Bioinformatics, 38(3), 2022, 597–603
https://doi.org/10.1093/bioinformatics/btab746
Advance Access Publication Date: 28 October 2021
Original Paper

Genome analysis

CpG Transformer for imputation of single-cell methylomes

Gaetan De Waele , Jim Clauwaert , Gerben Menschaert and Willem Waegeman *

Department of Data Analysis and Mathematical Modelling, Ghent University, Ghent 9000, Belgium

*To whom correspondence should be addressed.

Associate Editor: Peter Robinson

Received on June 8, 2021; revised on October 19, 2021; editorial decision on October 24, 2021; accepted on October 25, 2021

Abstract

Motivation: The adoption of current single-cell DNA methylation sequencing protocols is hindered by incomplete coverage, outlining the need for effective imputation techniques. The task of imputing single-cell (methylation) data requires models to build an understanding of underlying biological processes.

Results: We adapt the transformer neural network architecture to operate on methylation matrices through combining axial attention with sliding window self-attention. The obtained CpG Transformer displays state-of-the-art performances on a wide range of scBS-seq and scRRBS-seq datasets. Furthermore, we demonstrate the interpretability of CpG Transformer and illustrate its rapid transfer learning properties, allowing practitioners to train models on new datasets with a limited computational and time budget.

Availability and implementation: CpG Transformer is freely available at https://github.com/gdewael/cpg-transformer.

Contact: willem.waegeman@ugent.be

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

DNA methylation is the addition of a methyl group to the DNA. The best-known type is CpG methylation, where the methyl group is added to the C-5 position of CG dinucleotides. Its association with a broad range of biological processes, such as gene expression regulation, is well-established (Cedar, 1988). CpG methylation is also known as a driving factor in developmental biology and carcinogenesis, motivating the need to study this phenomenon on the cellular level (Bird, 2002).

The last decade, several protocols that measure DNA methylation at single-cell resolution have been developed. These methods make use of bisulfite conversion of DNA followed by sequencing (Krueger *et al.*, 2012), both on genome-wide scale (scBS-seq) (Smallwood *et al.*, 2014) and using reduced-representation protocols (scRRBS-seq) (Guo *et al.*, 2013). These methods have uncovered the heterogeneity and dynamics of epigenetic patterns between cells and have made it possible to describe epigenomic networks on an unprecedented scale and resolution (Angermueller *et al.*, 2016).

Due to the smaller amount of genetic material available per cell, profiling single cells comes with certain challenges not encountered in bulk sequencing experiments. In practice, the genome-wide coverage of CpG sites per cell is low, ranging from 1% for high-throughput studies (Farlik *et al.*, 2016) to 20% for low-throughput ones (Smallwood *et al.*, 2014). Furthermore, profiled sites are covered by a smaller number of reads, resulting in noisy measurements of DNA methylation. Effective imputation and denoising techniques are therefore crucial in unlocking the full potential of single-cell methylome analyses.

Prediction of methylation states in tissue samples is a well-established problem in bioinformatics, often tackled by leveraging dependencies between CpG sites. For example, variational autoencoders have been successfully applied for dimensionality reduction of methylation data (Levy *et al.*, 2020). Other methods focus on imputation of single CpG sites in tissue samples using, among others, linear regression (Di Lena *et al.*, 2019), random forests (Zhang *et al.*, 2015), autoencoders (Qiu *et al.*, 2018), gradient boosting (Zou *et al.*, 2018) or mixture models (Yu *et al.*, 2020). In addition to using intrasample dependencies between neighboring CpG sites, some of these methods adopt the idea of leveraging information from multiple (tissue) samples for prediction (Yu *et al.*, 2020; Zou *et al.*, 2018).

Most recent work on single-cell DNA methylation imputation has built upon this idea of leveraging both intra- and intercellular correlations between methylation states. Melissa (Kapourani and Sanguinetti, 2019) first defines specific regions of interest in the genome (such as a specific promoter region), then performs generalized linear model regression on CpG sites in that region. The model leverages information from other cells through a shared prior distribution determined by a Bayesian mixture model, effectively clustering cells. DeepCpG (Angermueller *et al.*, 2017) proposes a recurrent neural network (RNN) to process differences in local CpG profiles

across cells. For every cell, the local CpG profile consists of a vector containing the methylation states and distances of the 25 nearest observed CpG sites up- and downstream from the target site. Along with this RNN, a convolutional neural network (CNN) processes relevant information in the DNA sequence surrounding the target site. The two streams of information are combined near the end of the network. Finally, the output head returns predictions for every cell at a single CpG site. Using similar design principles, LightCpG uses gradient boosting to obtain faster training times at the cost of a lower performance (Jiang *et al.*, 2019). CaMelia (Tang *et al.*, 2021), also relying on gradient boosting models, restricts its imputation to CpG sites that are also recorded in at least one other cell. It additionally discards CpG sites whose local methylation profiles are too dissimilar of the profiles in all other cells. Uniquely, CaMelia introduces the notion of using bulk tissue samples to improve performance compared to DeepCpG and trains a separate model for every cell. It remains unclear, however, whether these performance gains can be attributed to the employed methods or to the aforementioned sample selection.

In this work, inspiration is drawn from recent developments in self-supervised learning of natural language. In particular, the language model BERT is trained by randomly replacing words in a sentence by a unique `[MASK]` token and attempting to predict the masked word given the newly formed sentence [called masked language modeling (MLM)] (Devlin *et al.*, 2018). In essence, this objective trains a model to fill in the gaps in a sentence. The similarity with imputation, where gaps in a matrix need to be filled in, is compelling but unexplored. In language modeling, transformer neural networks are used because of their capability of learning interactions between all input words, akin to the flow of information in a complete digraph (Radford *et al.*, 2018).

Biological systems can be elegantly represented by graphs (Barabasi and Oltvai, 2004). For example, the interactions of genes form distinct pathways in a regulatory network. Consequently, models should ideally reason over graphs or mimic graph structure. Most of the current deep learning practices in bioinformatics do not reflect this reality. For example, fully connected layers learn a set of fixed weights for all inputs and are hence unable to reason over how correlations between inputs differ when their contents change. Transformers mimic graph structure using a self-attention mechanism to explicitly reason over how every input is influenced by the others (Vaswani *et al.*, 2017). Because of this, transformers scale quadratically in computational- and memory cost with the number of inputs. They have previously been shown to outperform other neural architectures in DNA sequence annotation tasks (Clauwaert and Waegeman, 2020) and protein representation learning (Elnaggar *et al.*, 2020; Rives *et al.*, 2021). Recently, the use of transformers in biology has gone beyond 1D sequences. For example, MSA Transformer (Rao *et al.*, 2021) adapts axial attention (Ho *et al.*, 2019) to MSAs for unsupervised protein structure learning. AlphaFold2 (Jumper *et al.*, 2021) also adapts axial attention to process both MSAs and residue pair matrices. By processing 2D inputs, full self-attention learns $\mathcal{O}(n^2m^2)$ pairwise interactions for a $\mathbb{R}^{n\times m}$ matrix, making vanilla transformers impossible to apply on high-dimensional methylation data.

We introduce CpG Transformer, an adaptation of the transformer neural network architecture to operate on partially observed methylation matrices by combining axial attention (Ho *et al.*, 2019) with sliding window self-attention (Beltagy *et al.*, 2020), thereby obtaining state-of-the-art imputation performances on a wide range of datasets. The inputs to CpG Transformer consist of the CpG matrix along with their respective positions on the genome and the DNA sequences surrounding them. Cell identity is communicated to the model through learned cell embeddings. The model learns a representation for every CpG site and recombines their information in a graph-like manner. Because of this, the architecture captures general-purpose representations, allowing for quick transfer learning of imputation models on new datasets, a prospect of great interest to practitioners with limited computational resources. In addition, ablation studies and model interpretation demonstrate the contributing factors to single-cell DNA methylation.

2 Materials and methods

Here, CpG Transformer is described for the imputation of DNA methylation data. Our architectural contributions are twofold. First, CpG Transformer draws inspiration from collaborative filtering approaches to formulate its inputs to the transformer layers (He *et al.*, 2017). The transformer layers model the interactions between matrix entries. In this sense, CpG Transformer can be regarded as *contextualized collaborative filtering*. Second, we extend axial attention (Ho *et al.*, 2019) to incorporate sliding window self-attention (Beltagy *et al.*, 2020), where full self-attention is applied per individual column and sliding window self-attention is applied over all rows separately.

如果两个cpg位点相距较近，他们的dna上下文窗口是会存在重叠 这种重叠有助于模型更好地捕捉基因组上相邻位点之间地关系

2.1 Model inputs

The input to CpG Transformer is a three-dimensional tensor $\mathbf{H} \in \mathbb{R}^{n\times m\times d_{\text{model}}}$, where $\mathbf{H}_{i,j} \in \mathbb{R}^{d_{\text{model}}}$ represents the input representation at cell i (rows) and methylation site j (column) of the methylation matrix. Every representation $\mathbf{H}_{i,j}$ is the result of linear combination of a concatenation of three embeddings: $\mathbf{H}_{i,j} = \boldsymbol{W}\cdot[\boldsymbol{b}_{i,j}^{\text{CpG}}, \boldsymbol{b}_{i}^{\text{cell}}, \boldsymbol{b}_{j}^{\text{DNA}}]$ (Fig. 1). All three embeddings consist of 32 hidden dimensions, and are combined to $d_{\text{model}} = 64$ dimensions by $\boldsymbol{W}$. The CpG embedding $\boldsymbol{b}_{i,j}^{\text{CpG}}$ is obtained by embedding the methylation state (unknown `?`, unmethylated `0` or methylated `1`) of CpG site j in cell i. Similarly, row-wise cell embeddings $\boldsymbol{b}_{i}^{cell}$ encode a hidden representation for cell indices. Finally, DNA sequence information is included in the model by taking 1001 nucleotide windows centered around the methylation sites and processing them with a CNN to obtain column-wise DNA embeddings $\boldsymbol{b}_{j}^{\text{DNA}}$. In all experiments, the CNN architecture is adapted from DeepCpG, consisting of two convolutional layers, each followed by a max-pooling layer (Angermueller *et al.*, 2017). The exact parameters of the CNN backbone are elaborated in Supplementary Section S1.

这里关于CpG transformer中提到的query和key 他们是自注意力机制中的关键概念

2.2 CpG Transformer

Transformer layers employ self-attention to explicitly reason over how every input is influenced by the others (Vaswani *et al.*, 2017). All n entries of an input $\boldsymbol{X}$ are once encoded as a query and once as a key via learned linear layers. For this model setup, the input to the transformer layers is $\mathbf{H}$. Taking the inner product of the queries $\boldsymbol{Q}$ with the keys $\boldsymbol{K}$ results in an $n\times n$ matrix, whose values can be loosely interpreted as the importance of input j for input i, at row i and column j. These values are normalized and multiplied by a value matrix $\boldsymbol{V}$ (obtained via linear combination of the input $\boldsymbol{X}$ with learned weights) to produce outputs for every input entry in a matrix $\boldsymbol{Z}$. This process can be performed multiple times in parallel using separate weight matrices, constituting different attention heads. Corresponding outputs $\boldsymbol{Z}$ for every head can then be concatenated and linearly combined to an appropriate hidden dimension size. The scaled dot-product self-attention mechanism first described by Vaswani *et al.* (2017) is given by the following equations, where d_k denotes the hidden dimensionality of the queries and keys

QKt score matrix represents the relationships or similarities between the inputs

$$\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_q^\top, \boldsymbol{X}\boldsymbol{W}_k^\top, \boldsymbol{X}\boldsymbol{W}_v^\top$$
$$\boldsymbol{Z} = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}\right)\boldsymbol{V} \quad (1)$$

- These weight matrices are learned during the training process and are essential for capturing the relationships between different elements of the input data.

Intuitively, this mechanism simply learns how inputs should be recombined in order to propagate to an output at every position. As such, no structure in the input is assumed and fixed-length inputs are not required, as identical model weights are used for every position. For an input methylation matrix with n cells (rows) and m methylation sites (columns), an $n\cdot m\times n\cdot m$ attention matrix would be obtained. Because m can easily exceed millions, it is impossible to apply vanilla transformers to methylation data. To reduce the computational complexity of this operation, axial attention (Ho *et al.*, 2019) can be employed. In axial attention, dependencies between elements of the same row and elements of the same column are modeled separately by two distinct self-attention operations in every layer (Fig. 1). In doing so, the memory and computational complexity is reduced from $\mathcal{O}(n^2m^2)$ to $\mathcal{O}(mn(n+m))$. Further, known

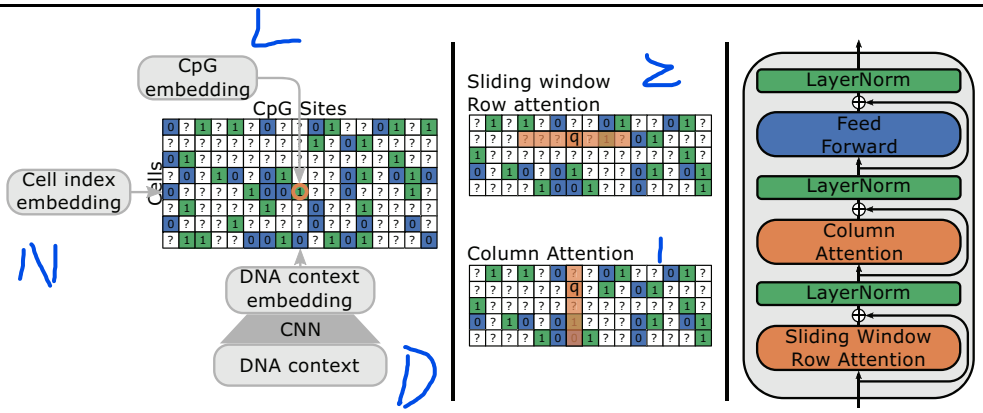

Fig. 1. (Left) Inputs to CpG Transformer. Cell, DNA and CpG embeddings are applied row-, column- and element-wise, respectively. (Middle) Illustration of sliding window row attention and (full) column attention. For sliding window row attention, every query attends to keys in the same row within a fixed window. For column attention, every query attends to the keys from all elements in the same column. (Right) A single CpG Transformer layer

autocorrelation between neighboring methylation sites can be leveraged. It is known that CpG sites in close proximity of each other on the genome are often correlated (Cokus *et al.*, 2008). Hence, we can limit row-wise attention to interactions between neighboring CpG sites in a sliding window attention mechanism (Beltagy *et al.*, 2020; Zaheer *et al.*, 2020), further reducing the complexity from $\mathcal{O}(mn(n+m))$ to $\mathcal{O}(mn(n+w))$, with a window size w (analogous to kernel size in convolutions). In order to communicate relative distances of CpG sites to the model, row-wise sliding window self-attention operation is supplied with relative sinusoidal positional encodings (Dai *et al.*, 2019). Pseudocode describing both row- and column-wise self-attention operations in more detail can be found in Supplementary Section S2.

CpG Transformer employs a stack of four identical layers (Fig. 1). The layer structure is similar to the one defined by Vaswani *et al.* (2017) and Rao *et al.* (2021). Each layer has three sublayers. The first and second sublayers consist of the previously described column-wise self-attention and row-wise sliding window self-attention with 8 heads of 8 hidden dimensions each. A window size of $w = 41$ is used for the sliding window self-attention (analogous to a convolutional kernel size of 41). The window size is selected considering a trade-off between computational complexity and inclusion of biological information. A larger window size means that CpG Transformer recombines information from more neighboring sites at the cost of computational- and memory complexity. The input and output dimensionality of the attention layer is $d_{\text{model}} = 64$. The last sublayer employs a position-wise fully connected feed-forward network consisting of two linear combinations with a ReLU activation in between: $\max(0, \boldsymbol{X}\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2$. The dimensionality of input and output is $d_{\text{model}} = 64$ and the inner-layer has 256 hidden dimensions. A residual connection (He *et al.*, 2016) followed by layer normalization (Ba *et al.*, 2016) is employed around all sublayers. The outputs of the last transformer layer are reduced to one hidden dimension by an output head and subjected to a sigmoid operation to obtain final predictions $\hat{\boldsymbol{Y}} \in \mathbb{R}^{n \times m}$ for all inputs.

2.3 Training objective

We adapt the MLM objective for DNA methylation imputation (Devlin *et al.*, 2018). MLM is a type of denoising autoencoding in which the loss function acts only on the subset of inputs that are perturbed. For CpG Transformer, the inputs are corrupted by randomly masking observed sites to the `?` token. In addition, 20% of the tokens that would be masked are instead randomized to a random state (`0` or `1`), sampled proportionally to the distribution of methylation states in the input. In doing so, CpG Transformer learns not only to impute but also to denoise. Finally, the cross-entropy loss optimizes the model to return the original methylation states given the corrupted input. Devlin *et al.* (2018) additionally proposes to leave a percentage of the masked tokens to be unchanged instead. Considering our limited vocabulary size (unknown `?`, unmethylated `0` or methylated `1`), a sufficient percentage of randomized tokens is actually unchanged, eliminating the need for this operation. An overview of the training procedure is given in Figure 2.

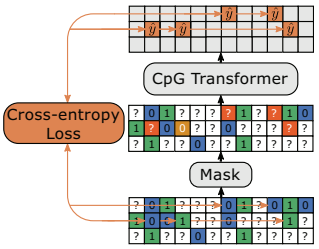

Fig. 2. Masked language modeling. Positive and negative sites are indicated in green and blue, respectively. Sites to train on (orange) are either masked (80%) or randomized (20%). The model is optimized to infer the original methylation state given the corrupted input using the cross-entropy loss

2.4 Datasets

Five publicly available datasets originating from both scBS-seq (Smallwood *et al.*, 2014) and scRRBS-seq (Guo *et al.*, 2013) experiments are obtained from the Gene Expression Omnibus.

The first dataset (GSE56879) consists of 20 mouse embryonic stem cells cultured in Serum. The second dataset is obtained from the same study and is made up of 12 cells of the same type cultured in 2i medium (Smallwood *et al.*, 2014). Both datasets were profiled using scBS-seq. A third dataset (GSE65364) comprises 25 human hepatocellular carcinoma cells profiled using scRRBS-seq (Hou *et al.*, 2016). scRRBS-seq profiles of 30 human monoclonal B-cell lymphocytes form a fourth dataset (GSE125499; sc05) (Kretzmer *et al.*, 2021). The final dataset (GSE87197) consists of 122 hematopoietic stem cells and progenitor cells profiled using scBS-seq (Farlik *et al.*, 2016). This dataset includes 18 hematopoietic stem cells, 18 multipotent progenitors, 19 common myeloid progenitors, 24 multilymphoid progenitors, 22 granulocyte macrophage progenitors and 21 common lymphoid progenitors. In the remainder of this paper, these datasets are referred to as Ser, 2i, HCC, MBL and Hemato, respectively. Corresponding reference genomes are as follows: Ser and 2i use genome build NCBIM37. GRCh38 is used by Hemato, and GRCh37 serves as reference genome for HCC and MBL. A brief summary of dataset sizes is available in Supplementary Table S1.

For all datasets, binary methylation states are obtained by assigning a positive (methylated) label when $\frac{\#(\text{reads}_{\text{positive}})}{\#(\text{reads}_{\text{total}})} \geq 0.5$. We use holdout validation to test the performance of the models. For all datasets and experiments, chromosome 5 and 10 constitute the validation and test set, respectively. All other chromosomes are used in training. More instructions on how to obtain and preprocess the

Table 1. Performance comparison of CpG Transformer with other methods

Dataset	# Cells	Sparsity (%)	ROC AUC			PR AUC		
			DeepCpG	CaMelia	CpG Transformer	DeepCpG	CaMelia	CpG Transformer
Ser	20	77.8	90.21	90.22	**91.55**	92.77	92.86	**93.87**
2i	12	77.9	84.80	83.02	**85.77**	71.69	68.87	**73.56**
HCC	25	88.5	96.89	97.42	**97.96**	92.58	94.10	**95.19**
MBL	30	90.8	88.22	89.17	**92.49**	87.61	87.6	**91.80**
Hemato	122	98.4	88.85	89.16	**90.65**	95.60	95.84	**96.43**

Note: Sparsity is defined as the percentage of entries in the methylation matrix that are unobserved. Best performers are indicated in bold. The reported metrics are computed for all cells together.

datasets, as well as their corresponding reference genomes, are available on the GitHub page of CpG Transformer.

2.5 Models and training

CpG Transformer is compared to two competing methods: DeepCpG (Angermueller *et al.*, 2017) and CaMelia (Tang *et al.*, 2021). A comparison with Melissa is not considered since the method is described as complementary to whole-genome imputation methods (Kapourani and Sanguinetti, 2019). For all datasets, the default hyperparameters of CaMelia and DeepCpG are used. We note that performances may vary by choosing or tuning alternative hyperparameters for every dataset. Our choice is motivated by the idea that practitioners will also typically not tune hyperparameters. To ensure a fair comparison, all models are trained using the same data preprocessing and splits. Due to this, performances are expected to deviate slightly from those reported in their respective manuscripts. Considering reproducibility concerns, a full list of differences in our implementations of DeepCpG and CaMelia is given in Supplementary Section S1.

A separate CpG Transformer with identical hyperparameters (obtained by manual tuning on the Ser dataset) is trained for every dataset on 2 V100 GPUs using Adam as optimizer (Kingma and Ba, 2014). A learning rate of $5 \cdot 10^{-4}$ with linear warmup over the first 1000 steps is used. The learning rate is multiplicatively decayed by a factor 0.9 after every epoch. Models are trained for a maximum of 100 epochs, with early stopping after no validation loss decrease has been observed for 10 epochs. The model arising from the epoch with the best validation loss is kept as final model. A dropout rate of 0.20 on elements of the attention matrix is employed during training. Batches are constructed by slicing the $n \times m$ methylation matrices vertically into $n \times b$ bins with $b = 1024$ CpG sites each. One such a bin makes up a batch. For every batch, the number of sites that are masked or randomized equals 25% the number of columns in the bin for all datasets. This masking percentage is chosen considering that a large proportion of the input already consists of masked ? tokens. For the Hemato dataset, we additionally randomly subsample 32 rows (cells) every training batch to reduce complexity and increase training speed. Finally, because random masking negatively biases evaluation, test performance is measured by masking every methylation site in the dataset separately in smaller batches. (Note that this is only necessary to fairly compare imputation performance on all available labels. In practice, inference would be performed without masking.)

3 Results

3.1 Imputation performance

To benchmark CpG Transformer, we evaluate against one competing deep learning method, DeepCpG (Angermueller *et al.*, 2017), and one traditional machine learning method, CaMelia (Tang *et al.*, 2021). The resulting imputation performances in terms of area under the receiver operating characteristic curve (ROC AUC) and area under the precision–recall curve (PR AUC) for all datasets are shown in Table 1. CpG Transformer consistently outperforms existing models on all datasets. As a trade-off, CpG Transformer roughly takes 2 times longer to train than DeepCpG and CaMelia with default hyperparameters (Supplementary Table S2). We detail a way to reduce this training time in Section 3.3. Cell-specific performance evaluation (Supplementary Fig. S1) shows that CpG Transformer is, out of all cells, only outperformed by competing methods for a single Hemato cell. Furthermore, CpG Transformer consistently outperforms DeepCpG and CaMelia in a variety of genomic contexts (Supplementary Fig. S2). The performance gain is most pronounced in contexts typically associated with higher cell-to-cell variability, such as CpG islands, regulatory elements and histone modification marks (Suzuki and Bird, 2008), demonstrating CpG Transformer's ability to encode relevant cell heterogeneity.

Table 2. Ablation study on Ser dataset.

Model	ROC AUC
Original	91.55
Without cell emb.	84.82
Without CpG emb.	71.18
Without DNA emb.	91.01
Without positional enc.	90.49

Note: The original model is compared to four models for which one type of input is removed.

A small ablation study (Table 2) on the Ser dataset shows the importance of the different inputs to the model. The original model is compared to four models, each trained and evaluated in a scenario where one specific input is left out: $\boldsymbol{h}_{i,j}^{\text{CpG}}$, $\boldsymbol{h}_i^{\text{cell}}$, $\boldsymbol{h}_j^{\text{DNA}}$ or the positional encodings. Without the CpG embedding, the model can only rely on cell identity and the DNA contexts of their own and neighboring CpG sites. The model without this embedding displays the lowest performance, illustrating the key importance of dependencies between methylation states for their prediction. Without cell embeddings, cell identity is lost and the prediction for every site is the same for all cells. As the second-most important input for the Ser dataset, this embedding highlights CpG Transformer's capability to exploit cell heterogeneity. Without positional encodings, the model has no way of knowing how far away two CpG sites are from each other. Since column-wise correlation between CpG sites decreases with distance (Cokus *et al.*, 2008), their role is to inform the effect of genomic distance on the degree of correlation in a flexible way. In practice, a minimal but noticeable effect of this encoding on the Ser dataset is observed. Consistent with the findings of DeepCpG (Angermueller *et al.*, 2017), the DNA embeddings, informing the model of DNA context surrounding CpGs, is indicated as the least important input for imputation of the Ser dataset. Further ablation studies of CpG Transformer hyperparameters (Supplementary Tables S3 and S4) show that scaling the architecture of CpG Transformer up or down does not significantly increase performance.

The performance of all models is heavily dataset-dependent, indicating their varying quality. Since single-cell sequencing experiments suffer from low sequencing depth, we hypothesize that performance is negatively influenced by limited coverage both at the CpG site in question (noisy labels) and in its neighborhood (in terms of number of unobserved entries, termed local sparsity). To test this,

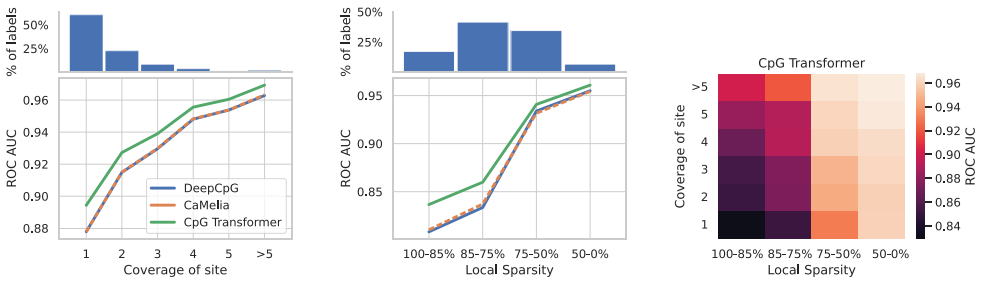

Fig. 3. Dependency of performance on sequencing depth for the Ser dataset. (Left) ROC AUC in function of coverage (# reads) of the label. On top of the plot the percentage of labels belonging to each bin is shown. (Middle) ROC AUC in function of local sparsity (defined as the percentage of unobserved entries in the local window used for prediction). (Right) ROC AUC in function of both factors for CpG Transformer. The biggest gradient in performance is observed for the local sparsity direction

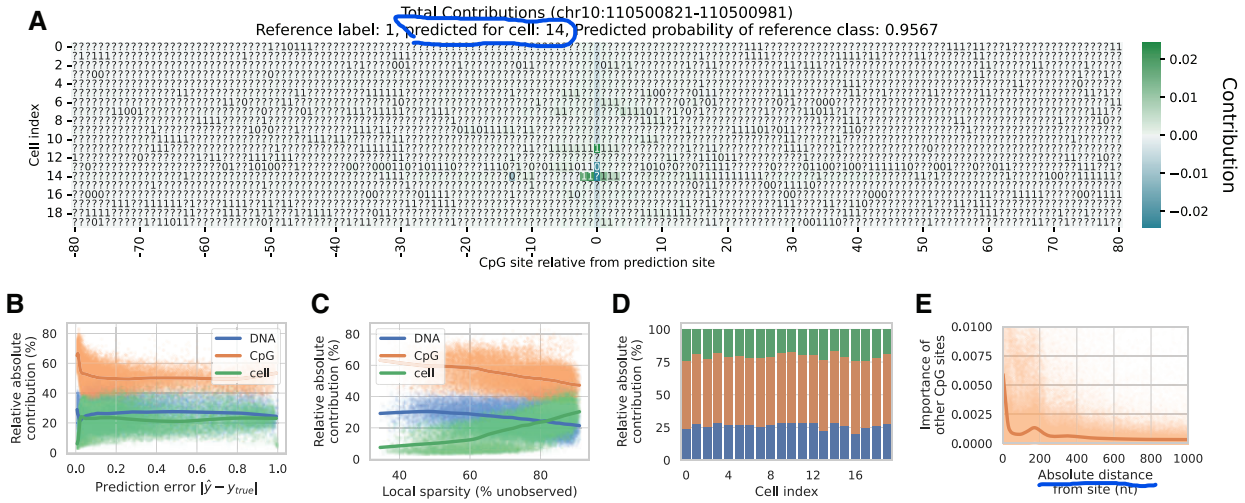

Fig. 4. Integrated Gradients interpretation for the Ser dataset. All trend lines are obtained with LOWESS (Cleveland, 1979). (**A**) Example Integrated Gradients contribution plot. The contributions for the prediction of a randomly chosen CpG site in cell 14 are shown. The true label of the site was originally 1, but is masked in the input here. Matrix entries with a positive and negative contribution are colored in green and blue, respectively. (**B**) Relative absolute contribution of the three input embeddings in function of prediction error. (**C**) Relative absolute contribution in function of local sparsity. (**D**) Relative absolute contribution stratified per cell. (**E**) Contribution of observed CpG sites in function of their absolute distance to the prediction site

the performance of the Ser dataset in function of these factors is plotted (Fig. 3). Similar plots for the other datasets are shown in Supplementary Figure S3. It is observed that CpG sites covered by a smaller number of reads have a less-confident label, resulting in negatively biased performance at evaluation. In addition, CpG sites with a higher local sparsity are harder to predict, presumably due to providing a noisier estimate of local methylation profiles. By making a heatmap of performance in function of both these factors, it is observed that local sparsity is most causal of lower predictive performance. In the context of these experiments, we note that a perfect imputation performance is realistically unattainable given the inherent noise in single-cell methylation datasets.

3.2 Model interpretation

Because CpG Transformer recombines information from CpG sites in a general way, it lends itself well to model interpretation methods. Here, we aim to attribute model predictions to its input features, a problem best approach with gradient-based saliency methods (Bastings and Filippova, 2020). Integrated Gradients (Sundararajan *et al.*, 2017) computes the gradients of the prediction with respect to the input features to measure how every input contributes to prediction. Contributions are obtained by decomposing the difference in prediction of the input sample with an all-zero baseline. For CpG Transformer, contribution scores are obtained for all inputs to the first transformer layer. Since four transformer layers with a window size of 41 are employed, the total receptive field for any prediction constitutes the 161 surrounding CpG sites for all n cells ($n \times 161$). Because Integrated Gradients returns contribution scores for all hidden dimensions, they are summed to obtain a single score for every input matrix entry. An example contribution for the Ser dataset is shown in Figure 4A.

The contributions of the individual matrix entries can be decomposed into those of their constituent embeddings $[\boldsymbol{h}_{i,j}^{\text{CpG}}, \boldsymbol{h}_i^{\text{cell}}, \boldsymbol{h}_j^{\text{DNA}}]$ by backpropagating Integrated Gradients one layer further. In doing so, contribution matrices similar to the one shown in Figure 4A are obtained for all three embeddings (Supplementary Fig. S4). Performing this for 1% of the samples in the test set, it is possible to investigate how embeddings contribute to prediction in different settings. This way, the total contribution of the embeddings in function of the prediction error, local sparsity and cells are obtained for the Ser dataset 4B–D. The same plots for the other datasets are shown in Supplementary Figure S5. We find that CpG embeddings relatively contribute more to predictions when the model is confident (with a small prediction error). In cases where the local sparsity is low (i.e. a low number of unobserved sites), CpG Transformer can rely more on local methylation profiles to make a prediction, increasing the relative importance of CpG embeddings. Between different cells, relative contribution differences are negligible. Figure 4E shows the contributions of neighboring observed CpG sites in function of their distance from the prediction site. A decreasing trend is observed with distance, with one bell-shaped bump appearing at ± 160 nucleotides from the prediction site. This relation has been reported on the same cell types in literature by Song *et al.* (2017), who suggested a relation between nucleosome modifications and DNA methylation.

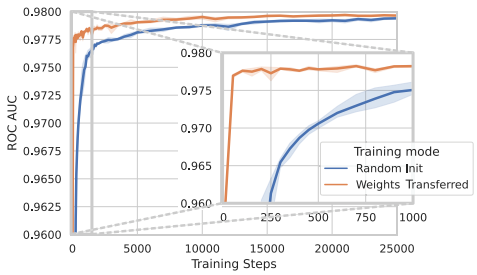

Fig. 5. Transfer learning dynamics on the HCC dataset. Two types of models are trained to impute the HCC dataset: once from a random initialization of weights and once with weights initialized from the model trained on the MBL dataset. Error bands indicate the standard deviation of performances over three runs

3.3 Transfer learning

Because of the generality of CpG Transformer's self-attention mechanism, it is expected that it learns general-purpose representations of DNA methylation dynamics. In this respect, CpG Transformer is envisioned to transfer well to other datasets. In this paper, transfer learning is examined in the context of improved convergence speed when fine-tuning a trained imputation model to impute a new dataset. In other words: using weights from a model previously trained to impute one dataset to initialize a model to impute another dataset. This improved convergence speed is of great interest to practitioners with a limited time and computational budget.

As an experiment, the dynamics of models learning to impute the HCC dataset are investigated (Fig. 5). Two CpG Transformer models are trained in the same way as in previous experiments: once with weights initialized randomly as before and once with weights initialized from the model trained on the MBL dataset. All model weights apart from the cell embeddings are transferred. Both training modes are run in triplicate.

The highest achieved performance of both models is similar (97.96 and 97.98 ROC AUC for random and transferred, respectively), but the models with transferred weights converge substantially faster, reaching an ROC AUC within 0.5% of the best performance after only 50 training steps, whereas the randomly initialized model needs 1000 training steps to reach the same performance, indicating an approximate convergence speed up of 20×. Furthermore, without any fine-tuning steps, transferred models still achieve an ROC AUC of 95.62, indicating the ability of the transformer weights to figure out cell identity from random cell embeddings. Together, these results show CpG Transformer is accessible to researchers wanting to train an imputation model on their own dataset with a limited computational budget.

4 Discussion

CpG Transformer adapts the transformer architecture to operate directly on methylation matrices by combining axial attention (Ho *et al.*, 2019) with sliding window attention (Beltagy *et al.*, 2020), providing a general-purpose way of learning interactions between neighboring CpG sites both within- and between cells. This approach gives rise to many advantages over competing methods. Most simple of all, state-of-the-art imputation performances are obtained over DeepCpG (Angermueller *et al.*, 2017) and CaMelia (Tang *et al.*, 2021). Second, our method lends itself well to interpretation and transfer learning. Finally, because CpG Transformer's model architecture uses learned cell embeddings to encode cell identity in a flexible way, we envision CpG Transformer to scale better to future larger datasets containing diverse cell types.

CpG Transformer allows the prediction of methylation states of thousands of CpG sites in parallel. It does, however, scale quadratically with the number of cells in the dataset. Given the size of the datasets used in this study, this did not pose a problem. For datasets consisting of thousands of cells, however, the application of CpG Transformer as outlined here becomes impossible. In this case, practitioners would need to split their dataset in multiple smaller subsets in which cells are as similar as possible. Alternatively, further extensions of the proposed axial attention could be made in order to allow inputs with a large number of cells. Self-attention sparsity for the column-wise attention operation could, e.g. be enforced through clustered attention (Roy *et al.*, 2021). In doing so, interactions would only be modeled between clustered, closely related cells, instead of between all cells. We consider such extensions to be future work.

The proposed axial attention attends to neighboring sites within a fixed window, irregardless of whether these neighbors have an observed label or not. A possible disadvantage of this strategy may be that, in cases with extreme sparsity, the model may not be able to properly estimate local methylation profiles. In this case, one approach would be not to model interactions within a fixed local window, but instead to attend to the n nearest neighboring observed entries in every cell. This mechanism would attend to a fixed number of observed CpG sites independent of local sparsity. Since such a mechanism would attend to sites far away on the genome in high sparsity settings, its added value is not straightforwardly estimated. Another approach would be to attend only to CpG sites within a fixed genomic width (e.g. 1 kbp). Unlike the previous proposed mechanism, this method would be at an advantage or disadvantage in regions with high or low CpG density, respectively. We consider comparisons with these approaches to be future work.

Model analysis and interpretation show that local sparsity is an obstacle for the performance of imputation models. Figure 3 surprisingly shows that lowly covered sites (whose labels are expected to be more noisy) can be more accurately predicted in a densely covered neighborhood. Some nuances should be made regarding genomic regions that are densely covered but only by a small number of reads for every site. CpG Transformer's masking and randomizing objective (falsely) assumes no structure in noise and missingness. In reality, e.g. one read covering two neighboring unmethylated sites could falsely report methylated signal for both sites if bisulfite treatment failed to convert the corresponding sequence. Hence, lowly covered sites in densely covered neighborhoods may be collectively noisy in the same, nonrandom way. Most contemporary imputation methods, including CpG Transformer, have no way of coping with systematic noise and missingness. In these cases, models will most likely propagate and amplify the noise, potentially compromising biologically relevant results.

Notwithstanding the above-mentioned considerations, given careful evaluation, CpG Transformer can greatly enhance single-cell methylation studies. A cautious practitioner may, e.g. wish to only retain imputations in regions where local sparsity is low and coverage of labels is high. To aid researchers in understanding their imputation results, interpretation methods are introduced. In addition, transfer learning experiments show that CpG Transformer can be used to obtain state-of-the-art imputation performances on a limited time and computational budget.

Data availability

The data underlying this article is freely available from the Gene Expression Omnibus following the identifiers listed in Section 2. Source code for CpG Transformer is freely available at https://github.com/gdewael/cpg-transformer

Funding

This work was supported by the Ghent University [BOFGOA2020000703 to G.D.W.]. W.W. also received funding from the Flemish Government under the 'Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen' Programme.

Conflict of Interest: none declared.

References

Angermueller,C. *et al.* (2016) Parallel single-cell sequencing links transcriptional and epigenetic heterogeneity. *Nat. Methods*, **13**, 229–232.

Angermueller,C. *et al.* (2017) DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning. *Genome Biol.*, **18**, 1–13.
Ba,J.L. *et al.* (2016) Layer normalization. arXiv:1607.06450.
Barabasi,A.-L. and Oltvai,Z.N. (2004) Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.*, **5**, 101–113.
Bastings,J. and Filippova,K. (2020) The elephant in the interpretability room: why use attention as explanation when we have saliency methods? arXiv: 2010.05607.
Beltagy,I. *et al.* (2020) Longformer: the long-document transformer. arXiv: 2004.05150.
Bird,A. (2002) DNA methylation patterns and epigenetic memory. *Genes Devel.*, **16**, 6–21.
Cedar,H. (1988) DNA methylation and gene activity. *Cell*, **53**, 3–4.
Clauwaert,J. and Waegeman,W. (2020) Novel transformer networks for improved sequence labeling in genomics. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1–1.
Cleveland,W.S. (1979) Robust locally weighted regression and smoothing scatterplots. *J. Am. Stat. Assoc.*, **74**, 829–836.
Cokus,S.J. *et al.* (2008) Shotgun bisulphite sequencing of the *Arabidopsis* genome reveals DNA methylation patterning. *Nature*, **452**, 215–219.
Dai,Z. *et al.* (2019) Transformer-xl: attentive language models beyond a fixed-length context. arXiv:1901.02860.
Devlin,J. *et al.* (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.
Di Lena,P. *et al.* (2019) Missing value estimation methods for DNA methylation data. *Bioinformatics*, **35**, 3786–3793.
Elnaggar,A. *et al.* (2020) ProtTrans: towards cracking the language of life's code through self-supervised deep learning and high performance computing. arXiv:2007.06225.
Farlik,M. *et al.* (2016) DNA methylation dynamics of human hematopoietic stem cell differentiation. *Cell Stem Cell*, **19**, 808–822.
Guo,H. *et al.* (2013) Single-cell methylome landscapes of mouse embryonic stem cells and early embryos analyzed using reduced representation bisulfite sequencing. *Genome Res.*, **23**, 2126–2135.
He,K. *et al.* (2016) Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.
He,X. *et al.* (2017) Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pp. 173–182.
Ho,J. *et al.* (2019) Axial attention in multidimensional transformers. arXiv: 1912.12180.
Hou,Y. *et al.* (2016) Single-cell triple omics sequencing reveals genetic, epigenetic, and transcriptomic heterogeneity in hepatocellular carcinomas. *Cell Res.*, **26**, 304–319.
Jiang,L. *et al.* (2019) LightCpG: a multi-view CpG sites detection on single-cell whole genome sequence data. *BMC Genomics*, **20**, 1–17.
Jumper,J. *et al.* (2021) Highly accurate protein structure prediction with alphafold. *Nature*, **596**, 583–589.
Kapourani,C.-A. and Sanguinetti,G. (2019) Melissa: Bayesian clustering and imputation of single-cell methylomes. *Genome Biol.*, **20**, 1–15.
Kingma,D.P. and Ba,J. (2014) Adam: a method for stochastic optimization. arXiv:1412.6980.
Kretzmer,H. *et al.* (2021) Preneoplastic alterations define CLL DNA methylome and persist through disease progression and therapy. *Blood Cancer Disc.*, **2**, 54–69.
Krueger,F. *et al.* (2012) DNA methylome analysis using short bisulfite sequencing data. *Nat. Methods*, **9**, 145–151.
Levy,J.J. *et al.* (2020) MethylNet: an automated and modular deep learning approach for DNA methylation analysis. *BMC Bioinform.*, **21**, 1–15.
Qiu,Y.L. *et al.* (2018) A deep learning framework for imputing missing values in genomic data. *bioRxiv*, 406066.
Radford,A. *et al.* (2018) Improving language understanding by generative pre-training. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.
Rao,R. *et al.* (2021) MSA transformer. *bioRxiv*.
Rives,A. *et al.* (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. USA*, **118**, e2016239118.
Roy,A. *et al.* (2021) Efficient content-based sparse attention with routing transformers. *Trans. Assoc. Comput. Ling.*, **9**, 53–68.
Smallwood,S.A. *et al.* (2014) Single-cell genome-wide bisulfite sequencing for assessing epigenetic heterogeneity. *Nat. Methods*, **11**, 817–820.
Song,Y. *et al.* (2017) Collaborations between CpG sites in DNA methylation. *Int. J. Mod. Phys. B*, **31**, 1750243.
Sundararajan,M. *et al.* (2017) Axiomatic attribution for deep networks. In International Conference on Machine Learning. PMLR, pp. 3319–3328.
Suzuki,M.M. and Bird,A. (2008) DNA methylation landscapes: provocative insights from epigenomics. *Nat. Rev. Genet.*, **9**, 465–476.
Tang,J. *et al.* (2021) Camelia: imputation in single-cell methylomes based on local similarities between cells. *Bioinformatics*, **37**, 1814–1820.
Vaswani,A. *et al.* (2017) Attention is all you need. Advances in neural information processing systems, 5998–6008.
Yu,F. *et al.* (2020) A novel computational strategy for DNA methylation imputation using mixture regression model (MRM). *BMC Bioinform.*, **21**, 1–17.
Zaheer,M. *et al.* (2020) Big bird: transformers for longer sequences. NeurIPS.
Zhang,W. *et al.* (2015) Predicting genome-wide DNA methylation using methylation marks, genomic position, and DNA regulatory elements. *Genome Biol.*, **16**, 1–20.
Zou,L.S. *et al.* (2018) BoostMe accurately predicts DNA methylation values in whole-genome bisulfite sequencing of multiple human tissues. *BMC Genomics*, **19**, 1–15.