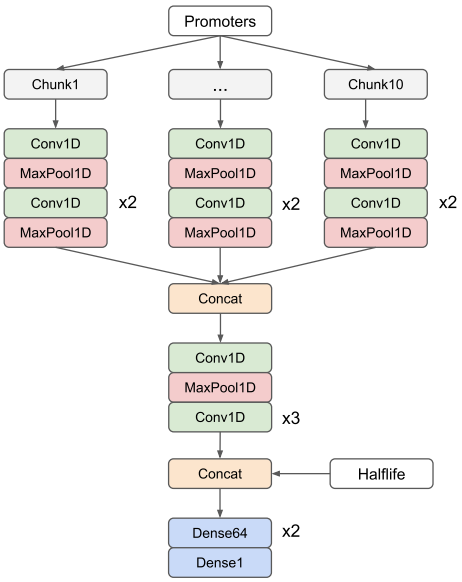


Figure 2: DivideEtImpera architecture.

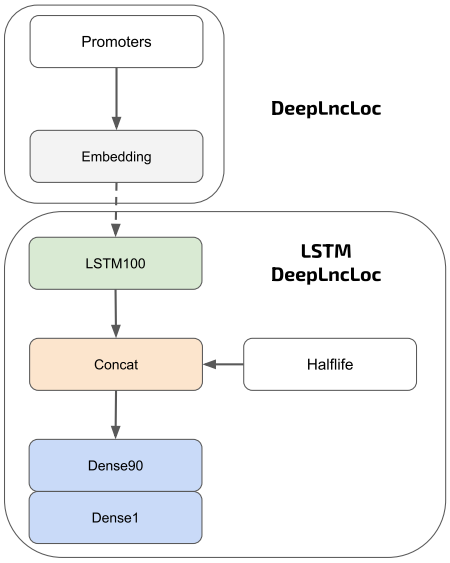


Figure 3: LSTM DeepLncLoc Architecture.

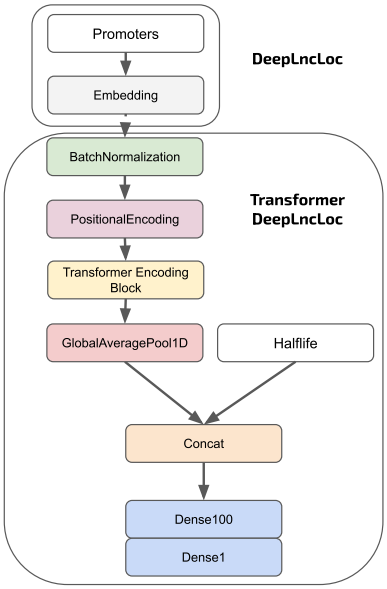


Figure 4: Transformer DeepLncLoc Architecture.

DivideEtImpera

DivideEtImpera is based on classical Conv1D layers, devised to find a more stable convolutional solution. It exploits a chunk separation of the input

sequences (inspired to DeepLncLoc embedding) and a deeper convolutional structure (128 filters per layer, 3 convolutional layers in total for each chunk and 4 after the concatenation) which exploits only kernels of size 3, because on average this allows extracting more complex features rather than a shallow Convolutional Network with a big filter such Xpresso.[13].
The main idea behind this model is to reduce the main problem in subparts, solve them and finally recombine everything and find the solution. We divide our sequence into ten chunks (found by validation), apply five conv/pool layers to each chunk separately, concatenate all the results, and then apply five conv/pool layers again, and the final result is processed by the dense layers [1]. We tried different combinations of depth in every stage of the network, ending up with 128 filters for each convolutional layer and a pool size of 5 for each MaxPooling layer. Finally, we found out that the best optimizer/loss combination is SGD [31] with MSE[30].
All the methods in this work are available on the github page.

Results

In this section, the evaluation of the models is done in three different experimental conditions [Experimental conditions], replicating the same evaluation setting of the *Xpresso* paper for the sake of comparison. The latter consists of using the R^2 metric for evaluation and keeping the best ten runs for each model to build confidence intervals (CIs). In regression, the R^2 coefficient of determination is a statistical measure of how well the regression predictions approximate the real data points, computed as the ratio of the explained variance to the total variance. [35]. By doing so, the stability of each model and the performance can be clearly stated. We clarify the fact that we run the *Xpresso*'s Google Colab notebook to obtain the CIs of the dataset composed by promoter and halflife features, while for the other conditions, we used an adapted version created for the sake of the project. The experiments are grouped and evaluated considering the same input data.
First of all, the models are evaluated only on the promoter gene sequences, and in Table[2] can be seen the CIs.
Afterwards, the experiments are evaluated considering promoter gene sequences and the halflife features.
The models then are finally evaluated considering all the data available: sequences, halflife data and transcription factors.

Model	**LB** (%)	**Mean** (%)	**UB** (%)
Xpresso	0.526	0.531	0.536
LSTM DeepLncLoc	0.574	0.580	0.585
DivideEtImpera	0.529	0.534	0.539
Transformer DeepLncLoc	**0.588**	**0.596**	**0.603**

Table 2: 95% Confidence Intervals based on R^2 scores produced from the best 10 independent trials using only promoters sequences.

Model	**LB** (%)	**Mean** (%)	**UB** (%)
Xpresso	0.559	0.567	0.574
Original Transformer	0.461	0.470	0.479
LSTM DeepLncLoc	0.603	0.606	609
DivideEtImpera	0.580	0.582	0.583
Transformer DeepLncLoc	**0.608**	**0.610**	**0.612**

Table 3: 95% Confidence Intervals based on R^2 scores produced from the best the best 10 independent trials using promoter sequences and halflife features.

Model	**LB** (%)	**Mean** (%)	**UB** (%)
Xpresso	0.742	0.745	0.747
LSTM DeepLncLoc	0.753	0.755	0.758
DivideEtImpera	0.757	0.759	0.760
Transformer DeepLncLoc	**0.756**	**0.760**	**0.764**

Table 4: 95% Confidence Intervals based on R^2 scores produced from the best 10 independent trials using promoter sequences, halflife features and transcription factors data.

Given the evident boost in performances due to the integration of transcription factors, a vanilla Multi-Layer Perceptron has been trained on transcription factors data only to assess their informative power. The evaluation's confidence interval is the following: (0.662, 0.670, 0.677).
In Figure[9] it's possible to have a visual evaluation of the models through the scatter plot. Given the predicted expression level on the x axes ad the median expression level on the y axes, ideally, the closer the points are to the bisector, the better the model's predictive power.

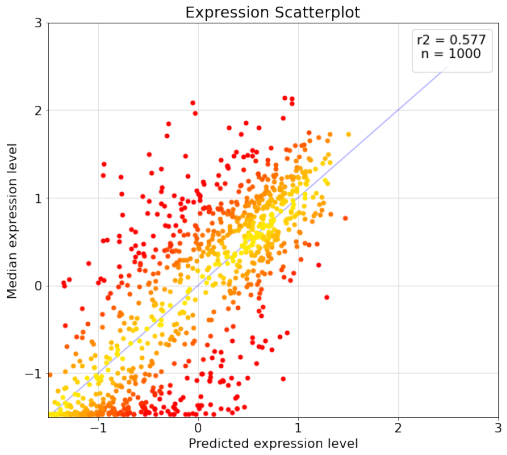


Figure 5: Xpresso

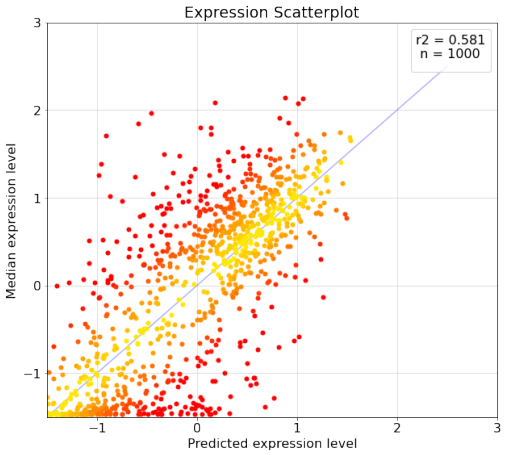


Figure 6: DivideEtImpera

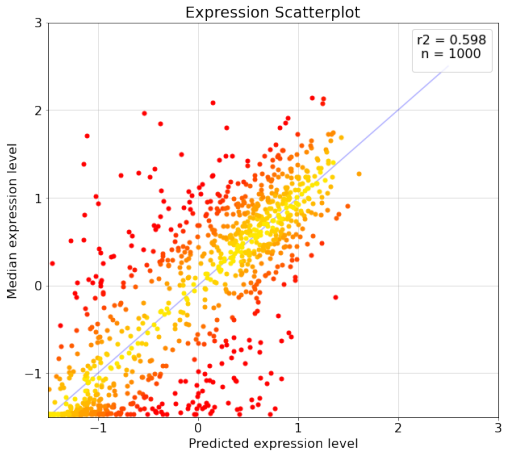


Figure 7: LSTM DeepLncLoc

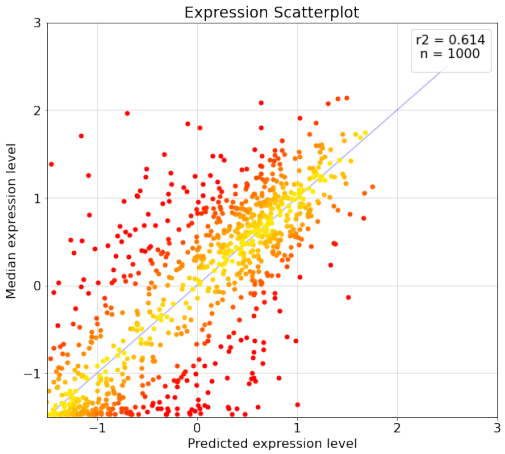


Figure 8: Transformer DeepLncLoc

Figure 9: Expression Scatter Plot

Discussion

In this section, we discuss the results, making comparisons between the models' performances and trying to highlight their strengths and weaknesses. Firstly, by giving a glance at the results tables, we can state that **Transformer DeepLncLoc** outperforms every model considered in every experimental condition (for more, please refer to section Experimental conditions).

This result is because Transformer DeepLncLoc architecture mixes the methods that try to solve the main limitations of the classical approaches. In fact, on the one hand, the **DeepLncLoc**[8] embedding method is capable of modeling and creating a compressed, dense, and domain-aware embedding. On the other hand, the Transformer Encoder block [6] has a great predisposition in long-range modeling dependencies. In addition, we can study and compare the results of **LSTM DeepLncLoc** and **DivideEtImpera** to understand better the effectiveness of the **DeepLncLoc** embedding. First of all, it is essential to remark that the **DeepLncLoc** embedding works offline so that you can compute the embedded version of the data one time, and then you can train the models feeding them directly with the embedded data. Consequently, if the embedding was computed in the right way, you need only an algorithm capable of finding patterns in sequences, like a Recurrent model or an Attention-based one. Indeed, implementing just a classic LSTM-based solution on top of such embedding entails reaching leading performance [7]. On the contrary, *DivideEtImpera* computes the embedding online every time that its training routine is invoked. It has to accomplish two tasks instead of one (computing the embedding and analyzing the resulting sequences), and for this reason, the likelihood of failure increases, resulting in lower performance on average. Similarly, this is the reason why **Transformer DeepLncLoc** shifts the odds of failure just on the Transformer Encoder.
From these considerations, we can conclude that the embedding of the raw sequences is a very crucial part of the process and that **DeepLncLoc** [8] offers an excellent solution to the problem. Nevertheless, a Transformer based solution [6] seems the perfect choice in order to analyze the embedded sequences, overtaking the performances of recurrent-based solutions thanks to the Multi-Headed Attention's capability to model longer relationships.
At this point, some useful considerations can be done about **DivideEtImpera**. As already stated, the logic behind the design of this model is not intended to create a competitive architecture like **Transformer DeepLncLoc**, but to understand how to create stable embedding and feature extraction exploiting solely Convolutional Layers. Indeed, its peak performances are not relevant like our LSTM or Transformer's ones. Nevertheless, it is the second performing model on the dataset integrated with Transcription Factors. Moreover, this framework is more stable and less dependent on the tuning of the hyperparameters with respect to **Xpresso** [11]. This point is achieved thanks to its peculiar deep chunking architecture, and in the end, it can be seen like the Convolutional counterpart of **DeepLncLoc** [8].

The final considerations are related to the integration of the Transcription Factors data. Thanks to the results shown in Table [4], we can state that TF additional data gives a massive boost to all the models. Moreover, it is essential to say that they achieve remarkable results also in a stand-alone evaluation, as stated in the Results section.
The main reason of the results achieved by the TF are due to their main role in the transcription regulation process. Indeed TF can either stimulate or repress transcription of the related gene affecting the gene expression levels [34].

Conclusion

The aim of this paper is to predict the abundance of the mRNA by processing gene promoter sequences, handling the problem as a regression task.
A deep study of the existent models like Xpresso[11] was performed in order to spot their main weak points and to devise new models capable of overcoming their performances.
The main drawback of the presented Convolutional-based solution is the embedding type, usually a one-hot encoding, and the limited receptive field typical of the Convolutional Neural Network. In this paper we used more dense and task-aware embeddings like DeepLncLoc[8] and architectures capable of modeling complex long-range dependencies like Transformers[6]. By analyzing the results, it is possible to understand that the Transformers can generalize better concerning LSTM[7], hence they can reach better results, and this is probably due to the multi-head attention layer that finds more complex patterns for LSTM[7].
From the dataset perspective, the relevant finding is related to the transcription factors capable of giving a massive boost in performance.
A possible future improvement is exhaustive hyperparameter research for models and better integration between the additional data.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

Vittorio Pipoli: Conceptualization, Methodology, Software, Investigation, Data Curation, Writing - Original Draft, Visualization, Validation, Writing -

Review & Editing **Mattia Cappelli:** Conceptualization, Methodology, Software, Investigation, Data Curation, Writing - Original Draft, Visualization, Validation, Writing - Review & Editing **Alessandro Palladini:** Conceptualization, Methodology, Software, Investigation, Data Curation, Writing - Original Draft, Visualization, Validation, Writing - Review & Editing **Carlo Peluso:** Conceptualization, Methodology, Software, Investigation, Data Curation, Writing - Original Draft, Visualization, Validation, Writing - Review & Editing **Marta Lovino:** Conceptualization, Validation, Writing - Review & Editing, Supervision, Project administration, Validation, Investigation **Elisa Ficarra:** Conceptualization, Funding acquisition, Supervision, Project administration, Validation, Investigation

Appendix A. Implementation

All this work and the data are available through the GitHub repository. The implementation has been possible thanks to Google Colaboratory. Furthermore, the project is fully coded in Python and the deep learning framework adopted for the realization of the models is TensorFlow.

References

[1] Zhou J, Theesfeld CL, Yao K, Chen KM, Wong AK, Troyanskaya OG. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. Nat Genet. 2018 Aug;50(8):1171-1179. doi: 10.1038/s41588-018-0160-6. Epub 2018 Jul 16. PMID: 30013180; PMCID: PMC6094955.

[2] David R Kelley , Yakir A Reshef , Maxwell Bileschi , David Belanger , Cory Y McLean , Jasper Snoek. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. Genome Res. 2018;28: 739-750. doi: 10.1101/gr.227819.117

[3] Yilun Zhang, Xin Zhou, Xiaodong Cai Predicting Gene Expression from DNA Sequence using Residual Neural Network doi: https://doi.org/10.1101/2020.06.21.163956

[4] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R. Ledsam, Agnieszka Grabska-Barwinska, Kyle R. Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, David R. Kelley. Effective gene expression

prediction from sequence by integrating long-range interactions. bioRxiv 2021.04.07.438649; doi:https://doi.org/10.1101/2021.04.07.438649

[5] Lonsdale, J., Thomas, J., Salvatore, M. et al. The Genotype-Tissue Expression (GTEx) project. Nat Genet 45, 580–585 (2013). https://doi.org/10.1038/ng.2653

[6] Ashish Vaswani and Noam Shazeer and Niki Parmar and Jakob Uszkoreit and Llion Jones and Aidan N. Gomez and Lukasz Kaiser and Illia Polosukhin. *Attention Is All You Need*. arXiv 1706.03762, 2017.

[7] Hochreiter, Sepp and Schmidhuber, Jürgen. *Long Short-term Memory* PubMed 10.1162/neco.1997.9.8.1735, 1997.

[8] Min Zeng, Yifan Wu, Chengqian Lu, Fuhao Zhang, Fang-Xiang Wu, Min Li. *DeepLncLoc: a deep learning framework for long non-coding RNA subcellular localization prediction based on subsequence embedding* bioRxiv 10.1101/2021.03.13.435245, 2021.

[9] Mikolov, Tomas; et al. *Efficient Estimation of Word Representations in Vector Space*,2013

[10] Chor, Benny; Horn, David; Goldman, Nick; Levy, Yaron; Massingham, Tim *Genomic DNA k-mer spectra: models and modalities*,2009. Genome Biology.

[11] Vikram Agarwal and Jay Shendure. *Predicting mRNA Abundance Directly from Genomic Sequence Using Deep Convolutional Neural Networks* bioRxiv 10.1101/416685v1, 2018.

[12] Chen CY, Ezzeddine N, Shyu AB. *Messenger RNA Half-Life Measurements in Mammalian Cells* doi: 10.1016/S0076-6879(08)02617-7, 2008.

[13] Vikram Agarwal and Jay Shendure. *Predicting mRNA Abundance Directly from Genomic Sequence Using Deep Convolutional Neural Networks* `https://github.com/vagarwal87/Xpresso`

[14] Christof Angermueller,Tanel Pärnamaa,Leopold Parts,Oliver Stegle refTSS: A Reference Data Set for Human and Mouse Transcription Start Sites. Journal of Molecular Biology, 2019

[15] *ENCODE Transcription Factor Targets dataset*
`https://maayanlab.cloud/Harmonizome/dataset/ENCODE+Transcription+Factor+Targets`

[16] Latchman DS (December 1997). "Transcription factors: an overview". The International Journal of Biochemistry & Cell Biology. 29 (12): 1305–12. ScienceDirect - PMC - PMID

[17] Karin M (February 1990). "Too many transcription factors: positive and negative interactions". The New Biologist. 2 (2): 126–31. PMID

[18] Magnusson R et al. *White-box Deep Neural Network Prediction of Genome-Wide Transcriptome Signatures*

[19] Definition of GC – content on CancerWeb of Newcastle University,UK

[20] Qiang Zhang, Hong Li, Xiao-qing Zhao, Hui Xue, Yan Zheng, Hu Meng, Yun Jia, Su-ling Bo, The evolution mechanism of intron length, Genomics, Volume 108, Issue 2, 2016, Pages 47-55, ISSN 0888-7543, doi. (ScienceDirect)

[21] Sieber, Patricia; Platzer, Matthias; Schuster, Stefan (March 2018). "The Definition of Open Reading Frame Revisited". Trends in Genetics. 34 (3): 167–170. PMID.

[22] Brody, Lawrence C. (2021-08-25). "Stop Codon". National Human Genome Research Institute. National Institutes of Health. Retrieved 2021-08-25.

[23] Slonczewski, Joan; John Watkins Foster (2009). Microbiology: An Evolving Science. New York: W.W. Norton Co.

[24] Lonsdale, J. et al. *The genotype-tissue expression (GTEx) project. Nature genetics 45, 580(2013).*

[25] Eraslan, G., Avsec, Ž., Gagneur, J. Teis, F. J. Deep learning: new computational modelling techniques for genomics. Nat. Rev. Genet. 20, 389–403 (2019).

[26] Christof Angermueller,Tanel Pärnamaa,Leopold Parts,Oliver Stegle Deep learning for computational biology. Molecular Systems Biology, 2016

[27] Chao Cheng, Roger Alexander, Renqiang Min, Jing Leng, Kevin Y. Yip, Joel Rozowsky, Koon-Kiu Yan, Xianjun Dong, Sarah Djebali, Yijun Ruan, Carrie A. Davi7, Piero Carninci, Timo Lassman, Thomas R. Gingeras, Roderic Guigó, Ewan Birney, Zhiping Weng, Michael Snyder and Mark Gerstein Understanding transcriptional regulation by integrative analysis of transcription factor binding data. Genome Res. 2012

[28] Jorge S. Reis-Filho Next-generation sequencing, in Breast Cancer Research,vol. 11, n. 3, 1º Jan. 2009

[29] Babu MM, Luscombe NM, Aravind L, Gerstein M, Teichmann SA (June 2004). "Structure and evolution of transcriptional regulatory networks" (PDF). Current Opinion in Structural Biology. 14 (3): 283–91. doi:10.1016/j.sbi.2004.05.004. PMID 15193307.

[30] Sammut, Claude and Webb, Geoffrey I. *Mean Squared Error - Encyclopedia of Machine Learning* 2010, Springer US, Boston (MA), isbn= 978-0-387-30164-8, doi= 10.1007/978-0-387-30164-8_528, https://doi.org/10.1007/978-0-387-30164-8_528

[31] Robbins, Herbert E.. "A Stochastic Approximation Method." Annals of Mathematical Statistics 22 (2007): 400-407.

[32] Ioffe, Sergey; Szegedy, Christian *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, 2015

[33] *Convolutional Neural Networks (LeNet) – DeepLearning 0.1 documentation*

[34] Definition of Transcription Factors - Scitable, Nature Education

[35] Steel, R. G. D.; Torrie, J. H. *Principles and Procedures of Statistics with Special Reference to the Biological Sciences, 1960, McGraw Hill*