DNAPerceiver and ProteinPerceiver: deep learning models for gene and protein expression prediction from DNA and amino-acid sequences

Matteo Stefanini[a], Marta Lovino[a,*], Rita Cucchiara[a], Elisa Ficarra[a]

[a] *University of Modena and Reggio Emilia, Via Vivarelli 10/1, 41125 Modena, Italy*

Abstract

The functions of an organism and its biological processes derive from the expression and activity of genes and proteins. Therefore quantifying and predicting gene and protein expression values is a crucial aspect of scientific research. Concerning the prediction of gene expression values, the available machine learning-based approaches use the gene sequence as inputs to the neural network models. Some techniques, including Xpresso and Basenjii, have been proposed to predict gene expression values in the samples. However, these architectures are mainly based on Convolutional or Long Short Term Memory Networks, neglecting the attention mechanisms impact in selecting the sequence's relevant portions for prediction purposes. In addition, as far as we know, there is no model for predicting protein expression values exploiting the sequence of genes (sequence of nitrogenous bases) or proteins (sequence of amino acids). Here, we present a new Perceiver-type model, which exploits a transformer-based architecture to profit from the attention module and overcome the quadratic complexity of the standard attention-based architectures. The contributions of this work are the following: 1. Development of the DNAperceiver model for improving the prediction of gene expression starting from the genes' sequence; 2. Development of a ProteinPerceiver model for predicting protein expression values starting from the amino acid sequence; 3. Development of a Protein&DNAPerceiver

*Corresponding author
Email address: marta.lovino@unimore.it (Marta Lovino)

model for predicting protein expression values by simultaneously combining the sequence of nitrogenous bases and amino acids. 4. Evaluation of models under multiple conditions: predicting values on cell lines, mice, and tumor tissues of glioblastoma and lung cancer. The results show the effectiveness of the Perceiver-type models in predicting gene and protein expression values starting from the gene/protein sequence. In this context, inserting further regulatory and epigenetic information into the model could improve the prediction task.

The source code is freely available at `https://github.com/MatteoStefanini/DNAPerceiver`.

Keywords: Deep Learning, DL, DNA, gene expression, genes, Perceiver, protein, protein expression, Proteomics, sequences

---

Background

Most of the biological processes that regulate the functions of an organism are due to the activity of proteins [9, 10, 21]. In recent decades, the incredible development of sequencing techniques and proteomics quantifications have enabled a systematic analysis of the activity level of thousands of mRNA molecules (the precursor of proteins) and proteins simultaneously [17, 24]. It is known that many regulatory and epigenomic processes regulate the expression values of mRNAs and proteins [4, 11, 12], and the sequence of nucleotides has long been investigated to predict the mRNA expression in various samples and tissues. In contrast, the protein expression prediction from nucleotides or amino acids has not yet been addressed in the literature.

In recent years, multiple deep learning techniques have been presented for health applications[2, 5, 6, 16]. In this context, previous works focused on gene expression level prediction from nucleotide sequences exploiting deep learning techniques. In particular, various types of Convolutional Neural Networks have been adopted to deal with the sequential nature of the DNA[1, 3, 15, 18, 26].

Specifically, Basenjii[15] applies convolutional layers followed by dilated convolutions to share information across large distances. Dilated convolutions are

convolutions with a wider filter created by regularly inserting spaces between the filter elements. Those gaps exponentially increase the receptive field width, thus taking into account potentially more regulatory regions for each gene.

Similarly, Expecto[26] applies convolutional layers to extract features from genomic sequences using a predefined window's size. Each window yields a set of features stacked together in a high dimensionality feature vector. Spatial transformations are then applied before the gene expression levels are produced to reduce this dimensionality.

On the same line, Xpresso[1] introduced a deep convolutional model composed of two sequential convolutional and max-pooling layers followed by two fully connected layers preceding the output prediction, demonstrating that the localized region around the promoter (i.e., upstream gene region) captures the most learnable information.

Although convolutions represent an effective way to deal with nucleotide sequences, they have some significant limitations that hinder their representational power. Above all, the locality nature of convolutions limits the information propagation in the network among distal elements, requiring many successive layers to expand the receptive field and thus not allowing to capture long-term relationships and dependencies between sequence elements[20].

On the other hand, the attention mechanism has achieved outstanding performance in various applications due to each token's global interaction in the input sequence, extracting global information from the first layer. These global interactions bring a quadratic complexity of attention operator $O(n^2)$, making it unfeasible to apply directly to long sequences of raw data.

The Enformer model[3] firstly applies self-attention to genomic data, capturing wide-ranging relationships and improving gene expression prediction. However, to keep the computation feasible, the model is composed of a first convolutional pipeline that extracts local features, that are then applied to self-attention layers in order to capture long-range interactions.

Our method, instead, is based on the Perceiver architecture[14], which allows for asymmetric attention between inputs and learnable query vectors, therefore

expanding its capabilities to attend longer sequences directly on the raw data without requiring a first feature extraction pipeline. Nonetheless, we take inspiration from the Enformer model to recreate a query pipeline for our Perceiver architecture.

In this work, we present three models, all based on the Perceiver architecture: DNAPerceiver, ProteinPerceiver, DNA&ProteinPerceiver. DNAPerceiver predicts the gene and protein expression value starting from the nucleotide sequence of the genes, and its performances are directly compared with competitor models on various datasets. ProteinPerceiver and DNA&ProteinPerceiver instead predict the protein expression values starting respectively from the sequence of amino acids (ProteinPerceiver) and the combination of nucleotides and amino acids sequences (DNA&ProteinPerceiver). The latter two models were evaluated under different experimental conditions. However, due to the task's novelty, it is impossible to report comparisons with models in the literature. Below, the Materials and Method section contains the technical details of the models developed and the data used. Subsequently, the Results and Discussion sections report the results obtained. Finally, the Conclusion section summarizes the main contributions of this work.

Material and methods

In order to evaluate the relationship between the sequence of genes or proteins and their gene and protein expression values, the coding genes (those genes annotated as protein-coding) were selected, and their nucleotide and amino acid sequences were obtained (see the details in the Dataset section). Then, three models were implemented to evaluate the predictive impact of nucleotide and amino acid sequences on the expression values of genes and proteins: DNAPerceiver, ProteinPerceiver, and DNA&ProteinPerceiver, all three based on a Perceiver architecture. The purpose of the DNAPerceiver is to predict gene expression values starting from the gene sequence alone. ProteinPerceiver and DNA&ProteinPerceiver instead predict protein expression values starting re-

spectively from the sequence of proteins and the combination of the sequence of proteins and genes. Although each model differs in the prediction task, the general structure is very similar. First, each model receives as input a sequence of letters representing the gene's nucleotide sequence or the protein's amino acid sequence. Then, the sequence of letters is processed by the Perceiver model, and a discrete number representing the protein or gene expression value is outputted for each sequence. This number represents the average amount of a gene or protein present in the samples. The greater the number, the greater the amount of molecule (mRNA or protein) circulating. Therefore the main difference between the three Perceiver architectures consists of the input data: DNA sequences for DNAperceiver; protein sequences for ProteinPerceiver, and DNA and protein sequences for DNA&ProteinPerceiver architecture. The dataset and DNAPerceiver, ProteinPerceiver, and DNA&ProteinPerceiver architectures are presented.

Datasets

We use different datasets to evaluate our models, depending on the desired predicted output and the input sequences. There are two input types: inputDNA and inputPROT. InputDNA consists of the nucleotide sequence of human protein-coding genes. In particular, the sequence refers to the region straddling the beginning of a gene (also known as the transcription start site - TSS). In this way, the model learns from the promoter region upstream of the TSS on which most of the regulatory processes act. The DNA sequences are taken from the Xpresso dataset [1] due to its particular data curation. Indeed, it is built from the human GRCh38 reference genome, and, in addition, sequences are accurately centered on the TSS exploiting CAGE experiments. Specifically, it comprises 18.377 human genes split into 16.377 genes for train, 1.000 for validation, and 1.000 for the test. The maximum length of a gene is set to 20.000 base pairs. Xpresso input also comes with half-life features, a set of characteristics that describe general information about the gene (e.g., gene length, number of introns). Therefore, whenever we use DNA sequences, we also

include half-life features as additional input to our models at different network points, as explained in the architecture section.

InputPROT, on the other hand, consists of the amino acid sequences of the proteins. Therefore, the promoter region is not included in the inputPROT sequence. All amino acid sequences were obtained from Uniprot database [8], processed with Biopython library [7] and intersected with Xpresso's list of protein-coding genes.

As for the labels, we used four conditions for the prediction of gene expression (labelExprMouse, labelExprHuman, labelGeneGlio, labelGeneLung) and two conditions for the prediction of protein expression (labelProtGlio and labelProtLung). LabelExprMouse and labelExprHuman come from the Xpresso publication, containing the mean gene expression values of some mouse and human samples, respectively. These labels were obtained in the biologically controlled context of cell lines, and therefore the prediction task is limited. To evaluate the predictive capabilities of the models on high throughput multi-omics human data coming from real clinical studies, we selected gene and protein expression data on patients with glioblastoma [22] and lung cancer [19]. LabelGeneGlio and labelGeneLung contain the labels of the mediated gene expression values for glioblastoma and lung cancer, respectively. The same procedure has been applied to obtain the mediated protein expression values for the same patients, named LabelProtGlio and labelProtLung, respectively [19, 22].

Given the scarcity of data, in all of our experiments, except for Xpresso comparisons, we adopt the K-Fold validation setting and average the results across the folds. We set the number of folds K to 10.

DNAPerceiver architecture

As stated above, various models in the literature focused on predicting gene expression starting from the gene sequence. This work aims to reveal how much gene expression values can be explained by the DNA sequence alone. All predictive models do not use the whole nucleotide sequence of the gene as input but only a portion straddling TSS, which involves numerous regulatory and tran-

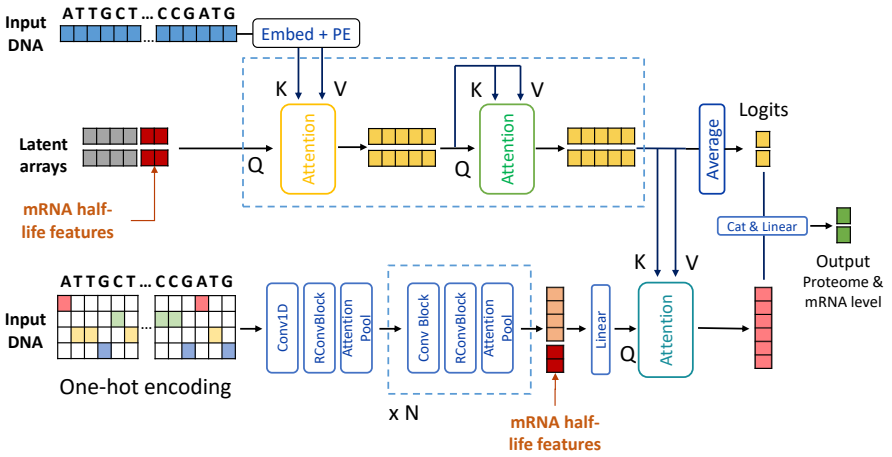

Figure 1: DNAPerceiver architecture. It is based on the Perceiver IO model [13]. The upper flow represents the asymmetric attention that distills the DNA sequence in a smaller latent space, where learnable arrays attend to all the input sequences and refine their representations with self-attention and feed-forward networks. The lower flow depicts the decoding stage of the Perceiver IO, where instead of using learnable vectors like in the original model, we use, as the final query, the same DNA sequence processed by a convolutional pipeline inspired by the Enformer model [3].

scriptional processes. In particular, the region preceding the TSS contains the promoter, a region targeted explicitly by transcription factors, elements responsible for the final quantity of molecule produced. The data used in this model are inputDNA and labelExprMouse, labelExprHuman, labelGeneGlio and labelGeneLung as output.

Figure 1 shows the architecture of the DNAPerceiver. The model is composed of two distinct flows: one with asymmetric attention as in the original Perceiver model [14], and another with a convolutional pipeline inspired by the Enformer model [3]. The asymmetric attention reduces the complexity of the attention from $O(n^2)$ to $O(n \times m)$ where n is the length of the input sequence, and m is a hyperparameter defying the latent space dimensionality. The model can attend to long sequences and condense their semantic information within a tight latent space. The convolutional pipeline extracts another representation

of the same DNA sequence and is then used to query the latent space in the final decoding stage.

Therefore, while our model still leverages convolutional pipelines, it takes more advantage of the recent advancements of attentive architectures, *i.e.* the Perceiver, that have originated from the original Transformer model. Transformers are a class of deep learning models, first introduced by Vaswani *et al.* [20], that attained substantial breakthroughs in natural language processing and computer vision. Specifically, they consist of attention blocks that aggregate information from the entire input sequence by computing a weighted sum across the representations of all other tokens for each token of the sequence. Since each token directly attends to all other positions in the sequence, they allow for a much better information flow between distal elements, in contrast with convolutional layers, which may require many successive layers to increase the receptive field.

These methods were also recently applied to model DNA sequences. However, given the quadratic complexity of attention $O(n^2)$, the length of the input can easily explode quadratically, rendering it infeasible to directly encode an unprocessed DNA input sequence or more than a few thousand length sequences. For this reason, our method is based on the Perceiver [14], a model that builds upon Transformers but scales to hundreds of thousands of inputs, as it leverages an asymmetric attention mechanism to iteratively distill inputs into a tight latent bottleneck, allowing it to deal with very large inputs. These latent arrays go through self-attention blocks to refine their representation and potentially other asymmetric attention layers before getting averaged to obtain the logits for the task at hand.

Specifically, we use the Perceiver IO [13], which improved the decoder capabilities of the model by adding a final decoding stage. This aspect acts as a query on the latent arrays, allowing the model to produce outputs of arbitrary size and semantics, and deal with diverse domains without sacrificing the benefits of deep, domain-agnostic processing.

In our implementation, however, we introduce substantial modifications concerning the Perceiver IO architecture. Firstly, instead of learning a different set of output arrays for the decoding stage, we use the same DNA sequence after being processed by a typical Convolutional pipeline. This pipeline consists of multiple Conv layers, Residual connections, and Attention Pooling layers, inspired by the Enformer model [3]. Secondly, another difference is that our model in the decoding stage also considers the processed latent arrays by applying a final head that computes their average and uses them as final logits. The processing is similar to that of the original Perceiver model. However, in our case, it is fused with the final decoding mechanism proposed by the Perceiver IO.

Hence, in our architecture, the DNA sequence or the protein sequence given in input are processed twofold: as learnable vectors for the perceiver flow, where the asymmetric attention is applied with the latent arrays, and as one-hot encoding vectors fed to the convolutional pipeline. To deal with positions in the asymmetric attention, after embedding the input letters, we also add a learnable Positional Encoding, initialized with a sinusoidal function as in the original Transformer model.

Moreover, mRNA half-life features are injected in both flows: they are appended to the latent arrays and the convolutional pipeline in the final feature representations.

In the most straightforward DNAPerceiver configuration, we predict the gene expression level for both human cell-line and mouse data, and we evaluate our model on the Xpresso dataset, comparing it with other similar methods and Xpresso itself. Further, we applied the DNAPerceiver model to predict mRNA and protein expression in human high throughput sequencing data. As discussed in the Dataset section, we take the proteomic labels from two real-human datasets, ending with 10.529 pairs of gene labels for Lung cancer (labelProtLung) and 10.280 pairs of gene labels for glioblastoma (labelProtGlio). In this configuration, we output two predictions for labelProt and labelGene, protein expression and gene expression, respectively, for both datasets.

To represent the nucleotides letters, A T C G, we use one-hot vectors, and

for the perceiver flow, we linearly project them to learnable vectors of 32 dimensionalities. In addition, we add the letter P as padding. To represent letter positions, we employ learnable positional encodings initialized in a standard sinusoidal fashion[20]. We use 128 latent learnable arrays with a dimensionality of 128 each, constituting the dimensionality of the following self-attention layers. The number of heads in asynchronous attention is set to 1, while self-attention is set to 8. The attention over the input is computed by considering only valid letters and masking the rest. Feed-forward layers have a dimensionality of 256 and GELU nonlinearity. The depth of the perceiver, the number of layers of asynchronous attention followed by self-attention, is set to 1. For the convolutional query flow, we adopt a similar strategy as Enformer[3] using the first layer of Conv1D with a kernel size of 15, channel dimensionality of 64 and Attention Pooling with a pooling size of 12. Subsequent convolutional layers, forming the Conv tower, have a kernel size of 5 and attention pooling size of 6. Each Conv layer applies GELU nonlinearity and is followed by a Residual connection. The length of the input DNA sequence is set to 10.500, taking the majority part from the promoter side and less from the actual gene, specifically considering 7.000 base pairs before the TSS and 3.500 after the TSS. We apply dropout throughout the model before each linear projection and attention layer, with a keep probability of 0.8. We train our model using ADAM optimizer[25], a batch size of 128, and we follow the learning rate scheduling strategy of [20] with a warmup equal to 8.000 iterations. We apply a weight decay of 0.2 and an early stopping strategy to avoid overfitting. We found it helpful to use the Tanh activation for our final predicted scores only in this configuration and when applied to Xpresso gene expression labels. In the end, we weighted the loss contribution using a weight of 10 for the mRNA.

ProteinPerceiver architecture

Although the gene expression prediction task is debated within the scientific community, to date, no model predicts protein expression. This choice is because the quantification data of gene expression have been available in

the literature in large quantities in the last decade. Instead, extracting and quantifying proteins simultaneously in a sample is more recent and less mature than mRNA extraction and quantification techniques. The quantification of proteins is what scientists are most interested in biologically. However, these techniques are currently more expensive, limiting the available data. Moreover, the mRNA quantification experiments can simultaneously evaluate more than 20000 protein-coding genes versus approximately 2 to 8 thousand proteins for protein quantification. The ProteinPerceiver model aims to measure how much protein expression depends on a single amino acid sequence. Unfortunately, given the experiments' novelty, no comparison model in the literature is available. The input consists of inputProtein and labelProtGlio and labelProtLung as output.

We match protein sequences and proteomics labels available, assembling a total of 10.430 amino acid sequences for Glioblastoma and 10.699 for Lung Carcinoma with corresponding proteomic labels.

Differently from the DNAPerceiver setting, here we only adjust the dropout keep probability to 0.7 and the attention pooling size in the convolutional query to 10 in the first layer and 5 in the following ones. Moreover, we set the maximum length of the protein sequence to 6.000 and the final weight of the MSE loss to 100 for Lung data and 3.000 for Glioblastoma data. We optimize the model using Lamb[23], a learning rate of 0.0005 and Cosine Annealing schedule strategy with 8000 steps of warmup.

Protein&DNAPerceiver architecture

The ProteinPerceiver model receives as input the amino acid sequence of the proteins in order to predict protein expression values. However, the protein expression value is determined by the sequence of its amino acids and by regulatory, transcriptional, and epigenetic factors. Although considering all regulatory processes is not straightforward, in this paper, we have evaluated the combined effect of the amino acid sequence and the promoter sequence to predict the protein expression value. The model simultaneously uses inputDNA

and inputProtein and outputs labelProtGlio and labelProtLung. DNA and protein sequences are matched together when both are available from the Xpresso dataset[1] and the protein sequence dataset, ending up with a total of 9.815 triplets gene-protein-labels for Lung Carcinoma and 9.534 triplets for Glioblastoma.

Explicitly, our model deals with two different input sequences, one for the Perceiver flow and one for the Convolutional query. In addition, we investigated the use of the input in an alternate manner: when the DNA sequence is in input to the Perceiver, we use the protein sequence as a query, and vice versa, with protein as the perceiver input, we use the DNA for the query computation. The Results section shows that the best version differs depending on the data and the prediction at hand. Moreover, in this configuration, the maximum length of the protein sequence is set to 6.000, while the DNA sequence length is set to 8.000. If not specified, we kept the same hyperparameters of the DNAPerceiver configuration.

A summary of the architecture names, prediction tasks, input and outputs is reported in Table 1.

Model	Tasks	Input	Output
DNAPerceiver	gene expr	DNA	labelGene
DNAPerceiver	gene&proteome expr	DNA	labelGene&labelProt
ProteinPerceiver	proteome expr	Prot	labelProt
DNA&ProteinPerceiver	proteome expr	DNA&Prot	labelProt

Table 1: Summary of the different configurations of our model depending on the prediction task and the input-output setting.

Results

This section discusses the results obtained and the comparison with the state-of-the-art approaches.

Results on gene expression prediction using Xpresso's labels

In this setting, DNAPerceiver was trained on the Xpresso sequences and its labels, aiming to predict the gene expression level, both from mouse and human organisms (labelExprMouse and labelExprHuman). We follow the split of the original dataset[1] thus obtaining 16.377 genes for training and 1.000 genes for both validation and test set. As shown in Table 2, DNAPerceiver performs better than Xpresso method in terms of r^2 in human and mouse data. In human cell-line data, it reaches an r^2 of 0.62, which, compared to the 0.59 of the Xpresso model, attains a 5% improvement. The basenji method has a similar gene expression prediction task to the one presented in this work. However, a direct comparison cannot be made as Basenji uses Cap Analysis Gene Expression (CAGE) input data which are not available for our dataset. However, under his experimental conditions, Basenji reaches a Pearson correlation coefficient ranging from 0.138 to 0.777 depending on the genes considered. These values would translate into a coefficient of determination r^2 between 0.019 and 0.604. In this context, the DNAPerceiver model seems to get consistent results.

Model	mRNA r^2
Xpresso [1] human data	0.59
Xpresso [1] mouse data	0.71
DNAPerceiver human data	0.62
DNAPerceiver mouse data	0.72

Table 2: Results on the test set of Xpresso dataset in predicting gene level expression of cell-line data.

Results on gene and protein expressions

In this configuration, DNAPerceiver is trained on the inputDNA (Xpresso sequences) and predicts the labels from the Lung and Glioblastoma datasets for gene and protein expression (labelGeneLung, labelGeneGlio, labelProtLung and labelProtGlio) Table3 reports the results. High throughput sequencing data

from human explanted tissues is much more complex than data obtained from cell lines. Indeed, the cell lines are systematically obtained in the laboratory to have a controlled context and genetic variability as small as possible between the cells. By contrast, the sequencing data from explanted tissues (carcinogenic, too) has a high genetic variability as a multiplicity of regulatory factors between cells and tissues are present. Given the noisy nature of high throughput sequencing data, the gene expression prediction is not comparable to that of a cell line culture, but it reaches 0.181 of r^2. Furthermore, our focus is to predict the protein expression level using only the DNA sequence. Our model can predict the protein expression achieving 0.161 of r^2, demonstrating its capability to perceive the direct connection between the DNA sequence and its corresponding protein expression.

Model	mRNA r^2	proteomics r^2
DNAPerceiver Lung	0.181	0.161
DNAPerceiver Glioblastoma	0.150	0.026

Table 3: Results on Lung and Glioblastoma data in predicting gene level and proteome expression. The input gene sequences is inputDNA, taken from Xpresso[1], while predicted labels for mRNA and protein expression are labelGeneLung, labelGeneGlio, labelProtLung and labelProtGlio. Results are the average of the k-fold validation method with k equal to 10.

Results on protein expression using only protein sequence as input

Table4 reports the result of our model applied only to protein sequence as input. In this configuration of our model, named ProteinPerceiver, the input consists of the amino acid sequences, as explained in the Datasets subsection in the Material and methods Section. Our scope is to predict the protein level expression given the amino acid sequence, a task peculiarly complex, as it will be discussed later in the next section. The obtained outcome varies depending on the data: for Lung data, we found that predicting protein expression from the amino acid sequence is more complex, achieving a r^2 of 0.085, comparing the

0.161 obtained from the DNA sequence. Nonetheless, for Glioblastoma data, our ProteinPerceiver can score a r^2 of 0.028 for protein expression, which is slightly better compared to 0.026 obtained by the DNAPerceiver. Despite the impact of data quality and prediction task complexity on the results, our model can still capture a part of the relationship between the protein sequence and its corresponding expression level.

Model	proteomics r^2
ProteinPerceiver Lung	0.085
ProteinPerceiver Glioblastoma	0.028

Table 4: Results in predicting protein expression only from the protein sequence. The input of amino acids sequence is InputPROT, while while predicted labels for protein expression are labelProtLung and labelProtGlio. Results are the average of the k-fold validation method with k equal to 10.

Results on protein expression using DNA and protein sequence as inputs

We wanted to investigate further the capabilities of our model with a peculiar configuration, in which we give as input both the DNA sequence and the protein sequence. Our model's design facilitates this accomplishment, already treating the input in a double structure: one for the perceiver flow and one for the convolutional query. Therefore, we tried the possible configuration in this arrangement, providing the model with the protein sequence as input to the perceiver and the DNA sequence to the convolutional query and vice-versa. We report the results in Table5. In this configuration, performances also depend on the specific data: for Lung data, surprisingly, the use of both inputs does not improve the total performances of the model, reaching 0.141 of r^2 compared to the 0.161 obtained using only DNA sequence. On the contrary, using both inputs slightly improves the results on Glioblastoma data, achieving 0.031 of r^2.

Model	proteomics r^2
Protein&DNA Perceiver Lung	0.141
Protein&DNA Perceiver Glioblastoma	0.031

Table 5: Results in predicting proteome expression from both the DNA sequence and the protein sequence used as inputs. The input of amino acids sequence is InputPROT, the DNA sequence is inputDNA, and the predicted labels for protein expression are labelProtLung and labelProtGlio. Results are the average of the k-fold validation method with k equal to 10.

Discussion

The main advantage of the Perceiver architecture compared to the other standard techniques in the literature consist on the asymmetrical attention mechanism, which reduces the complexity of the attentive operator for the same sequence length input. As regards predicting gene expression values starting from the DNA sequence straddling the TSS (thus including part of the promoter and part of the gene), DNAPerceiver shows results superior to Xpresso in the case of the human cell lines and murine samples. Moreover, as can be expected, the prediction of expression values in cell lines achieves better results than the prediction of the expression values in human tumor samples. This aspect can be explained by the different boundary conditions of the two situations. In the first case, the expression of the genes is suitably controlled to ensure reproducibility and stability of the cell lines. In contrast, in the second case, the biological context is profoundly altered by pathological conditions that can differently impact the samples. Besides the improvement in gene expression prediction, the main novelty of this work is the prediction of protein expression values starting from the sequence of genes and proteins. This aspect is doubly challenging: 1. Protein extraction and quantification techniques have developed significantly in recent years; 2. The amino acid sequence lacks specific target regions for regulatory factors, except for post-transcriptional phenomena. In this context, it is noted that, for the same samples, the prediction of protein expression is considerably lower than the prediction of gene expression, whether the prediction takes place

using the nucleotide or amino acid sequence. The complexity of the problem can explain this phenomenon. The expression of proteins is influenced by notable post-transcriptional and post-translational regulatory phenomena (e.g., ubiquitination). Moreover, the nucleotide sequences (therefore composed of a part of the promoter and a part of the gene) have a greater predictive power of the protein expression values than the amino acid sequences. This behavior could depend on the presence of the promoter. Indeed, the promoter is the gene region that favors the regulation of expression (both of genes and therefore of proteins) and is responsible for interaction with transcription factors. When the model is trained by simultaneously joining the nucleotide sequence with the amino acid sequence, the predictive capacity of protein expression increases; however, it remains lower than the prediction of protein expression using only the nucleotide sequence. In this sense, the nucleotide sequence seems more informative than the amino acid one.

Conclusions

Various papers have addressed gene expression prediction in the literature, which mainly includes convolutional or long short-term memory networks. This work presents three Perceiver-type architectures for predicting gene expression values on cell lines and high-throughput human samples. Furthermore, a novel task is introduced, presenting the prediction of protein expression values starting from the nucleotide and amino acid sequences. The results show the advantages of the Perceiver architecture in predicting gene expression compared to competitors. Protein expression, on the other hand, benefits more from the nucleotide sequence than the amino acid sequences. This aspect seems to be explained by the presence of the promoter region within the nucleotide sequence. Although various experimental conditions have been considered, numerous other biological factors can be included in the models to improve prediction.

Funding

This study was funded by the European Union's Horizon 2020 research and innovation programme DECIDER under Grant Agreement 965193, and partially supported by AI4CHSites CNR4C program (CUP B15J19001040004), by AI4Media under GA 951911, by the "Artificial Intelligence for Cultural Heritage (AI4CH)" project, co-funded by the Italian Ministry of Foreign Affairs and International Cooperation.

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), p. 107663.

[2] Stefano Allegretti et al. "Supporting Skin Lesion Diagnosis with Content-Based Image Retrieval". In: *2020 25th International Conference on Pattern Recognition (ICPR)* (Milan, Italy). IEEE, Jan. 2021, pp. 8053–8060. ISBN: 978-1-7281-8808-9. DOI: 10.1109/ICPR48806.2021.9412419.

[3] Žiga Avsec et al. "Effective gene expression prediction from sequence by integrating long-range interactions". In: *Nature methods* 18.10 (2021), pp. 1196–1203.

[4] Adrian Bird. "Perceptions of epigenetics". In: *Nature* 447.7143 (2007), p. 396.

[5] Federico Bolelli, Stefano Allegretti, and Costantino Grana. "One DAG to Rule Them All". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.7 (Jan. 2021), pp. 3647–3658. ISSN: 0162-8828. DOI: https://doi.org/10.1109/TPAMI.2021.3055337.

[6] Michele Cancilla et al. "The DeepHealth Toolkit: A Unified Framework to Boost Biomedical Applications". In: *2020 25th International Conference on Pattern Recognition (ICPR)* (Milan, Italy). IEEE, Jan. 2021.

[7] Peter JA Cock et al. "Biopython: freely available Python tools for computational molecular biology and bioinformatics". In: *Bioinformatics* 25.11 (2009), pp. 1422–1423.

[8] UniProt Consortium. "UniProt: a worldwide hub of protein knowledge". In: *Nucleic acids research* 47.D1 (2019), pp. D506–D515.

[9] Francis Crick. "Central dogma of molecular biology". In: *Nature* 227.5258 (1970), pp. 561–563.

[10] Francis Crick et al. "General nature of the genetic code for proteins". In: (1961).

[11] Manel Esteller. "Epigenetics in cancer". In: *New England Journal of Medicine* 358.11 (2008), pp. 1148–1159.

[12] Eva Jablonka and Marion J Lamb. "The changing concept of epigenetics". In: *Annals of the New York Academy of Sciences* 981.1 (2002), pp. 82–96.

[13] Andrew Jaegle et al. "Perceiver io: A general architecture for structured inputs & outputs". In: *arXiv preprint arXiv:2107.14795* (2021).

[14] Andrew Jaegle et al. "Perceiver: General perception with iterative attention". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 4651–4664.

[15] David R Kelley et al. "Sequential regulatory activity prediction across chromosomes with convolutional neural networks". In: *Genome research* 28.5 (2018), pp. 739–750.

[16] Alessio Mascolini et al. "A Novel Proof-of-concept Framework for the Exploitation of ConvNets on Whole Slide Images". In: *Progresses in Artificial Intelligence and Neural Systems*. Springer, 2021, pp. 125–136.

[17] Eric Phizicky et al. "Protein analysis on a proteomic scale". In: *Nature* 422.6928 (2003), pp. 208–215.

[18] Vittorio Pipoli et al. "Predicting gene expression levels from DNA sequences and post-transcriptional information with transformers". In: *Computer Methods and Programs in Biomedicine* (2022), p. 107035.

[19] Shankha Satpathy et al. “A proteogenomic portrait of lung squamous cell carcinoma”. In: *Cell* 184.16 (2021), pp. 4348–4371.

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

[21] Akiyoshi Wada and Haruki Nakamura. “Nature of the charge distribution in proteins”. In: *Nature* 293.5835 (1981), pp. 757–758.

[22] Liang-Bo Wang et al. “Proteogenomic and metabolomic characterization of human glioblastoma”. In: *Cancer Cell* 39.4 (2021), pp. 509–528.

[23] Yang You et al. “Large batch optimization for deep learning: Training bert in 76 minutes”. In: *arXiv preprint arXiv:1904.00962* (2019).

[24] Fangfei Zhang et al. “Data-independent acquisition mass spectrometry-based proteomics and software tools: a glimpse in 2020”. In: *Proteomics* 20.17-18 (2020), p. 1900276.

[25] Jingzhao Zhang et al. *Why ADAM Beats SGD for Attention Models*. 2019. arXiv: `1912.03194 [math.OC]`.

[26] Jian Zhou et al. “Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk”. In: *Nature genetics* 50.8 (2018), pp. 1171–1179.