DeepZ: A Deep Learning Approach for Z-DNA Prediction

Nazar Beknazarov and Maria Poptsova

Abstract

Here we describe an approach that uses deep learning neural networks such as CNN and RNN to aggregate information from DNA sequence; physical, chemical, and structural properties of nucleotides; and omics data on histone modifications, methylation, chromatin accessibility, and transcription factor binding sites and data from other available NGS experiments. We explain how with the trained model one can perform whole-genome annotation of Z-DNA regions and feature importance analysis in order to define key determinants for functional Z-DNA regions.

Key words Z-DNA, Machine learning, Deep learning, CNN, RNN, Omics data, DNA secondary structures

1 Introduction

Computational detection of Z-DNA regions based exclusively on the information from sequence is a difficult task. Though some sequences with specific patterns (such as GT repeats) are more prone to flip from B- to Z-conformations, the entire set of potential Z-DNA-forming sequences are far larger. Initially the general understanding was that Z-DNA is formed from the alternating purine–pyrimidine repeats, but ChIP-seq data on protein binding with Z-DNA revealed that this is not always the case, and the sequences that at first glance have no definite sequence patterns are shown to adopt Z-DNA conformation [1].

On the other hand, even if a sequence is a potential Z-forming sequence, it does not necessarily serve as a functional Z-DNA element. Often genomic functional elements are surrounded by other functional elements such as histone marks or DNA motifs for transcription factors and other DNA-binding proteins. Combinatorial patterns of different genomic and epigenomic signals should accompany functional Z-DNA regions, and it is a nontrivial task to determine those regions.

Kyeong Kyu Kim and Vinod Kumar Subramani (eds.), *Z-DNA: Methods and Protocols*, Methods in Molecular Biology, vol. 2651,
https://doi.org/10.1007/978-1-0716-3084-6_15,

Another problem that one may face when applying machine learning approach to Z-DNA detection is the scarcity of experimental data. The experiments for detection of Z-DNA structure have many biases (see [2] for a summary); that is why at the time DeepZ was developed, there were only few whole-genome maps available. The first Z-DNA map of the human genome was generated by using Zα domain of the double-stranded RNA editing enzyme ADAR [3]. There were 186 Z-DNA hotspots found, among which 46 hotspots were located in centromeres of 13 human chromosomes. The first ChIP-seq experiment for detection of Z-DNA regions [4] used Zaa protein with two Z-DNA-binding domains. The generated genome-wide map of Z-DNA sites contained 391 regions with the majority of the Z-DNA located in promoter areas. Below we will describe how we overcome the problem of small training data set by considering nucleotide-level approach rather than region-based.

Here we take advantage of machine learning approach that can aggregate information from multiple layers of genome organization together with information on DNA sequence and structure and predict functional elements of interest, here Z-DNA.

Deep learning models were shown to be successful in predicting gene expression [5] and differential gene expression from histone modification signals [6], histone modifications from sequence information and chromatin accessibility data [7], protein–RNA binding preferences from sequence and RNA secondary structure information [8], and promoters and enhancers from histone modification and TF binding ChIP-seq, DNase-seq, FAIRE-seq, and ChIA-PET data [9]. Here we describe a deep learning approach to predict Z-DNA regions incorporating information about sequence, structure, epigenetic code, chromatin accessibility, and transcription factor and RNA polymerase binding sites.

2 The Input Data

The input data is taken from ChIP-seq experiments and usually are represented in the form of intervals (typically, in .bed format). In the original study [10], where we described DeepZ model, we used two Z-DNA data sets: one from ChIP-seq experiment that reported 391 Z-DNA regions [4] and the second data set composed of data from Wu et al. [11] and Kouzine et al. [12]. The data sets should be cleaned from ENCODE blacklist regions [13].

Often, for usage with deep learning methods, the regions of interest are centered and adjusted to the same width and are treated as objects for positive class. In our approach due to the small number of items in the positive class, we propose a different method. Instead of intervals we consider the level of nucleotides where the entire genome is represented by a Boolean array, where

1 is assigned to nucleotides in Z-DNA regions and 0 otherwise. With this approach a minor class will contain enough elements to use in machine learning models (e.g., around 150,000 for 380 sites of Z-DNA regions from ChIP-seq experiments each approximately 400 bp long). The second class is composed from random positions in the genome.

Along with the sequence, the model allows incorporating any additional information. This can be information on physical, chemical, or structural properties of dinucleotides and any omics data from NGS experiments.

We also included in DeepZ model B–Z transition energy that was originally used in Z-Hunt (*see* Table 2 in [14]) and the additional information on histone marks (HM), DNase I hypersensitive sites (DNase-seq), transcription factor (TF), and RNA polymerase (RNAP) binding sites. Methylation variation maps were taken from [15].

In fact, any genomic track can be added as an informational layer (*see* Fig. 1). In the original DeepZ publication, the total set included 1058 markers of which there were 100 histone marks, 947 transcription factor binding sites, 10 RNA polymerase binding sites, and DNase I hypersensitive sites. The full list of features can be found in Supplementary Table S1 in [10].

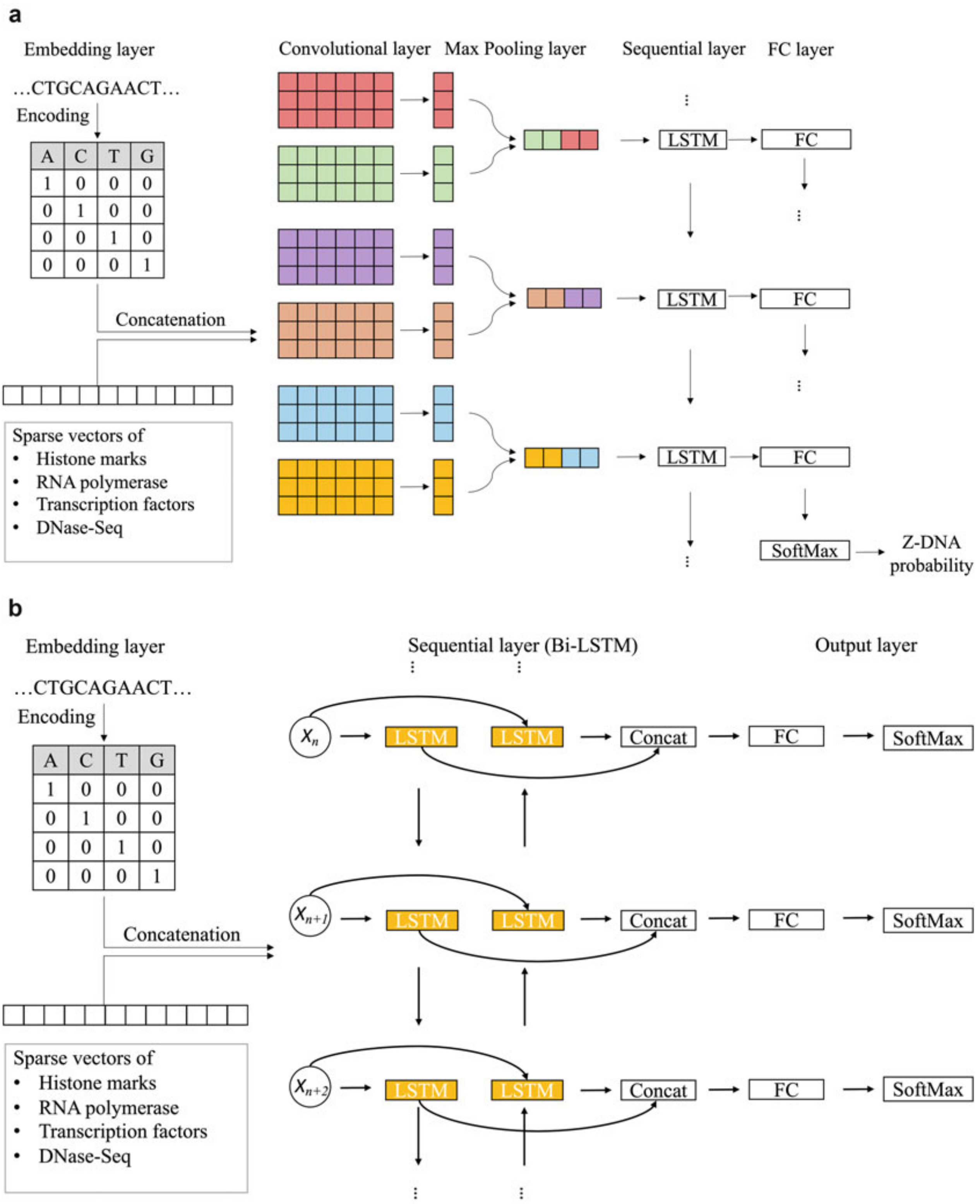
3 Data Compression

Each feature was normalized to the interval [0, 1]. The entire genome was mapped to the matrix of size L × N where L is the size of the genome and N is the number of features used in the model. The total size of the human genome exceeds 3×10^9 nucleotides, and it requires 3 terabytes of RAM to store the entire matrix with each value encoded by 4-byte float.

To overcome this problem, we propose to compress the data with the sparse vector method. The basic idea of the method is to encode the data by two vectors. The first data vector stores directly the values of the encoded vector; the second vector stores the indexes of the values in the encoded vector. This vector supports the following operations: (1) returns standard vector values for a given slice [i, j] and (2) changes vector values on a given slice [i, j].

On the real data—histone data labels—the compression level exceeded 100. Thus, instead of 1 terabyte, about 100 megabytes will solve the task.

For DeepZ model described in [10] with 1058 markers, all the data for human genome took up only about 200 megabytes. Hereby all the input data can run in RAM permanently. This package was implemented in Python 3 using the NumPy library and is available in the repository https://github.com/Nazar1997/Sparse-vector.



A	C	T	G
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

A	C	T	G
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Fig. 1 General schema of deep learning models for Z-DNA prediction. (**a**) CNN-based deep model architectures. (**b**) RNN-based deep model architecture for Z-DNA prediction. The second LSTM cell takes reversed order of data and then concatenates the result with the first LSTM cell with the original order to improve the performance

4 Deep Learning Architectures

The proposed method is based on deep learning approach. We considered three architectures comprising three types of deep neural networks: CNN, RNN, and hybrid CNN–RNN. Comparative analysis performed by us in [10] showed that all three architectures performed relatively well for the task of Z-DNA prediction.

The typical CNN and RNN blocks are presented in Fig. 1. They can have different number of layers. The model ends by a fully connected (FC) block, which also can be represented by more than one layer. A dropout layer can be placed in between FC layers with a probability of every dropout layer set to 0.5. The last FC layer has two output neurons corresponding to two classes.

CNN-Based Architecture This type of DL models consists of only CNN and FC layer blocks (Fig. 1). One and two CNN layers with ReLU activation in between CNN layers were tried. Number of convolutional kernels and kernel size varied from 1 to 17. Stride was set to 1; padding was set to (kernel size − 1)/2, to keep the same size of the output. Every convolutional kernel has 1D conformation. An output of the CNN block is sent to the FC block, where final prediction is made.

RNN-Based Architecture This type of DL models consists of only RNN and FC blocks (Fig. 1). Untouched input is sent to the RNN block. The RNN block consists of the LSTM network with different hyperparameters. We tested one and two LSTM layers, one and bidirectional LSTM with various hidden sizes. Output of the RNN block is sent to the FC block where final prediction is made.

Hybrid CNN–RNN-Based Architecture This type of DL models consists of both RNN and CNN and FC blocks. The input is first sent to the CNN block and then to the RNN block, and the final prediction is made in the FC block. Searching for hyperparameters for each block was the same as described above.

In the original DeepZ publication, all models were trained using RMSprop via backpropagation (RMSprop is the unpublished, adaptive learning rate method proposed by Geoff Hinton). Instead of the full-gradient calculation, the gradient was calculated on a subset of the training set, and model parameters were updated accordingly after each gradient calculation.

5 Train and Test Set

Every chromosome is divided into a set of subsequences. We recommend to avoid generating boundaries of subsequences based on the sites of Z-DNA as it takes place when the functional element is centered in the region (*see* **Note 1**). Every chromosome was evenly cut into pieces with the length of 5000 nucleotides. For train and test sets, we included all subsequences containing Z-DNA and background sequences that do not contain Z-DNA, which were randomly chosen from the entire genome. Randomization was fixed for reproducibility. The number of non-Z-DNA sequences was triple the number of Z-DNA-containing sequences. Training and test sets were stratified and divided in the ratio of 4 to 1. The stratification was based on Z-DNA presence and chromosome number.

6 Whole-Genome Annotation with Z-DNA Regions

Once the model is trained, it can be used to predict novel functional Z-DNA regions. The problem with training DeepZ model was the scarcity of experimental Z-DNA data at the time DeepZ model was developed. To minimize the bias toward the available training set, we implemented procedure similar to five-fold cross validation. We describe it further in detail. The entire data set, which is the entire genome, is divided into 5 folds of equal size, and each fold is stratified by chromosome number and indication of Z-DNA presence/absence (1 or 0). At each consequent step, one fold out of 5 is chosen for a test set and the DeepZ model is trained on the remaining 4 folds. The procedure is repeated five times. In total, five DeepZ models are trained. Each of the five models is used for predictions of the genomic regions outside of the training set. The final prediction is calculated as an average of all five models' predictions, and these are probabilities for a nucleotide to belong to a Z-DNA region. Thus, every nucleotide from every chromosome will have a probability to belong to a Z-DNA-forming region. We assign a nucleotide as belonging to a Z-DNA region if the predicted probability is above a threshold. The threshold is recommended to choose as the value that maximized F1 score on the combined set of all 5 folds (*see* **Note 2**).

This method can assign short DNA regions being Z-DNA that can be located at a short distance from each other. To avoid fragmentation, we combined short regions into longer one based on the rule that all intervals with a gap less than 11 bp can be joined together taking into account that 11 bp is the length of one turn of DNA helix (*see* **Note 3**).

7 DeepZ Model Interpretation

One of the important aspects of machine learning approach is the interpretability of the constructed model. The value of the machine learning model depends on whether it is possible to extract important features that contribute to the model performance. Since RNN architectures are not good for interpretations, we used the best CNN model, which performance was only slightly inferior to the best RNN model. We applied different approaches to interpretation and describe each separately.

CNN was originally developed for image recognition where image is supplied in the form of matrix with pixel values. The CNN model applies different filters (small matrices) to reveal important elements in the image regardless of their position inside the image. Here genomic data is digitalized and represented as a matrix with real values similar to image representation with pixel values. The idea of applying different filters is the same as for images but here the important filters correspond to the recurrent sequence motifs. This methodology of extracting important filters from CNN trained to predict genomic regions of interest was successfully applied in many works including prediction of DNA-binding sites [16] and others [8, 17]. This method uses only sequence information that is converted to the DNA sequence motifs characteristic for regions of interest (*see* **Note 4**).

The second method for getting feature importance from DeepZ model consists in quantifying both positive and negative contribution of each feature from omics data and physical, chemical, or structural properties of DNA. To obtain these values, CNN model should be trained separately with a high regularization penalty.

For image classification task, the proposed method computes the gradient of the class score with respect to the input image [18]. Our method is similar with the difference that the input is a 1D image of the nucleotide sequences. The training of the CNN model is done with an addition of 10^{-3} (or 10^{-2}) weights of L1 regularization in the loss function. L1 regularization has the property of nullifying all unnecessary model weights, and all features with zero weights in the first convolutional layer are further ignored. The nonzero weights of the model are frozen, and the trainable input is passed again to this model. The structure of the model allows limiting the trainable input length to nine nucleotides (Fig. 2). The most distant filter of the second layer is located at a distance of two nucleotides; in turn the most distant nucleotide is located at a distance of two nucleotides from the side filter. Thus, the dependence on the target nucleotide will not exceed four nucleotides to the left and to the right. A sequence of nine elements will completely define one output of the trained CNN model as shown in Fig. 2.

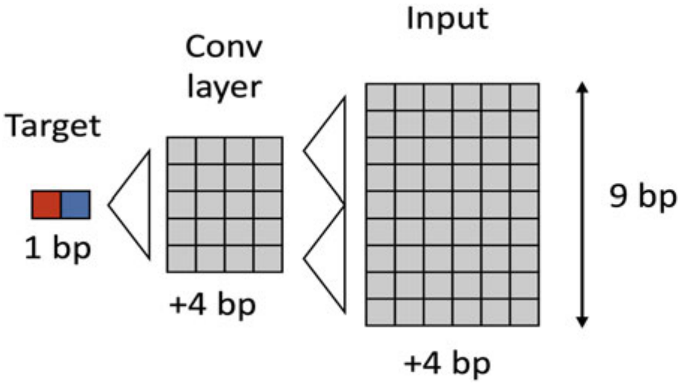


Fig. 2 Model interpretation scheme

However, unlike a neural network, whose weights can take any real value, values of this input can only take values from 0 to 1. In order to find features that from the model's point of view increase the probability of Z-DNA formation, the range of values was set from −1 to 1. This way we can quantify features with both positive and negative contribution. The target function maximizes the predicted probability of becoming a Z-DNA site for the central nucleotide. RMSprop with learning rate 10^{-2} was used for input learning. Input values were mapped to the interval [−1, 1] after every learning iteration.

After the input that maximizes the output of the CNN is found, it is difficult to find a DNA sequence that corresponds to its maximum output, since the sequence itself is encoded by the one-hot encoding method. This means that all four input features depend on each other, and their independent maximization can give an incorrect answer unlike other features. In order to find such a sequence, a separate maximization was performed for the encoded sequence but with additional restrictions. The sum of four features for each nucleotide is equal to one. With these restrictions, the problem is not solved by an ordinary gradient descent, but it is solved using sequential least squares programming. The output is the weight matrix, which is interpretable as a Z-DNA probability (*see* **Note 5**).

Availability The DeepZ model implementation is available at https://github.com/Nazar1997/DeepZ.

8 Notes

1. Because of the small number of the positive class elements and large number of features, the model is prone to overfitting. Every step in the training should be taken with caution of overfitting. A method to fight against overfitting is to avoid

Z-DNA to be centered in the region submitted for training. Otherwise the model will know boundaries and it will result in a target leakage. That is why we partition chromosome in 5000 bp intervals and real Z-DNA regions are randomly distributed over that 5000 bp intervals.

2. We can recommend two strategies to set up a threshold. The one is chosen as the value that maximized F1 (or any other) score on the combined set of all 5 folds as it was done in DeepZ. The other way is to set up the desired number of predicted intervals one expects from the model taking into account model performance metrics. In DeepZ original study, we used the cutoff threshold of 0.343 as it was the value that maximized F1 score on the combined set of all 5 folds.
3. The accumulating data on Z-DNA binding indicate that Z-DNA-binding sites can be shorter than 11 bp. It is up to a researcher to set up the value for the Z-DNA region minimum length. If the value is too small, then the prediction can result in many fragmented regions.
4. Extraction of important filters and their conversion into DNA motifs are done differently from the approaches when only sequence information is used in the form one-hot-encoded matrix. Here we must perform optimization with boundary conditions, and for this task the standard gradient methods must be modified.
5. Feature importance analysis for deep neural network models is a developing field and there does not exist one solution. Different approaches and methods can be employed. In the original DeepZ publications, we applied regularization to the first convolution layer. Later we found that linear regression with regularization also works given that the linear model shows a good performance.

References

1. Li H, Xiao J, Li J, Lu L, Feng S, Droge P (2009) Human genomic Z-DNA segments probed by the Z alpha domain of ADAR1. Nucleic Acids Res 37(8):2737–2746. https://doi.org/10.1093/nar/gkp124
2. Herbert A (2020) ALU non-B-DNA conformations, flipons, binary codes and evolution. R Soc Open Sci 7(6):200222. https://doi.org/10.1098/rsos.200222
3. Herbert A, Alfken J, Kim YG, Mian IS, Nishikura K, Rich A (1997) A Z-DNA binding domain present in the human editing enzyme, double-stranded RNA adenosine deaminase. Proc Natl Acad Sci U S A 94(16):8421–8426. https://doi.org/10.1073/pnas.94.16.8421
4. Shin SI, Ham S, Park J, Seo SH, Lim CH, Jeon H, Huh J, Roh TY (2016) Z-DNA-forming sites identified by ChIP-Seq are associated with actively transcribed regions in the human genome. DNA Res 23:477. https://doi.org/10.1093/dnares/dsw031
5. Singh R, Lanchantin J, Robins G, Qi Y (2016) DeepChrome: deep-learning for predicting gene expression from histone modifications. Bioinformatics 32(17):i639–i648. https://doi.org/10.1093/bioinformatics/btw427
6. Sekhon A, Singh R, Qi Y (2018) DeepDiff: DEEP-learning for predicting DIFFerential gene expression from histone modifications.

Bioinformatics 34(17):i891–i900. https://doi.org/10.1093/bioinformatics/bty612

7. Yin Q, Wu M, Liu Q, Lv H, Jiang R (2019) DeepHistone: a deep learning approach to predicting histone modifications. BMC Genomics 20(Suppl 2):193. https://doi.org/10.1186/s12864-019-5489-4
8. Ben-Bassat I, Chor B, Orenstein Y (2018) A deep neural network approach for learning intrinsic protein-RNA binding preferences. Bioinformatics 34(17):i638–i646. https://doi.org/10.1093/bioinformatics/bty600
9. Li Y, Shi W, Wasserman WW (2018) Genome-wide prediction of cis-regulatory regions using supervised deep learning methods. BMC Bioinform 19(1):202. https://doi.org/10.1186/s12859-018-2187-1
10. Beknazarov N, Jin S, Poptsova M (2020) Deep learning approach for predicting functional Z-DNA regions using omics data. Sci Rep 10(1):19134. https://doi.org/10.1038/s41598-020-76203-1
11. Wu T, Lyu R, You Q, He C (2020) Kethoxal-assisted single-stranded DNA sequencing captures global transcription dynamics and enhancer activity in situ. Nat Methods 17(5): 515–523. https://doi.org/10.1038/s41592-020-0797-9
12. Kouzine F, Wojtowicz D, Baranello L, Yamane A, Nelson S, Resch W, Kieffer-Kwon KR, Benham CJ, Casellas R, Przytycka TM, Levens D (2017) Permanganate/S1 nuclease footprinting reveals non-B DNA structures with regulatory potential across a mammalian genome. Cell Syst 4(3):344–356. e347. https://doi.org/10.1016/j.cels.2017.01.013
13. Amemiya HM, Kundaje A, Boyle AP (2019) The ENCODE blacklist: identification of problematic regions of the genome. Sci Rep 9(1): 9354. https://doi.org/10.1038/s41598-019-45839-z
14. Ho PS, Ellison MJ, Quigley GJ, Rich A (1986) A computer aided thermodynamic approach for predicting the formation of Z-DNA in naturally occurring sequences. EMBO J 5(10): 2737–2744
15. Gao Y, Li L, Yuan P, Zhai F, Ren Y, Yan L, Li R, Lian Y, Zhu X, Wu X, Kee K, Wen L, Qiao J, Tang F (2020) 5-Formylcytosine landscapes of human preimplantation embryos at single-cell resolution. PLoS Biol 18(7):e3000799. https://doi.org/10.1371/journal.pbio.3000799
16. Alipanahi B, Delong A, Weirauch MT, Frey BJ (2015) Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nat Biotechnol 33(8):831–838. https://doi.org/10.1038/nbt.3300
17. Kalkatawi M, Magana-Mora A, Jankovic B, Bajic VB (2019) DeepGSR: an optimized deep-learning structure for the recognition of genomic signals and regions. Bioinformatics 35(7):1125–1132. https://doi.org/10.1093/bioinformatics/bty752
18. Simonyan K, Vedaldi A, Zisserman A (2013) Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv preprint arXiv:13126034