Computer Methods and Programs in Biomedicine 234 (2023) 107504

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

Predicting gene and protein expression levels from DNA and protein sequences with Perceiver

Matteo Stefanini, Marta Lovino*, Rita Cucchiara, Elisa Ficarra

DIEF, University of Modena and Reggio Emilia, Via Vivarelli 10/1, Modena, 41125, Italy

ARTICLE INFO

Article history:
Received 25 October 2022
Revised 6 March 2023
Accepted 21 March 2023

Keywords:
Deep learning
DNA
mRNA expression
Perceiver
Protein expression
Sequence

ABSTRACT

Background and Objective: The functions of an organism and its biological processes result from the expression of genes and proteins. Therefore quantifying and predicting mRNA and protein levels is a crucial aspect of scientific research. Concerning the prediction of mRNA levels, the available approaches use the sequence upstream and downstream of the Transcription Start Site (TSS) as input to neural networks. The State-of-the-art models (e.g., Xpresso and Basenjii) predict mRNA levels exploiting Convolutional (CNN) or Long Short Term Memory (LSTM) Networks. However, CNN prediction depends on convolutional kernel size, and LSTM suffers from capturing long-range dependencies in the sequence. Concerning the prediction of protein levels, as far as we know, there is no model for predicting protein levels by exploiting the gene or protein sequences. *Methods:* Here, we exploit a new model type (called Perceiver) for mRNA and protein level prediction, exploiting a Transformer-based architecture with an attention module to attend to long-range interactions in the sequences. In addition, the Perceiver model overcomes the quadratic complexity of the standard Transformer architectures. This work's contributions are 1. DNAPerceiver model to predict mRNA levels from the sequence upstream and downstream of the TSS; 2. ProteinPerceiver model to predict protein levels from the protein sequence; 3. Protein&DNAPerceiver model to predict protein levels from TSS and protein sequences. *Results:* The models are evaluated on cell lines, mice, glioblastoma, and lung cancer tissues. The results show the effectiveness of the Perceiver-type models in predicting mRNA and protein levels. *Conclusions:* This paper presents a Perceiver architecture for mRNA and protein level prediction. In the future, inserting regulatory and epigenetic information into the model could improve mRNA and protein level predictions. The source code is freely available at https://github.com/MatteoStefanini/DNAPerceiver

1. Introduction

Most of the biological processes that regulate the functions of an organism are due to the activity of proteins [1–3]. In recent decades, the incredible development of sequencing techniques and proteomics quantifications have enabled a systematic analysis of the activity level of thousands of genes and proteins [4,5]. In addition, it is known that many regulatory and epigenetic processes regulate the expression of mRNAs and proteins [6–8], and the sequence upstream and downstream the transcription start site (TSS) has long been investigated to predict the mRNA levels in various tissues. In this paper, the periphrasis *TSS sequence* will be improperly used from now on to define the sequence upstream and downstream of the TSS to make the method clear.

Vice versa to the mRNA level prediction problem, the protein level prediction from sequences has yet to be addressed to the best of our knowledge. This work focuses on the necessary network developments to predict population mRNA and protein expression levels in specific species and tissues (e.g., mouse and human cell lines, human cancer tissues) as a prerequisite and baseline for future works on sample-specific mRNA and protein expression prediction.

In recent years, deep learning techniques spread in health applications [9–14] and previous works focused on mRNA level prediction from TSS sequences [15–18]. In particular, Convolutional Neural Networks have been adopted to deal with the sequential nature of the DNA [15–17,19].

Specifically, Basenjii [15] applies convolutional layers followed by dilated convolutions to share information across large distances in the gene sequences. Dilated convolutions have a wider filter created by inserting spaces in the filter elements. Those gaps expo-

* Corresponding author.
E-mail address: marta.lovino@unimore.it (M. Lovino).

https://doi.org/10.1016/j.cmpb.2023.107504

nentially increase the receptive field width, thus taking into account longer dependencies in the sequences.

Similarly, Expecto [16] applies convolutional layers to extract features from the 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 to reduce feature vector dimensionality to output mRNA levels.

On the same line, Xpresso [17] introduced a deep convolutional model composed of two sequential convolutional and max-pooling layers followed by two fully connected layers, demonstrating that a localized region around the transcription start site captures the most relevant information for mRNA level prediction.

Although convolutions represent an effective way to deal with gene 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 of long-range relationships and dependencies in sequence elements [20].

In 2017 the attention mechanism revolutionized sequence processing, achieving outstanding performance in capturing long-range dependencies due to each token's global interaction in the input sequence (so-called self-attention), extracting global information directly from the first layer [20]. However, the self-attention operator has a quadratic complexity $O(n^2)$, making the prediction unfeasible for long sequences.

The Enformer model [19] firstly applies self-attention to genomic data, capturing wide-ranging relationships and improving mRNA level prediction. However, to keep the computation feasible, the model is composed of a first convolutional step that extracts local features that are then applied to self-attention layers to capture long-range interactions.

Our method, instead, is based on the Perceiver architecture [21], 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 an initial convolutional step. The advantage of the Perceiver architecture is not limited to the computational aspects. The regulatory parts of a gene (e.g., enhancer and silencer) can be at a considerable distance from the gene region on which they act. Unlike CNN and LSTM, these long-range interactions are modeled in the Perceiver architecture, allowing a better mRNA level prediction.

In this work, we present three models, all based on the Perceiver architecture: DNAPerceiver, ProteinPerceiver, and DNA&ProteinPerceiver. DNAPerceiver predicts the mRNA and protein levels from the DNA sequence, and its performances are directly compared with competitor models on various datasets. ProteinPerceiver and DNA&ProteinPerceiver instead predict protein levels from the protein sequence and the combination of the DNA and protein sequences, respectively. 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.

2. Materials and methods

In order to predict mRNA and protein levels, human protein-coding genes were selected, and their TSS and protein sequences were obtained (see details in the Dataset section). Then, we developed three models that slightly differ in input and output configurations based on the prediction task they tackle, leaving the general structure very similar. Each model receives the TSS or the protein sequence as input and a number representing the samples' average amount of mRNA or protein levels is outputted for each sequence. The greater the number, the greater the amount of molecule (mRNA or protein) circulating. To summarize, the main differences between the three Perceiver architectures consist of the output desired and the input data: TSS sequences for DNAperceiver, protein sequences for ProteinPerceiver, and TSS and protein sequences for DNA&ProteinPerceiver.

2.1. Datasets

We evaluate our models adopting different settings based on the task. Overall, there are two input types: inputDNA and inputProt. InputDNA consists of the sequence of human protein-coding genes upstream and downstream of the transcription start site (TSS). The sequence upstream of the TSS contains the gene's promoter, while the sequence downstream of the TSS contains the exons and introns of the gene. InputDNA sequences are taken from the Xpresso publication [17] due to its particular data curation. Indeed, in this dataset, the TSS positions were accurately revised by Xpresso's authors exploiting Cap Analysis Gene Expression (CAGE) experiments, a method to measure the actual TSS location. Specifically, it comprises 18377 genes split into 16377 genes for training, 1000 for validation, and 1000 for the test. The maximum length of the TSS sequence of a gene is set to 20,000 base pairs. Xpresso DNA input also comes with half-life features, which contain general information about the gene (e.g., gene length and number of introns). Therefore, whenever we use InputDNA sequences, we also include half-life features as input to our models at different network points, as explained in the architecture section.

InputProt, on the other hand, consists of protein sequences. Therefore, the promoter region and all non-coding parts of a gene are not included in the inputProt sequence. All protein sequences were obtained from Uniprot database [22], processed with Biopython library [23], and intersected with Xpresso's list of protein-coding genes.

As for the labels, we used four typologies for predicting mRNA levels (labelGeneMouse, labelGeneHuman, labelGeneGlio, labelGeneLung) and two typologies for predicting protein levels (labelProtGlio and labelProtLung). labelGeneMouse and labelGeneHuman come from the Xpresso publication, containing the mean mRNA levels of 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 from clinical studies, we selected mRNA and protein levels on patients with glioblastoma [24] and lung cancer [25]. LabelGeneGlio and labelGeneLung contain the labels of the mediated mRNA values for glioblastoma and lung cancer tissues, respectively. The same procedure has been applied to obtain the mediated protein levels for the same patients, named labelProtGlio and labelProtLung [24,25].

Given the scarcity of data, 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.

2.2. Metric

To measure the effectiveness of our methods, we compute the variance explained r^2, also known as the coefficient of determination: $r^2 = 1 - \frac{SSR}{SST}$, where SSR stands for Sum Squared Regression (the sum of the residuals-actual values minus predicted value-squared) and SST for Total Sum of Squares (the sum of the distance the data is away from the mean all squared). This coefficient is the most widely adopted metric for mRNA level prediction, ranging from 0 to 1. When it is 0, the model makes a prediction no better than random, while when it is 1 the model perfectly predicts the actual labels.

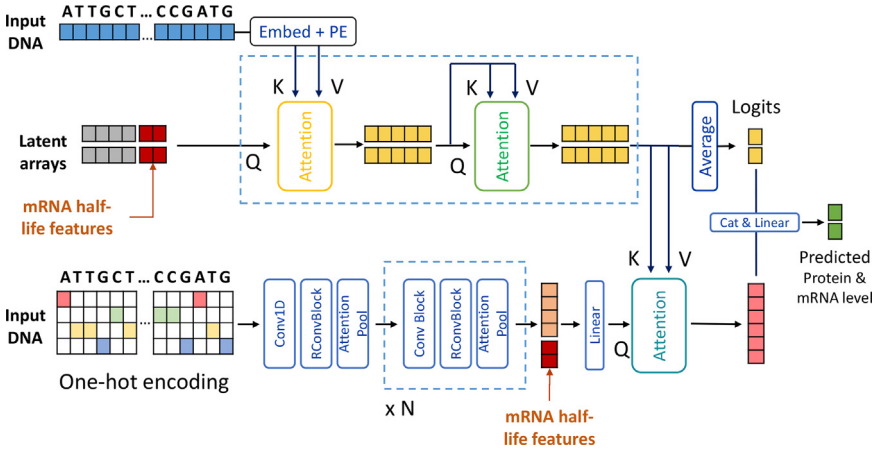

Fig. 1. DNAPerceiver architecture. It is based on the Perceiver IO model [26]. The upper flow represents the asymmetric attention that distills the sequence in a smaller latent space, where learnable arrays attend to all the input sequences and refine their representations with self-attention and feed-forward 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 sequence processed by a convolutional pipeline inspired by the Enformer model [19]. In this figure, Q,K,V stands for Query, Keys and Values as in typical Transformer architecture, PE is the Positional Encoding, Conv1D is a 1-dimensional Convolution and RConvBlock is a 1D Convolution with a residual connection. The first Convolutional layer is applied to the one-hot encoded version of the sequence, as all previous model of literature, while the upper part of the model embeds the one-hot vectors into learnable embedding vectors through linear projections, as typical Transformer architecture requires.

Table 1
Summary of the sources of data and labels used in our work.

Data	Sources	Data type	Length x Samples
InputDNA human	Xpresso [17]	Sequence	20,000 × 18,377
InputDNA mouse	Xpresso [17]	Sequence	20,000 × 21,856
Half-life feats human	Xpresso [17]	Sequence	8 × 18,377
Half-life feats mouse	Xpresso [17]	Sequence	8 × 21,856
InputProt	UniProt DB [22]	Sequence	6000 × N.A.
labelGeneHuman	Xpresso [17]	Value	1 × 18,377
labelGeneMouse	Xpresso [17]	Value	1 × 21,856
LabelGeneGlio	Glioblastoma DB [24]	Value	1 × 10,430
LabelProtGlio	Glioblastoma DB [24]	Value	1 × 10,430
LabelGeneLung	Lung cancer DB [25]	Value	1 × 10,699
LabelProtLung	Lung cancer DB [25]	Value	1 × 10,699

2.3. DNAPerceiver architecture

As stated above, various models in the literature focused on predicting mRNA levels from the TSS sequence. This work aims to reveal if mRNA levels can be explained by the TSS sequence alone. All predictive models do not use the whole gene sequence as input but only the TSS portion, which involves numerous regulatory and transcriptional 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 mRNA produced. The data used in this model configuration are inputDNA as input and labelGeneMouse, labelGeneHuman, 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 [21], and another with a convolutional step inspired by the Enformer model [19]. 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 step 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 a convolutional step, 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 sequence token. 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 [20].

These methods were recently applied to model mRNA sequences. However, given the quadratic complexity of attention $O(n^2)$, the length of the input can explode quadratically, rendering it infeasible to encode sequences of more than a few thousand letters. For this reason, our approach is based on the Perceiver [21], a model that builds upon Transformers but scales to hundreds of thousands of inputs, as it leverages an asymmetric attention mechanism to distill inputs into a tight latent bottleneck iteratively. Then, the 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 [26], which improved the decoder capabilities of the model by adding a final decoding stage. This stage 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 InputDNA sequence after being processed by a Convolutional step. This step consists of multiple Conv layers, Residual connections, and Attention Pooling layers inspired by the Enformer model [19]. 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 TSS or the protein sequence given in input is 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 step. After embedding the input letters, we also add a learnable Positional Encoding, initialized with a sinusoidal function as in the original Transformer model to deal with positions in the asymmetric attention.

Latent arrays are initialized with random numbers from a normal distribution with mean 0 and variance 1, while the inputDNA is represented with one-hot encoding vectors, applied to the Convolutional step, and linearly projected into embedding vectors for the attention path. Moreover, mRNA half-life features are injected in both flows: they are appended to the latent arrays and the convolutional step in the final feature representations.

In the DNAPerceiver configuration, we predict the mRNA 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 levels in human high throughput sequencing data. As discussed in the Dataset section, we take the protein labels from two real-human datasets, ending with 10,529 pairs of labels for Lung cancer (labelProtLung) and 10,280 pairs of labels for glioblastoma (labelProtGlio). In this configuration, we output two predictions for labelGene and labelProt, mRNA and protein levels, respectively, for both datasets.

To represent the A, T, C, and G letters, we use one-hot vectors, and for the perceiver flow, we linearly project them to learnable vectors of dimensionality 32. 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 [19] 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 InputDNA sequence is set to 10,500, taking the majority part from the promoter side and less from the actual gene, specifically considering 7000 base pairs before the TSS and 3500 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 [27], a batch size of 128, and we follow the learning rate scheduling strategy of [20] with a warmup equal to 8000 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 mRNA levels. In the end, we weighted the loss contribution using a weight of 10 for the mRNA.

2.4. ProteinPerceiver architecture

Supplementary Figure S1 shows the ProteinPerceiver model, which aims to measure how much protein levels depend on the protein sequence. The main differences with respect to the DNAPerceiver are the protein sequence as input (inputProtein) and the protein level as output (labelProtGlio and labelProtLung). Although the mRNA level prediction task is debated within the scientific community, to the best of our knowledge, there are no publicly available models for protein level prediction using protein sequences. In the last decade, the quantification of mRNA levels has been available in large quantities. Instead, extracting and quantifying proteins is more recent and less mature than mRNA extraction and quantification techniques. Protein quantification is what scientists are most interested in biologically. However, these techniques are currently more expensive, limiting data availability. Moreover, the mRNA level quantification can evaluate more than 20000 protein-coding genes versus approximately 2 to 8 thousand proteins for protein quantification. Unfortunately, given the experiments' novelty, no comparison model in the literature is available.

We match protein sequences and proteomics labels available, assembling a total of 10,430 protein sequences for Glioblastoma and 10,699 for Lung Carcinoma with corresponding proteomic labels.

The dropout keep probability is set 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 6000 and the final weight of the MSE loss to 100 for Lung data and 3000 for Glioblastoma data. We optimize the model using Lamb [28], a learning rate of 0.0005, and a Cosine Annealing schedule strategy with 8000 steps of warmup.

2.5. DNA&ProteinPerceiver architecture

The previous ProteinPerceiver model receives the protein sequence as input to predict protein levels. However, the protein level is determined by the protein sequence and by regulatory, transcriptional, and epigenetic factors. Although considering all regulatory processes is not straightforward, in this configuration, called DNA&ProteinPerceiver, we have evaluated the combined effect of the protein and the TSS sequence to predict the protein levels. This configuration simultaneously uses inputDNA and inputProtein and outputs labelProtGlio and labelProtLung. TSS and protein sequences are matched when both are available from the Xpresso dataset [17] and the protein sequence dataset, ending up with a total of 9815 triplets gene-protein-labels for Lung cancer and 9534 triplets for Glioblastoma.

Since our model deals with two different input sequences, we investigated the use of the inputs in an alternate manner: when the protein sequence is given to the Perceiver, we use the DNA TSS sequence as a query in the convolutional pipeline, as shown in Supplementary Figure S2, and vice versa, with DNA TSS sequence as the perceiver input, we use the protein for the query computation, shown in Supplementary Figure S3. The Results section shows that the best version differs depending on the data and the prediction. The maximum length of the protein sequence is set to 6000, while the DNA sequence length is set to 8000. If not specified, we kept the same hyperparameters of the DNAPerceiver configuration.

Table 2
Summary of the different configurations of our model depending on the prediction task and the input-output setting.

Model Configuration	Tasks	Input	Output
DNAPerceiver	mRNA levels	InputDNA	labelGene
DNAPerceiver	mRNA&protein levels	InputDNA	labelGene&labelProt
ProteinPerceiver	protein levels	InputProt	labelProt
DNA&ProteinPerceiver	protein levels	InputDNA&InputProt	labelProt

A summary of the architecture names, prediction tasks, input, and outputs is reported in Table 2.

3. Results

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

3.1. Results on mRNA level prediction using Xpresso's labels

In this setting, DNAPerceiver was trained on the Xpresso sequences and their labels, aiming to predict the mRNA level, both from mouse and human organisms (labelGeneMouse and labelGeneHuman). We follow the split of the original dataset [17], thus obtaining 16,377 genes for training and 1000 genes for both validation and test set. As shown in Table 3, DNAPerceiver performs better than the 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, gains 0.03 points of r^2.

The basenji method has a similar mRNA level 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 (Xpresso's dataset released the sequences but not the CAGE information). 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 gets consistent results.

3.2. Results on mRNA and protein levels

Table 4 reports the results obtained with the DNAPerceiver architecture. High-throughput sequencing data from human 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 tissues (tumor tissues, 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, its mRNA level 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 level using only the InputDNA sequence. As a result, our model can predict the protein levels achieving 0.161 of r^2, demonstrating its capability to perceive the direct connection between the InputDNA sequence and its corresponding protein level.

Table 3
Results on the test set of Xpresso dataset in predicting mRNA levels of cell-line data. The input is the InputDNA sequence, and the output is the mRNA level, expressed with the coefficient of determination r^2.

Model	mRNA r^2
Xpresso [17] human data	0.59
Xpresso [17] mouse data	0.71
DNAPerceiver human data	0.62
DNAPerceiver mouse data	0.72

Table 4
Results on Lung and Glioblastoma data in predicting mRNA and protein level. The input is inputDNA, taken from Xpresso [17] publication, while predicted labels for mRNA and protein levels are labelGeneLung, labelGeneGlio, labelProtLung, and labelProtGlio. Results are the average of the k-fold validation method with k equal to 10.

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

Table 5
Results in predicting protein levels from the protein sequence. The input is InputProt, while predicted labels for protein levels are labelProtLung and labelProtGlio. Results are the average of the k-fold validation method with k equal to 10.

Model	proteomics r^2
ProteinPerceiver Lung	0.085
ProteinPerceiver Glioblastoma	0.028

Table 6
Results in predicting protein levels from both the DNA sequence and the protein sequence used as inputs. The input is inputProt and 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.

Model	proteomics r^2
DNA&ProteinPerceiver Lung	0.141
DNA&ProteinPerceiver Glioblastoma	0.031

3.3. Results on protein level using protein sequence as input

Table 5 reports the result of our ProteinPerceiver model. The obtained outcome varies depending on the data: for Lung data, we found that predicting protein levels from the protein sequence is more complex, achieving a r^2 of 0.085, comparing the 0.161 obtained from the InputDNA. Nonetheless, for Glioblastoma data, our ProteinPerceiver can score a r^2 of 0.028 for protein levels, 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 protein level.

3.4. Results on protein levels using TSS and protein sequences as input

We wanted to investigate further the model's capabilities with a peculiar configuration, in which we give as input both the TSS (InputDNA) and the protein sequence. Therefore, the protein sequence was input to the perceiver and the InputDNA to the convolutional query and vice-versa. We report the results in Table 6. 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 InputDNA sequence. On the contrary, using both inputs slightly improves the results on Glioblastoma data, achieving 0.031 of r^2. Finally, we computed the r^2 values obtained from a random model for each of the three inputs (DNAPerceiver, ProteinPerceiver, and DNA&Protein Perceiver) versus the glioblastoma dataset (where proteomic results are limited). The random models' 10-fold cross-validation means are 0.00253 (max 0.00813, min 2.25e-05), 0.000917 (max 0.00391, min 8.98e-08), and 0.00145 (max 0.00657, min 2.17e-06). Glioblastoma 10-fold cross-validation means in predicting proteomic values for the three models are 0.026 (max 0.034, min 0.017), 0.028 (max 0.035, min 0.014), and 0.031 (max 0.037, min 0.021). The relative protein expression value (how much a protein is expressed compared to the others) is crucial too. Thus, although the overall effectiveness of the protein model is limited, its predictive power can be of interest to scientists. Indeed, 60% and 68% (globally and in medulloblastoma, respectively) of the most highly expressed proteins are predicted as expressed. The main reason for the noisy result could be attributed to post-transcriptional regulatory processes which are widely known as crucial players in protein expression.

4. Discussion

Regarding predicting mRNA levels from the sequence upstream and downstream of 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. Unlike the Xpresso model, the DNAPerceiver model exploits the self-attention mechanism to predict the mRNA levels. Having the same input sequence size and output levels as Xpresso, the DNAPerceiver model achieves superior results since long-range interactions between the most distant regions of the promoter and the gene sequence are fully exploited in the model and not limited by the size of the convolutional kernel. Moreover, as can be expected, the prediction of mRNA levels in cell lines achieves better results than mRNA level prediction in tumor samples. This aspect could be explained by the different boundary conditions of the two situations. In the first case, the mRNA expression is controlled to ensure the reproducibility and stability of the cell lines. In the second case, the intrinsic samples' variability cannot be limited and pathological conditions profoundly alter the biological context. Since no comparable studies in predicting protein expression levels are available, more distant works that predict protein expression are described. In particular, Barzine et al. [29] purpose is the imputation of unquantified proteins exploiting mRNA expression data. Indeed, it does not answer whether it is possible to predict protein expression starting from the gene sequence. In detail, mRNA expression values are known in the literature to predict protein expression values, as there is often a positive correlation between the mRNA and its protein expression. Barzine et al. also consider the variability of the same protein in different samples (e.g., people) based on mRNA expression variability. As innovative as it is, Barzine et al. answer a very different question, namely quantifying the expression values of those proteins whose mRNA value is known. Fernandes and Vinga [30] aim to predict the expression of proteins by encoding their codons. However, the prediction was made for Escherichia coli from two datasets with a limited number of proteins. The first one contains the expression levels of two proteins, a DNA polymerase, and a single-chain antibody, for 55 codon encodings. The second one contains the level of a green fluorescent protein produced with 154 different codon encodings. The main limitation of Fernandes et al.s work is the number of proteins quantified based on the specific sequence detected in the sample. Moreover, it is based on the correlation between the levels of the green fluorescent protein and the free energy of the protein itself. Although the purpose is similar to the Perceiver, there is no way to make a direct comparison with our work.

Besides the improvement in mRNA level prediction, the main novelty of this work is the first adoption of the Perceiver architecture for gene expression, and the prediction of protein levels from the TSS and protein sequences. This aspect is doubly challenging: 1. Protein extraction and quantification techniques have emerged recently, so data availability still needs to be improved compared to mRNA datasets; 2. The protein sequence has target regions for post-translation regulators; however, the promoter region is not used as input in the ProteinPerceiver model. It is noted that the prediction of protein levels is considerably lower than mRNA ones, whether the prediction exploits the TSS or the protein sequence. Although the results are limited, when compared to random models, protein level prediction is statistically relevant. The complexity of the problem can explain this phenomenon. The protein level is influenced by notable post-transcriptional and post-translational regulatory phenomena (e.g., ubiquitination), which are not fed to the models. Moreover, the TSS sequences (composed of the promoter and a part of the gene) have a greater predictive power of the protein level than the protein sequences. This behavior could depend on the presence of the promoter. Indeed, the promoter is the region that favors the expression regulation (both of genes and, therefore, of proteins), and it is responsible for interacting with transcription factors. When the model is trained simultaneously with the TSS and the protein sequence, the predictive power of protein level increases; however, it remains lower than the prediction of protein levels using only the TSS sequence. In this sense, the TSS sequence seems more informative than the protein one. Although the protein expression level prediction is critical, the expression value of a protein compared to the others is crucial too. In this sense, the predictive power of the proposed model can be of interest to scientists. Indeed, 60% and 68% (globally and in medulloblastoma, respectively) of the most highly expressed proteins are predicted as expressed. In the end, the main reason for the noisy result could be attributed to post-transcriptional regulatory processes which are widely known as crucial players in protein expression.

5. Conclusions

Various papers have addressed mRNA level prediction in the literature, mainly convolutional or long short-term memory networks. This work presents three Perceiver-type architectures for predicting mRNA levels on cell lines and high-throughput human samples. Furthermore, a novel task is introduced, presenting the prediction of protein levels from the TSS and protein sequences. The results show the advantages of the Perceiver architecture in predicting mRNA levels compared to competitors. On the other hand, protein level prediction benefits more from the TSS sequence than the protein one. This aspect could be explained by the presence of the promoter region in the TSS sequence.

The Perceiver architecture benefits are not limited to the computational aspects. Since the regulatory parts of a gene (e.g., enhancer and silencer) can be at a considerable distance from the TSS, these regions can be directly attended by Perceiver models. Furthermore, unlike CNN and LSTM, long-range interactions can be exploited in the Perceiver architecture, allowing a better prediction.

Although various experimental conditions have been considered, in the future, other biological post-transcriptional and post-translation regulations can be included in the models to improve prediction. Moreover, it could be possible to predict the protein expression levels in sample-specific mode (the protein expression level in a sample based on the expression levels in similar sam-

ples). This step is essential when it is not possible to quantify the protein expression from the detected peptides directly.

Funding

This study was funded by the European Unions Horizon 2020 research and innovation programme DECIDER under Grant Agreement 965193, and partially supported by the Artificial Intelligence for Cultural Heritage (AI4CH) project, co-funded by the Italian Ministry of Foreign Affairs and International Cooperation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cmpb.2023.107504.

References

[1] F. Crick, L. Barnett, S. Brenner, R.J. Watts-Tobin, et al., General nature of the genetic code for proteins, Nature (1961).
[2] F. Crick, Central dogma of molecular biology, Nature 227 (5258) (1970) 561–563.
[3] A. Wada, H. Nakamura, Nature of the charge distribution in proteins, Nature 293 (5835) (1981) 757–758.
[4] F. Zhang, W. Ge, G. Ruan, X. Cai, T. Guo, Data-independent acquisition mass spectrometry-based proteomics and software tools: a glimpse in 2020, Proteomics 20 (17-18) (2020) 1900276.
[5] P.H. Reyes-Herrera, E. Ficarra, Computational methods for clip-seq data processing, Bioinf. Biol. Insights 8 (2014) BBI–S16803.
[6] E. Jablonka, M.J. Lamb, The changing concept of epigenetics, Ann. New York Acad. Sci. 981 (1) (2002) 82–96.
[7] A. Bird, Perceptions of epigenetics, Nature 447 (7143) (2007) 396.
[8] M. Esteller, Epigenetics in cancer, New Engl. J. Med. 358 (11) (2008) 1148–1159.
[9] M. Lovino, M. Montemurro, V.S. Barrese, E. Ficarra, Identifying the oncogenic potential of gene fusions exploiting miRNAs, J. Biomed. Inf. 129 (2022) 104057.
[10] A. Mascolini, S. Puzzo, G. Incatasciato, F. Ponzio, E. Ficarra, S. Di Cataldo, 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.
[11] S. Allegretti, F. Bolelli, F. Pollastri, S. Longhitano, G. Pellacani, C. Grana, Supporting skin lesion diagnosis with content-based image retrieval, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 8053–8060, doi:10.1109/ICPR48806.2021.9412419.
[12] A. Khan, Z. Rehman, H.F. Hashmi, A.A. Khan, M. Junaid, A.M. Sayaf, S.S. Ali, F.U. Hassan, W. Heng, D.-Q. Wei, An integrated systems biology and network-based approaches to identify novel biomarkers in breast cancer cell lines using gene expression data, Interdiscip. Sci. Comput. Life Sci. 12 (2) (2020) 155–168.
[13] W. Zhang, X. Xue, C. Xie, Y. Li, J. Liu, H. Chen, G. Li, CEGSO: boosting essential proteins prediction by integrating protein complex, gene expression, gene ontology, subcellular localization and orthology information, Interdiscip. Sci. Comput. Life Sci. 13 (3) (2021) 349–361.
[14] A. Mascolini, D. Cardamone, F. Ponzio, S. Di Cataldo, E. Ficarra, Exploiting generative self-supervised learning for the assessment of biological images with lack of annotations, BMC Bioinf. 23 (1) (2022) 1–17.
[15] D.R. Kelley, Y.A. Reshef, M. Bileschi, D. Belanger, C.Y. McLean, J. Snoek, Sequential regulatory activity prediction across chromosomes with convolutional neural networks, Genome Res. 28 (5) (2018) 739–750.
[16] J. Zhou, C.L. Theesfeld, K. Yao, K.M. Chen, A.K. Wong, O.G. Troyanskaya, Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk, Nat. Genet. 50 (8) (2018) 1171–1179.
[17] V. Agarwal, J. Shendure, Predicting mRNA abundance directly from genomic sequence using deep convolutional neural networks, Cell Rep. 31 (7) (2020) 107663.
[18] V. Pipoli, M. Cappelli, A. Palladini, C. Peluso, M. Lovino, E. Ficarra, Predicting gene expression levels from dna sequences and post-transcriptional information with transformers, Comput. Methods Programs Biomed. (2022) 107035–107045.
[19] Ž. Avsec, V. Agarwal, D. Visentin, J.R. Ledsam, A. Grabska-Barwinska, K.R. Taylor, Y. Assael, J. Jumper, P. Kohli, D.R. Kelley, Effective gene expression prediction from sequence by integrating long-range interactions, Nat. Methods 18 (10) (2021) 1196–1203.
[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is All you Need, in: I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, 30, Curran Associates, Inc., 2017. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
[21] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, J. Carreira, Perceiver: general perception with iterative attention, in: International Conference on Machine Learning, PMLR, 2021, pp. 4651–4664.
[22] U. Consortium, UniProt: a worldwide hub of protein knowledge, Nucl. Acids Res. 47 (D1) (2019) D506–D515.
[23] P.J. Cock, T. Antao, J.T. Chang, B.A. Chapman, C.J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, et al., Biopython: freely available Python tools for computational molecular biology and bioinformatics, Bioinformatics 25 (11) (2009) 1422–1423.
[24] L.-B. Wang, A. Karpova, M.A. Gritsenko, J.E. Kyle, S. Cao, Y. Li, D. Rykunov, A. Colaprico, J.H. Rothstein, R. Hong, et al., Proteogenomic and metabolomic characterization of human glioblastoma, Cancer Cell 39 (4) (2021) 509–528.
[25] S. Satpathy, K. Krug, P.M.J. Beltran, S.R. Savage, F. Petralia, C. Kumar-Sinha, Y. Dou, B. Reva, M.H. Kane, S.C. Avanessian, et al., A proteogenomic portrait of lung squamous cell carcinoma, Cell 184 (16) (2021) 4348–4371.
[26] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, et al., Perceiver IO: a general architecture for structured inputs & outputs, arXiv preprint arXiv:2107.14795(2021).
[27] J. Zhang, S.P. Karimireddy, A. Veit, S. Kim, S.J. Reddi, S. Kumar, S. Sra, Why adam beats SGD for attention models, 2019, 1912.03194
[28] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, C.-J. Hsieh, Large batch optimization for deep learning: training bert in 76 minutes, arXiv preprint arXiv:1904.00962(2019).
[29] M.P. Barzine, K. Freivalds, J.C. Wright, M. Opmanis, D. Rituma, F.Z. Ghavidel, A.F. Jarnuczak, E. Celms, K. Čerāns, I. Jonassen, et al., Using deep learning to extrapolate protein expression measurements, Proteomics 20 (21-22) (2020) 2000009.
[30] A. Fernandes, S. Vinga, Improving protein expression prediction using extra features and ensemble averaging, PloS One 11 (3) (2016) e0150369.