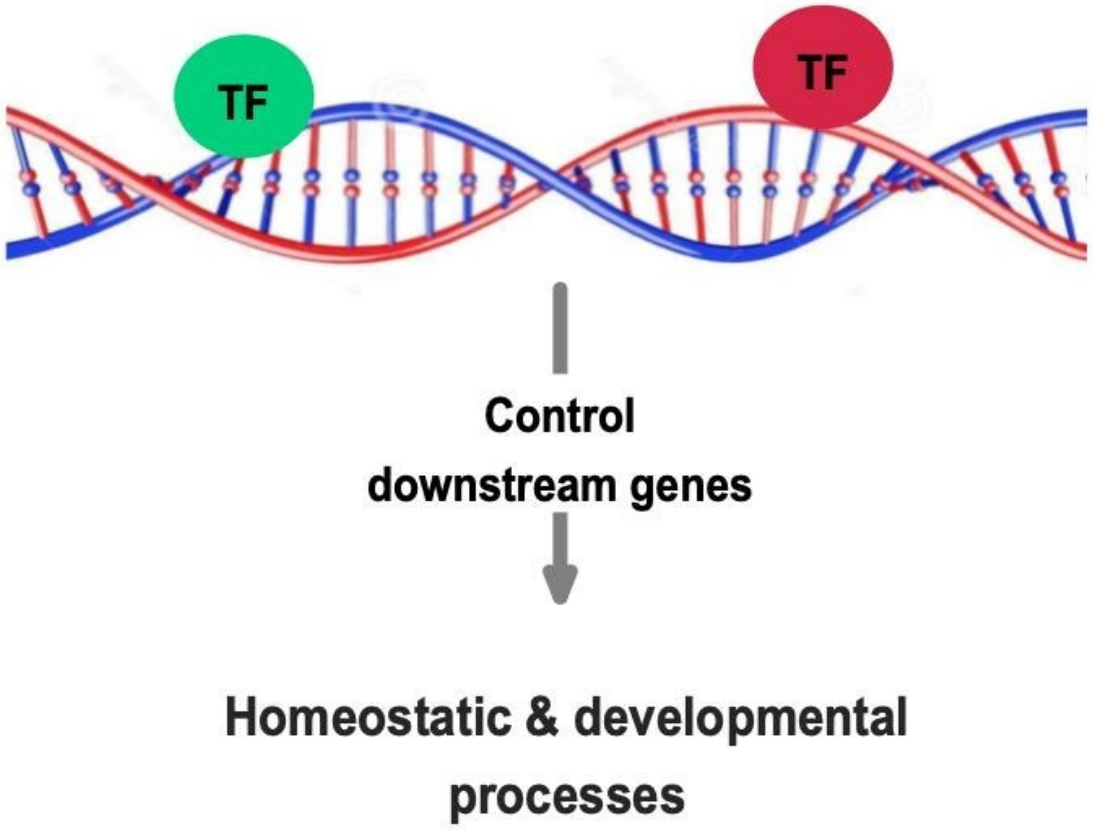
Interpretable Attention Model in Transcription Factor Binding Site Prediction with Deep Neural Network

2020-10-08

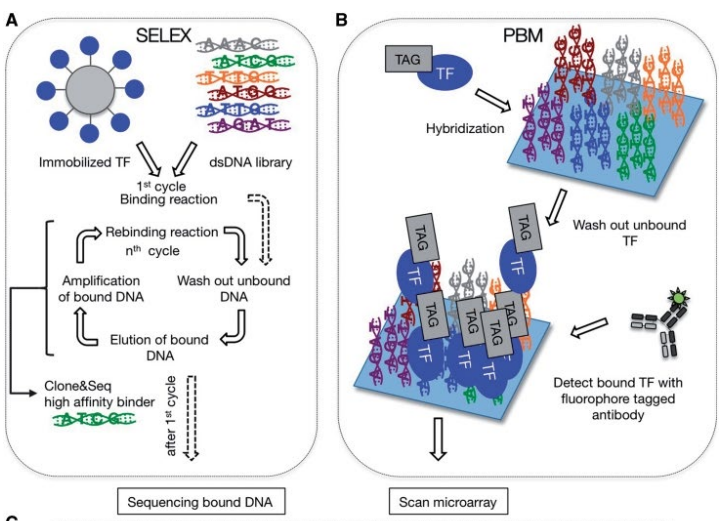
Ping-Han Hsieh

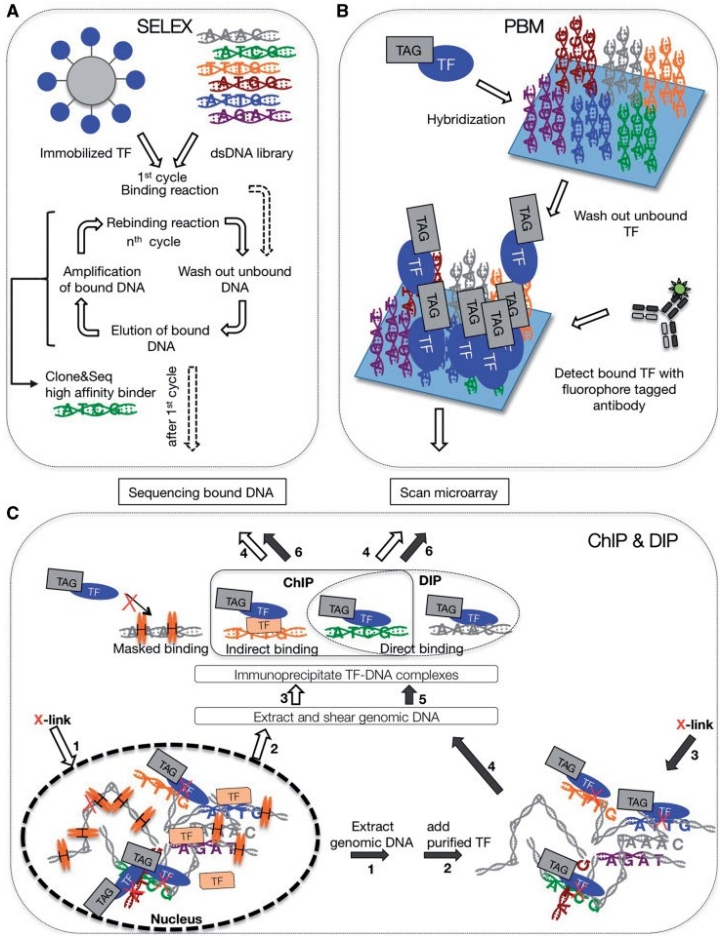
Roza Berhanu Lemma

Introduction



Methods for profiling TF-DNA interactions





Geertz and Maerk, 2010. *Brief Funct Genomics*

Deep Neural Networks (DNNs) for TFBSs Prediction

- Advantageous over classical computational methods
  - Learning high-level feature from extremely large sized data
  - Require less domain knowledge
  - Powerful when little to no prior knowledge of potential binding sites

- Combination of two DNN architectures are used in TFBSs prediction currently (e.g. DeepBind, TFImpute, DeepSea)
  - **RNNs:** better in learning useful information with long-term dependencies
  - **CNNs**: able to extract both local features (genomic signals) and regions
  - **Attention mechanism**: interpretable model by assigning 'attention weights' to different positions according to their importance

Problem Definition

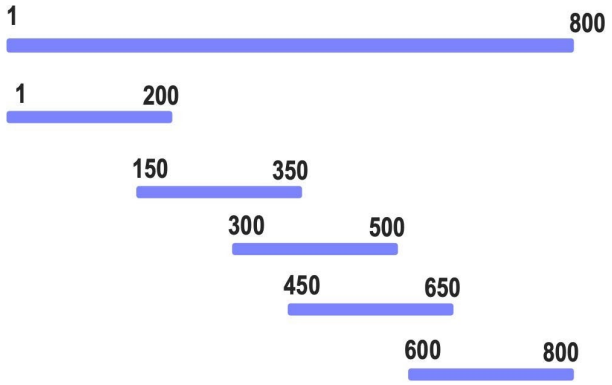
- **Problem:**
  - Complexity of biological factors may influence binding of TFs to DNA
  - Prediction of potential binding site is a difficult task in computational biology

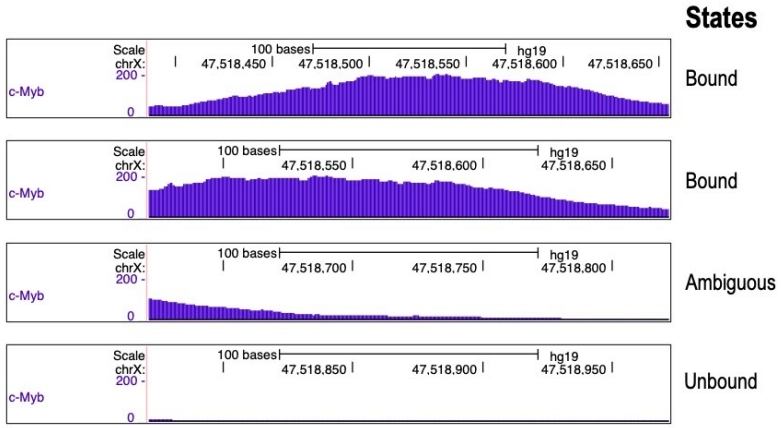
- **Aim:**
  - Introducing the attention mechanism
  - Increase prediction accuracy &
  - Level of interpretability for existing CNN-RNN architecture models

Methods

- **DeepGRN:** a tool that combines attention mechanisms with DNNs
- **ENCODE-DREAM** challenge datasets as input features for model training
  - Ground truth: Binding status of TFs from ChIP-seq data
  - Chromatin accessibility information from DNase-seq data
  - RNA-seq data
- Similar organization of input features used by **FactorNet**
  - DNase-seq and primary sequence information are transformed into sequential features
  - Gene expression and annotations are transformed into non-sequential features

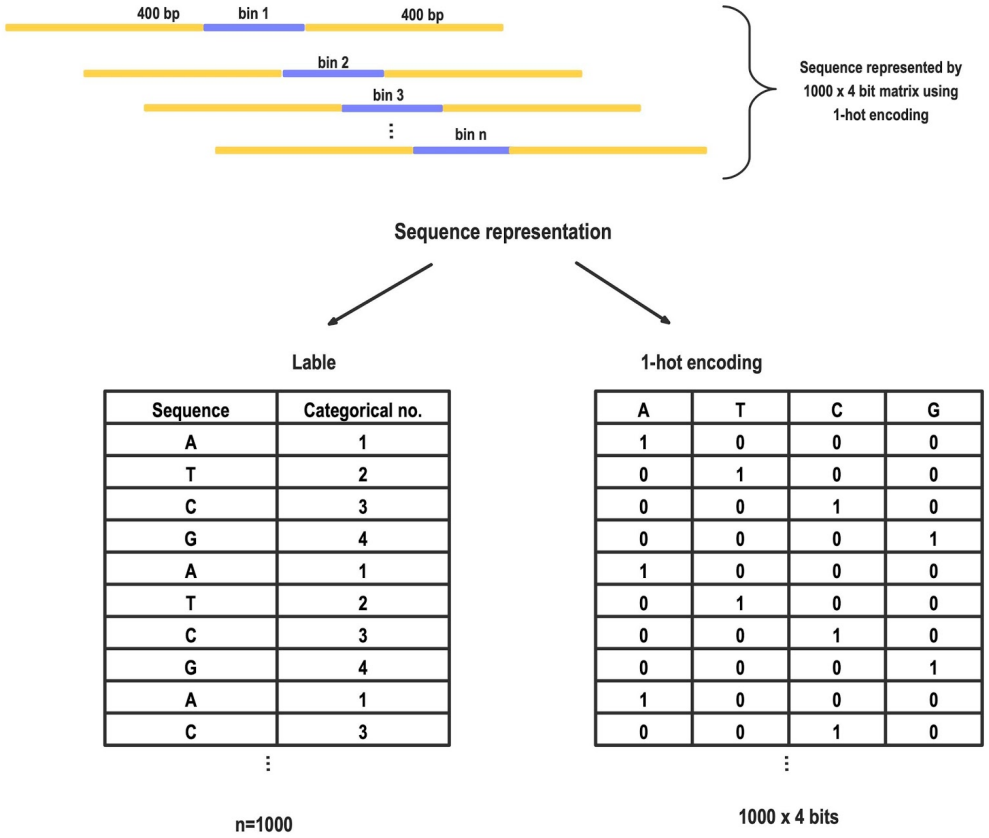
Transcription Factor Binding Data

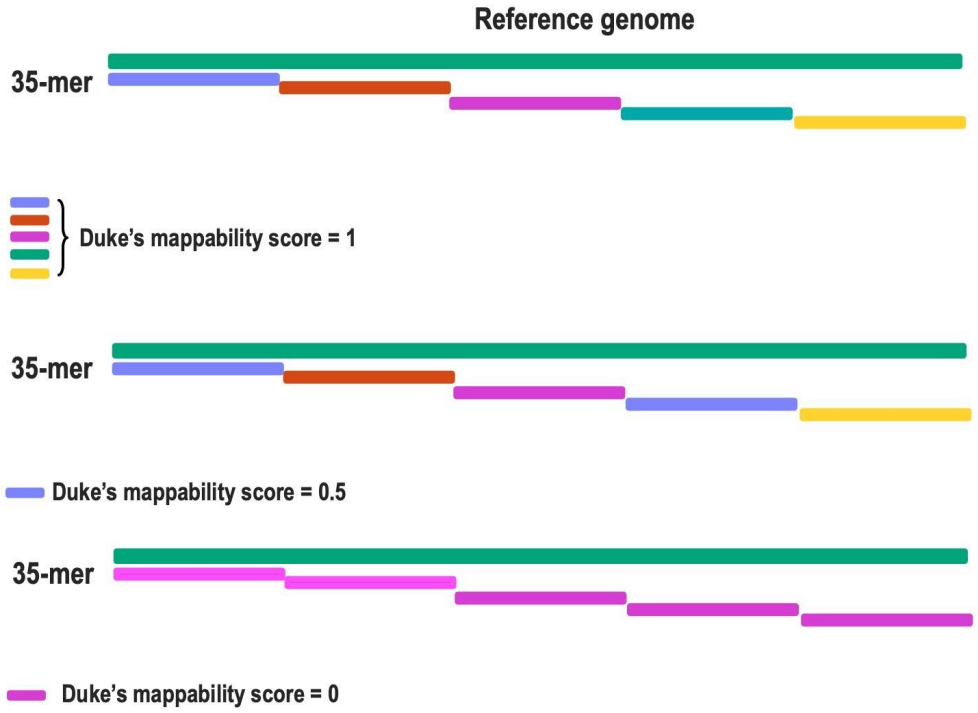




- **IDR (Irreproducibility discovery rate) threshold of 5%**
- Bins overlapping peaks and pass IDR threshold → **Bound** (Positive sites)
- Bins Overlapping peaks but fail IDR threshold → **Unbound** (negative sites)
- The rest → **Ambiguous**

DNA Primary Sequence

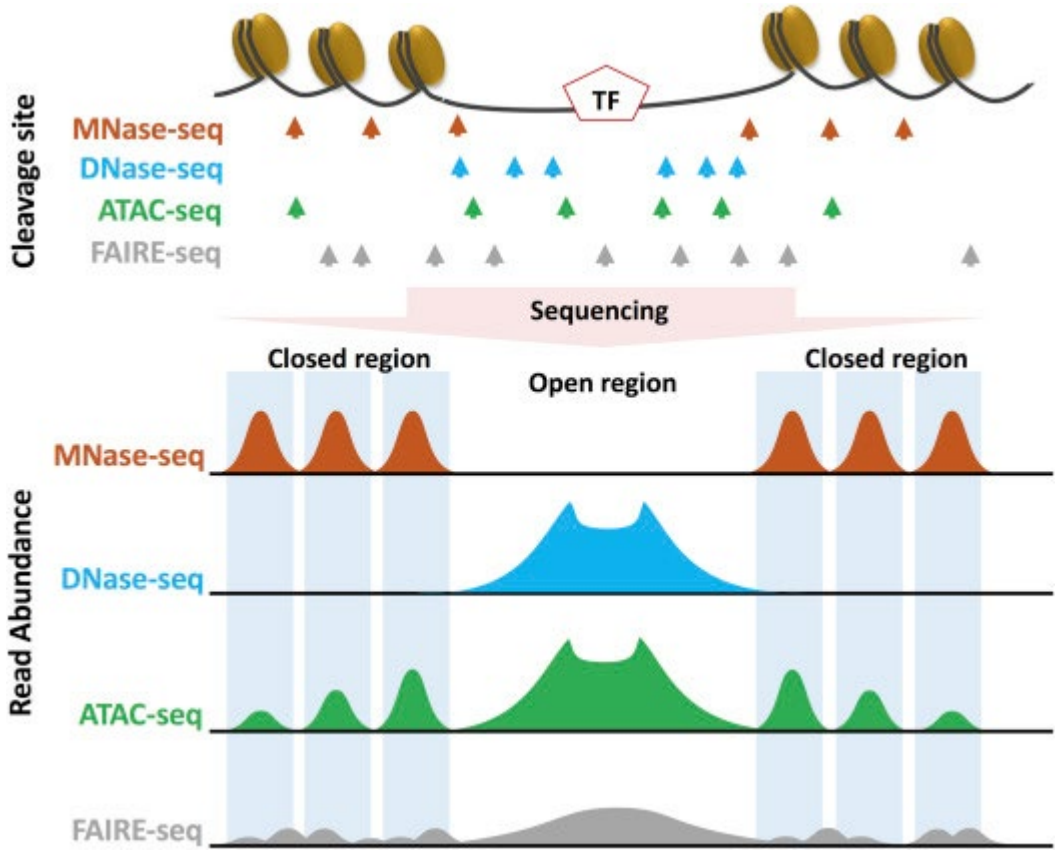




DNase-seq Data

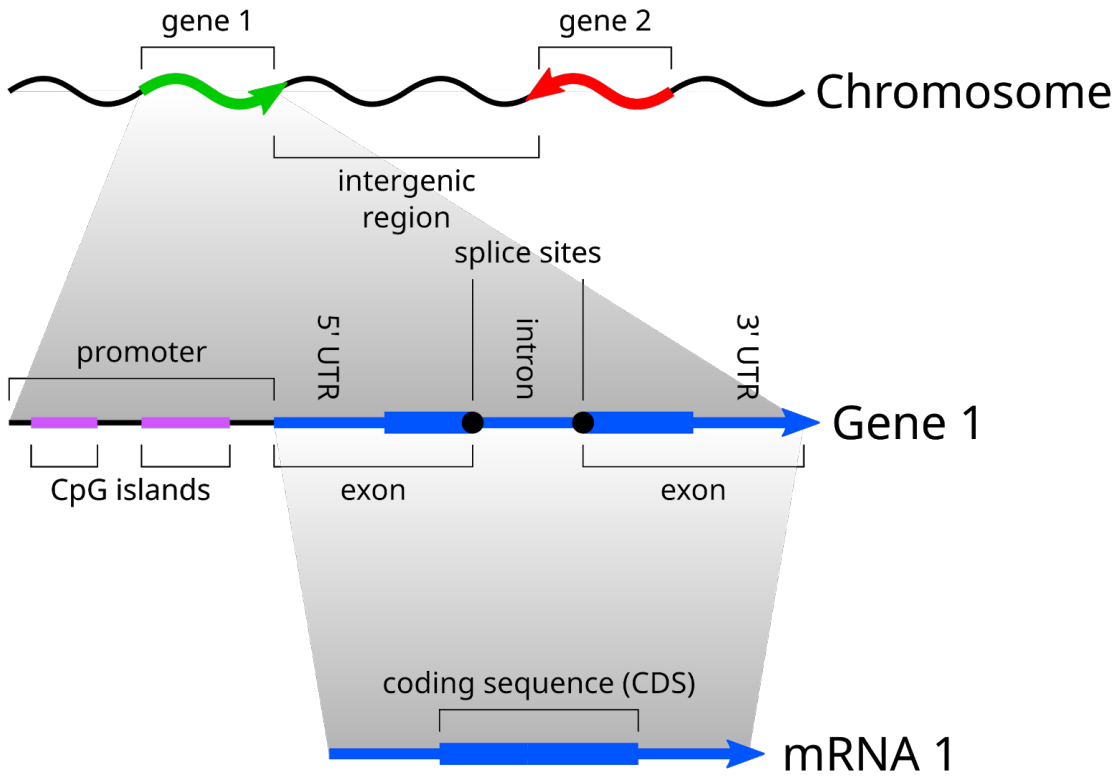
- Obtained Normalized 1x coverage score from BAM files with bin size =1
  - **Represented by 1000 x 1 vector** for each bin

Hsu *et al.* 2018. *Epigenetics in Human Disease (Second Edition)*



Annotation

- Annotation features for each bin is encoded as binary vector of length 6.



Gene Expression

- PCA on TPM normalized count from RNA-seq data provided by the challenge
  - Took the **first eight components** of a cell type as expression scores for all bins for a given cell type
- Both annotation and gene expression non-sequential features are fused into the first dense layer in the deep learning model

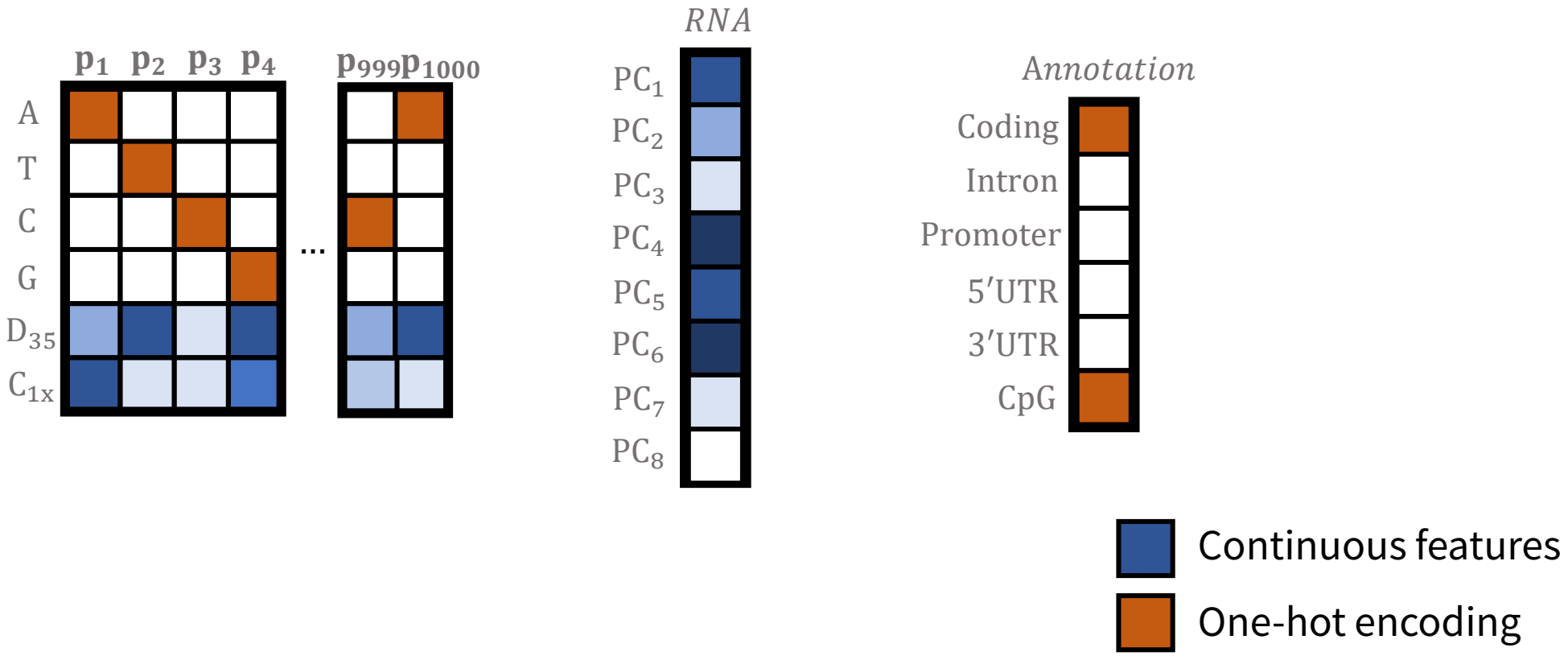
Training, validation and evaluation

- All datasets (chromosome 1-22 and X)
- Chromosome 1, 8 and 21 of the following cell types are used to train the models. Chromosome 11 are used as validation set.

Transcription Factor	Cell Types		
	Training	Optional Model Tuning	Evaluation
CTCF	A549, H1-hESC, HeLa-S3, HepG2, IMR90, K562, MCF-7	GM12878	PC-3, induced pluripotent stem cell
E2F1	GM12878, HeLa-S3		K562
EGR1	GM12878, H1-hESC, HCT116, MCF-7	K562	liver
FOXA1	HepG2	MCF-7	liver
FOXA2	HepG2		liver
GABPA	GM12878, H1-hESC, HeLa-S3, HepG2, MCF-7	K562	liver
HNF4A	HepG2		liver
JUND	HCT116, HeLa-S3, HepG2, K562, MCF-7	H1-hESC	liver
MAX	A549, GM12878, H1-hESC, HCT116, HeLa-S3, HepG2, K562	MCF-7	liver
NANOG	H1-hESC		induced pluripotent stem cell
REST	H1-hESC, HeLa-S3, HepG2, MCF-7, Panc1	K562	liver
TAF1	GM12878, H1-hESC, HeLa-S3, K562	HepG2	liver

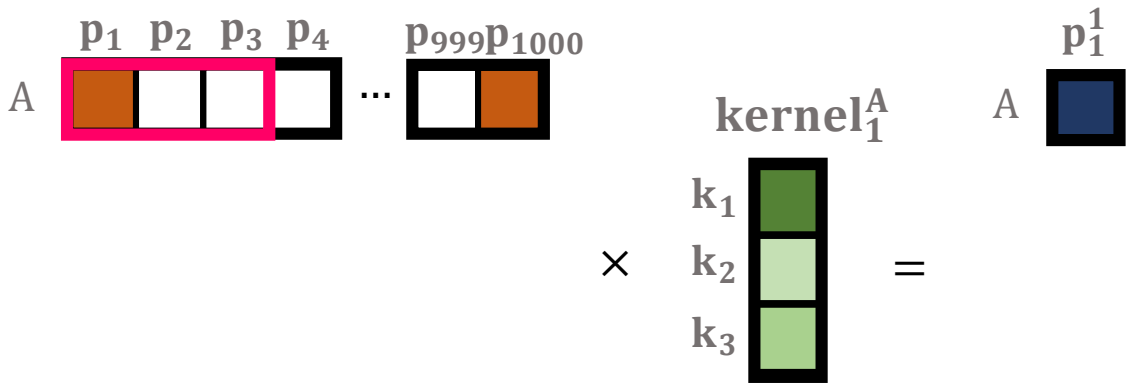
Review on the data

For each bin (instance)



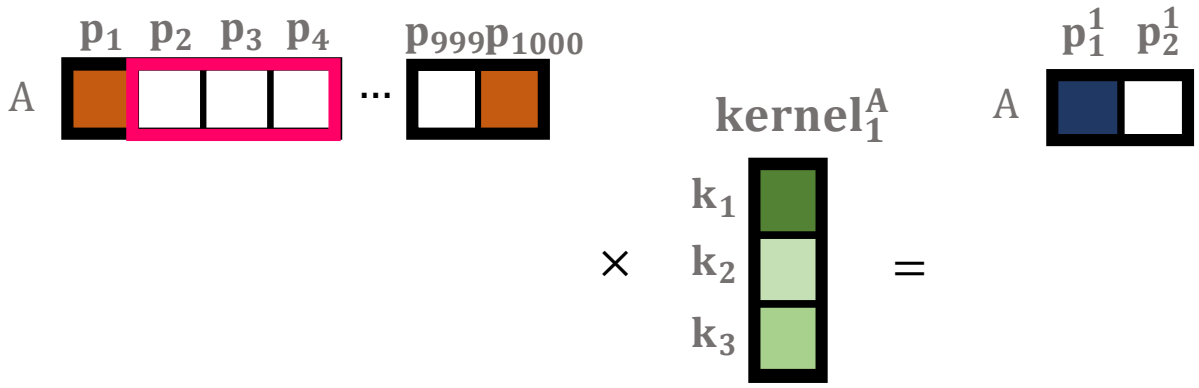
Convolutional neural network

Example, kernel: 3



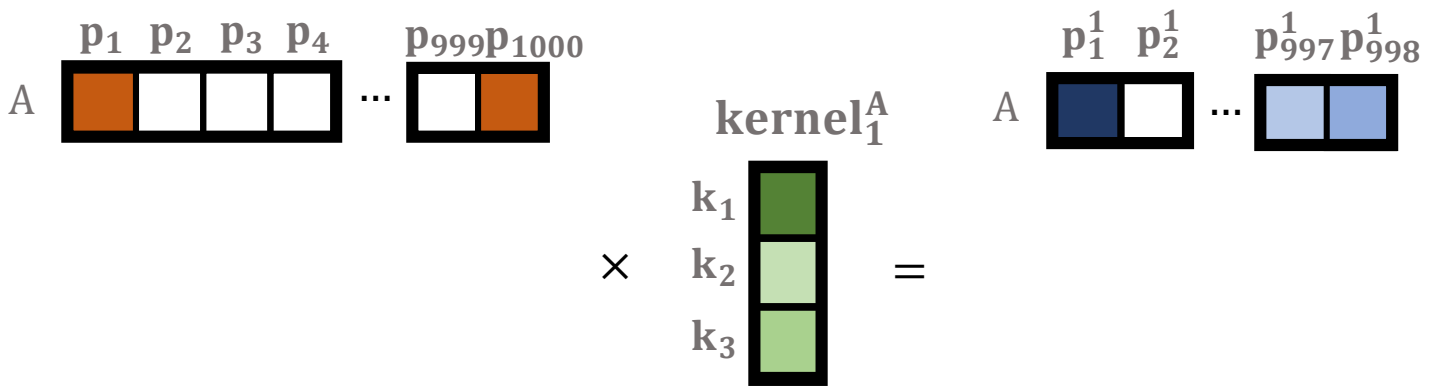
Convolutional neural network

Example, kernel: 3



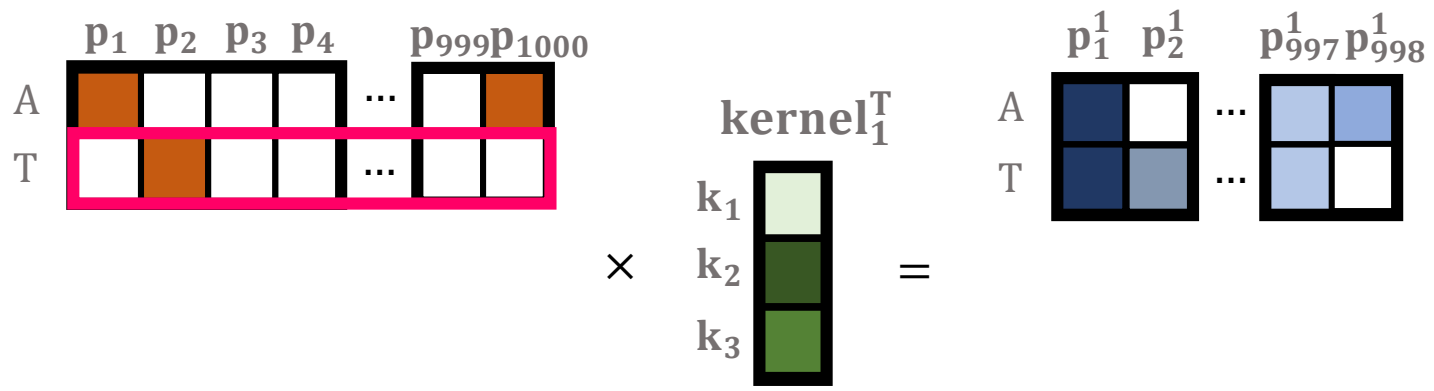
Convolutional neural network

Example, kernel: 3



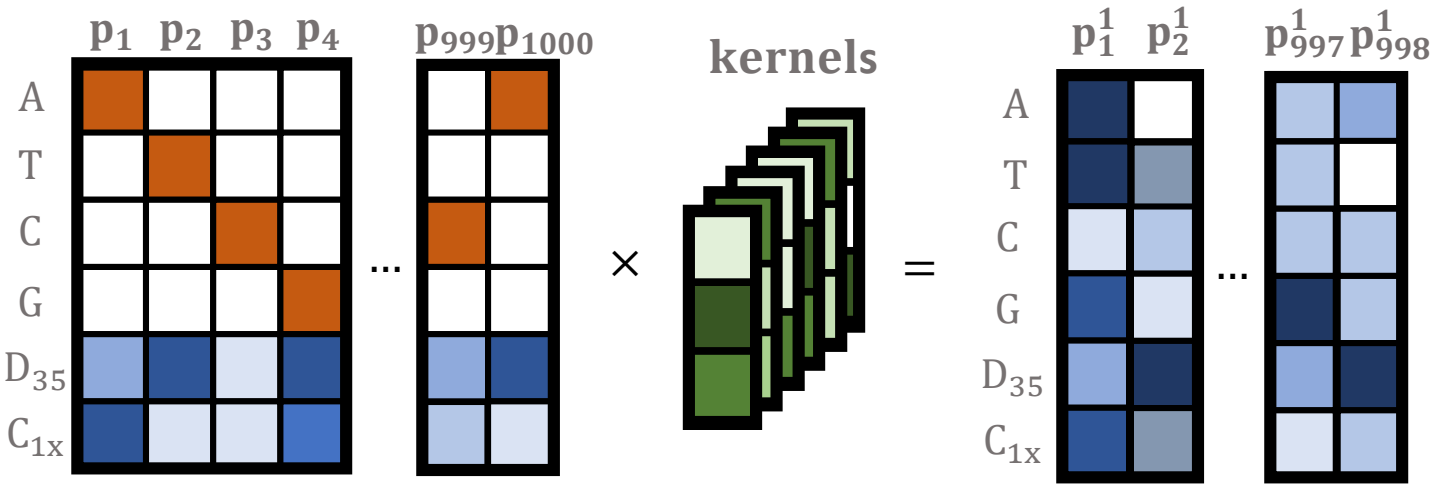
Convolutional neural network

Example, kernel: 3

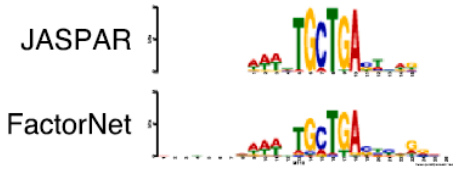


Convolutional neural network

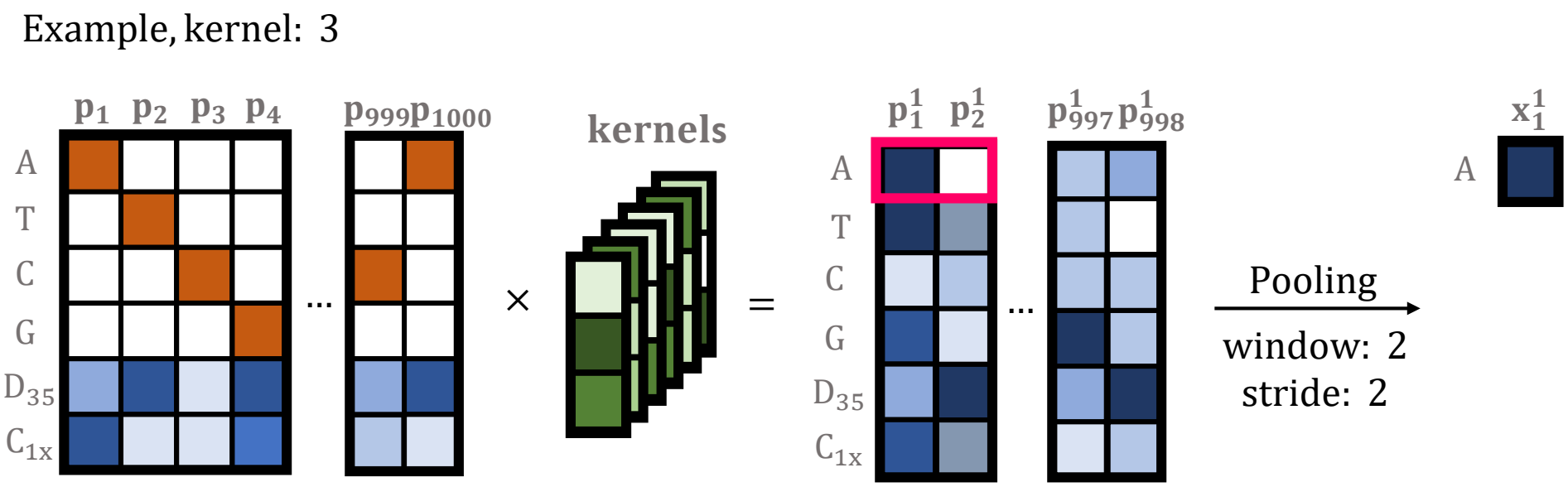
Example, kernel: 3



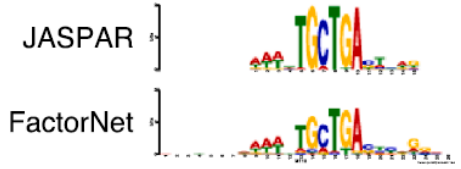
1. The weight of the kernels can be viewed as the unnormalized position weight matrix that is relevant to the prediction.
2. Usually, multiple kernels are used in the model.



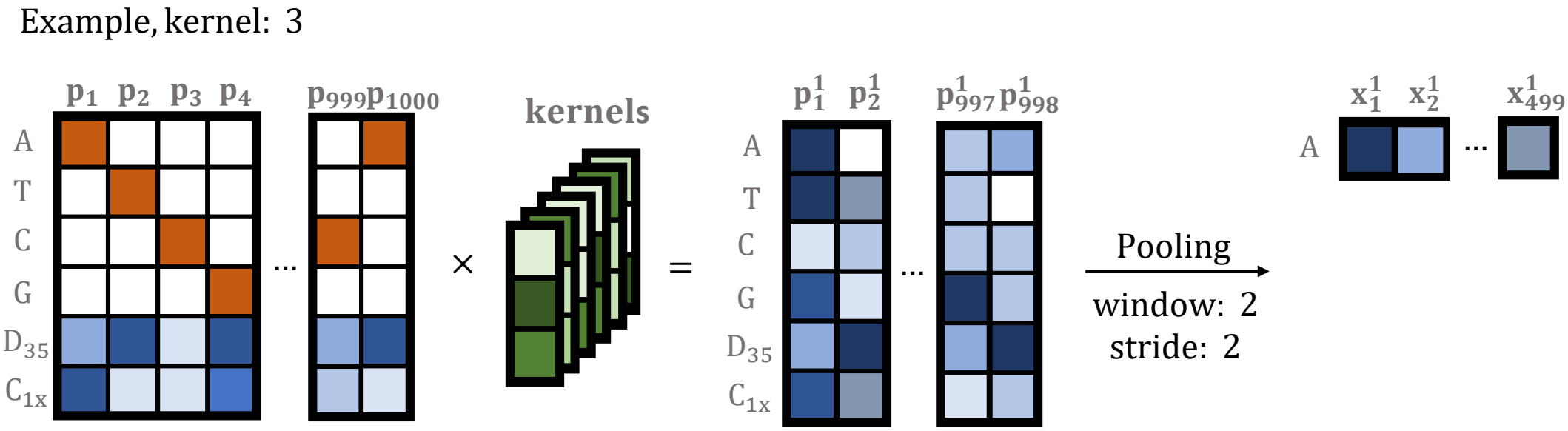
Convolutional neural network



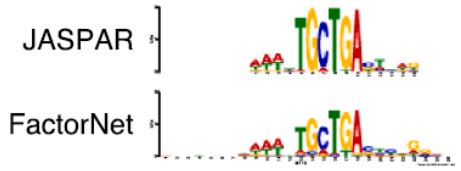
1. The weight of the kernels can be viewed as the unnormalized position weight matrix that is relevant to the prediction.
2. Usually, multiple kernels are used in the model.



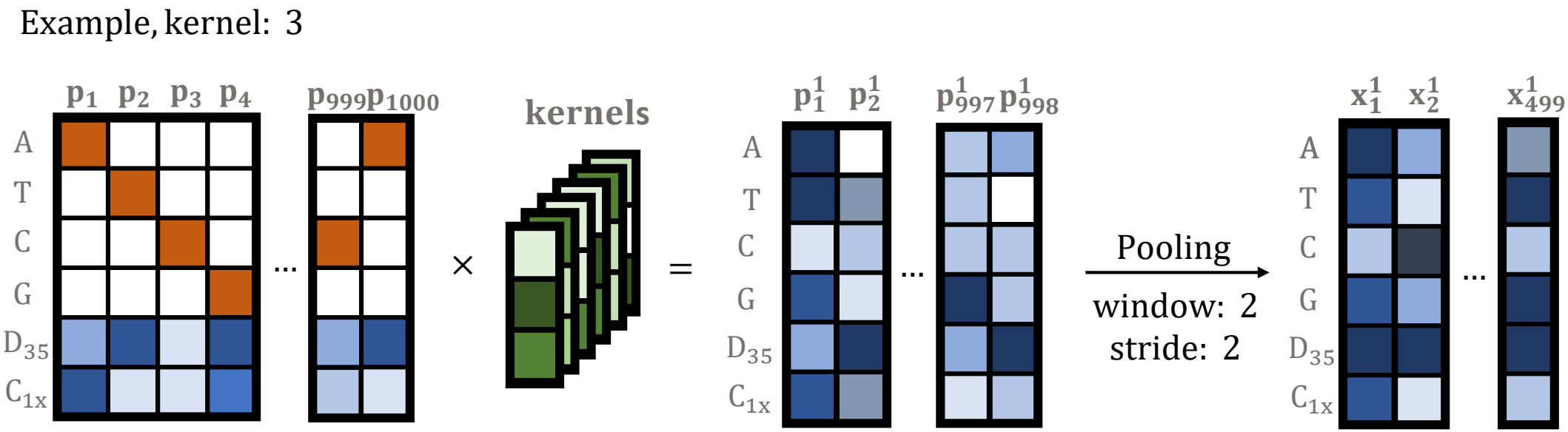
Convolutional neural network



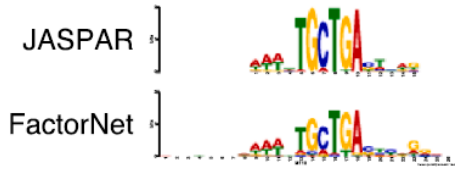
1. The weight of the kernels can be viewed as the unnormalized position weight matrix that is relevant to the prediction.
2. Usually, multiple kernels are used in the model.



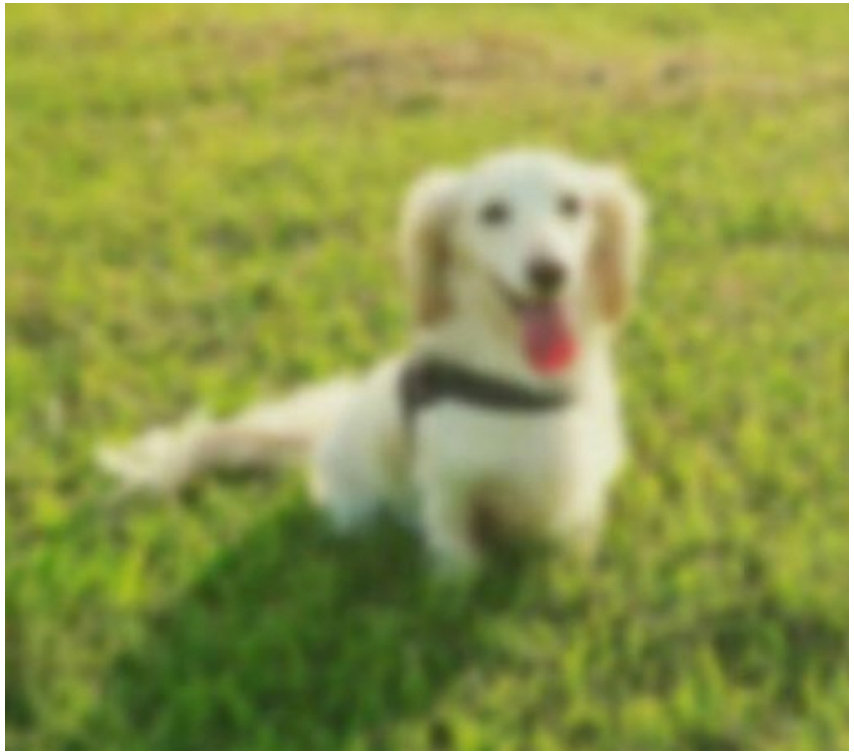
Convolutional neural network



1. The weight of the kernels can be viewed as the unnormalized position weight matrix that is relevant to the prediction.
2. Usually, multiple kernels are used in the model.

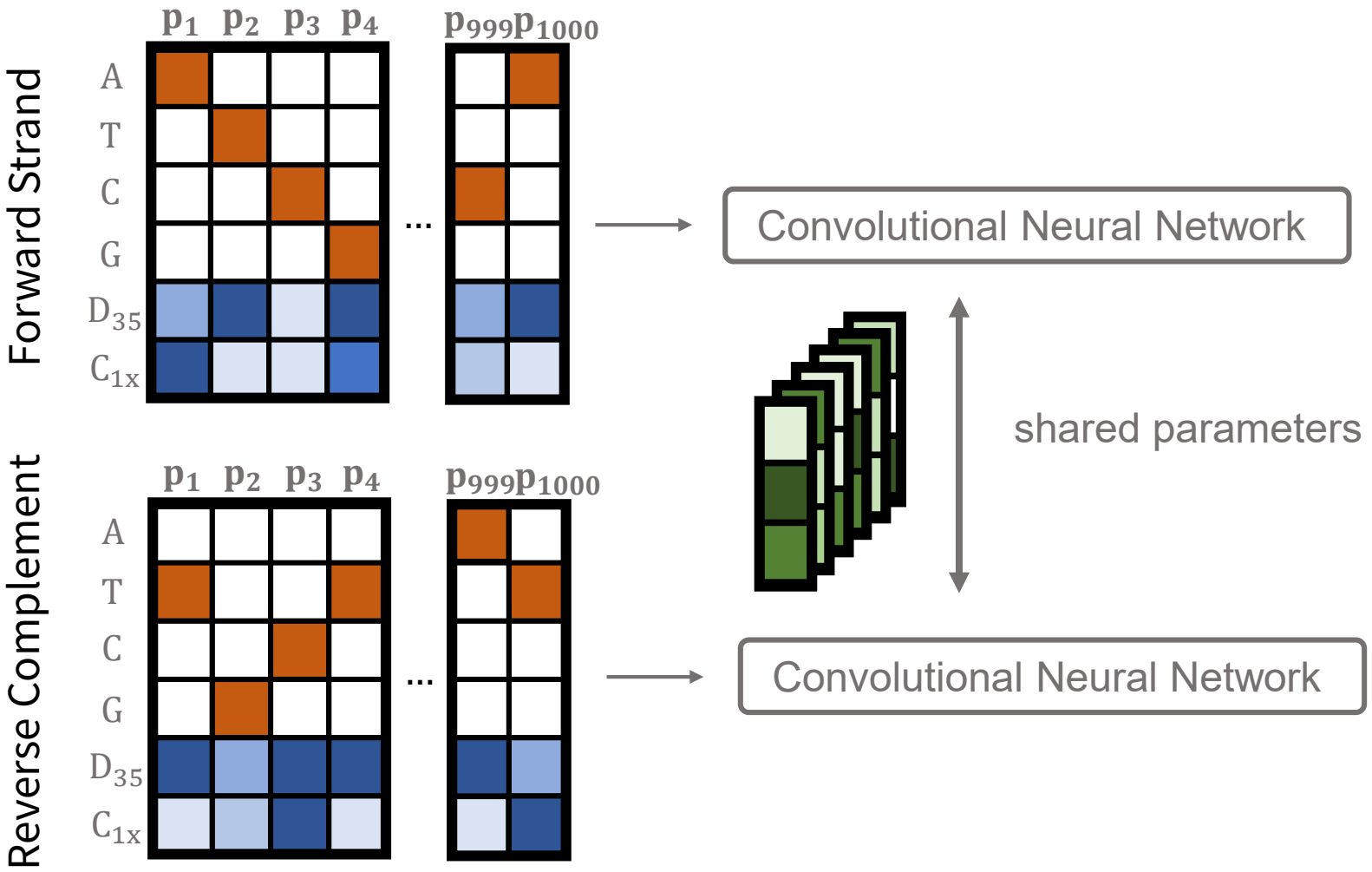


Intuition with pooling

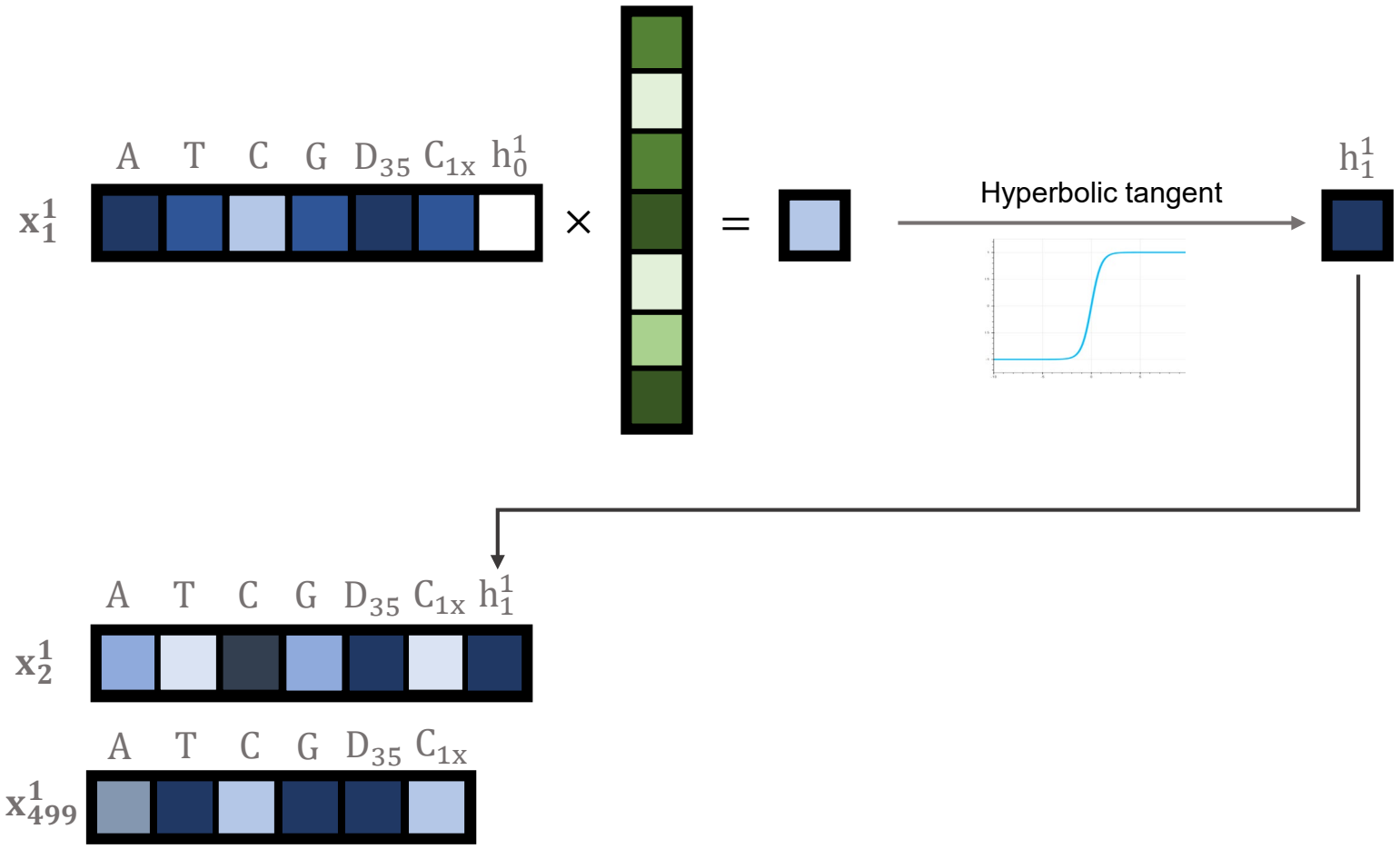


Reduce resolution does not really affect much with the prediction (in terms of the image)

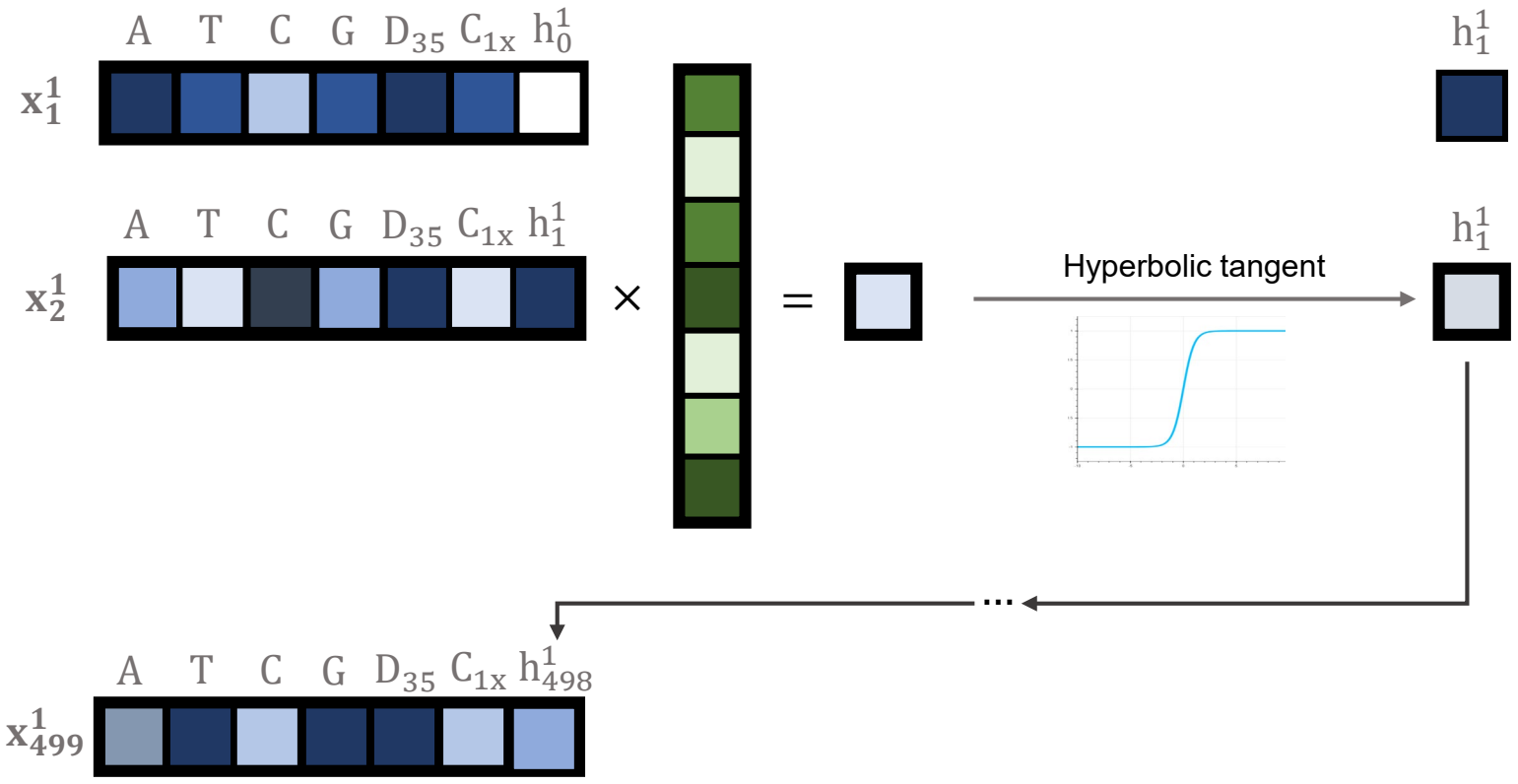
Siamese Network



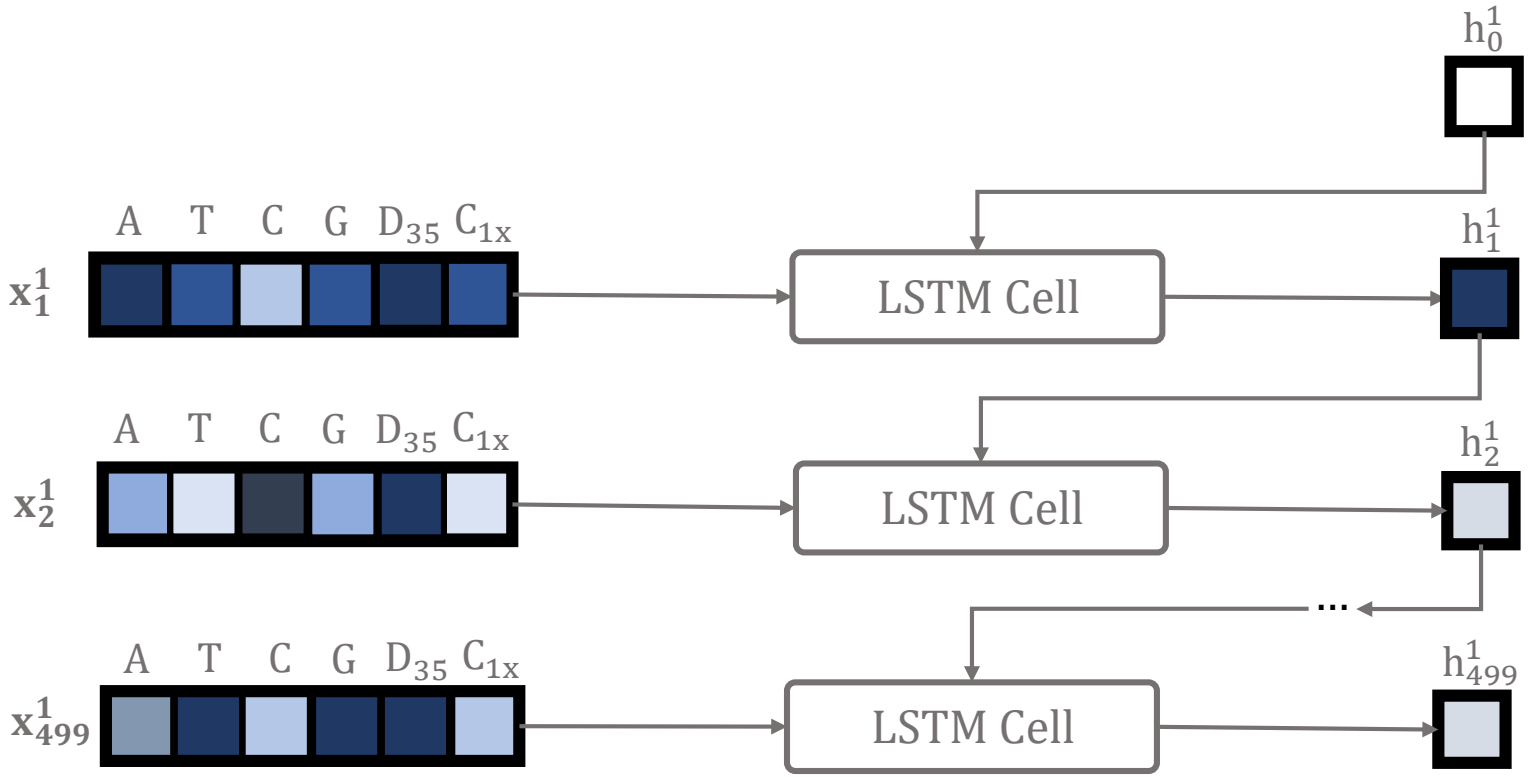
Recurrent Neural Network



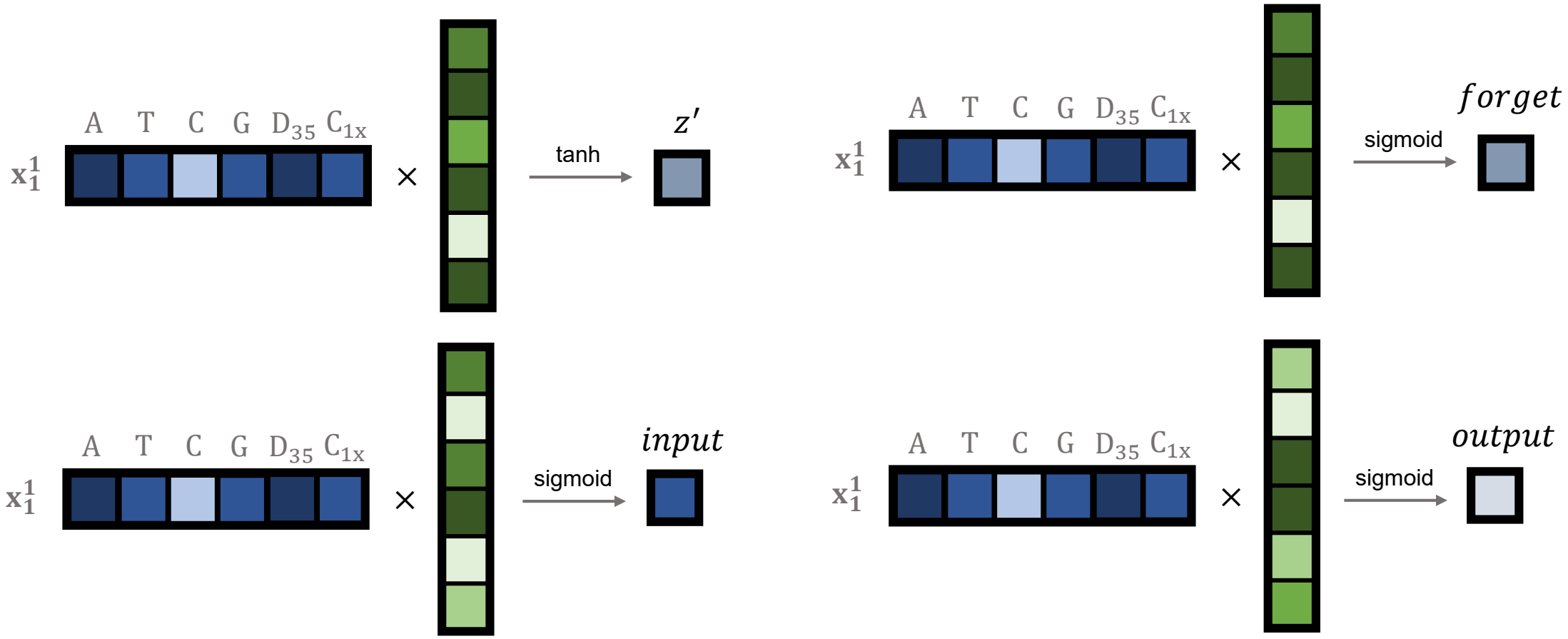
Recurrent Neural Network



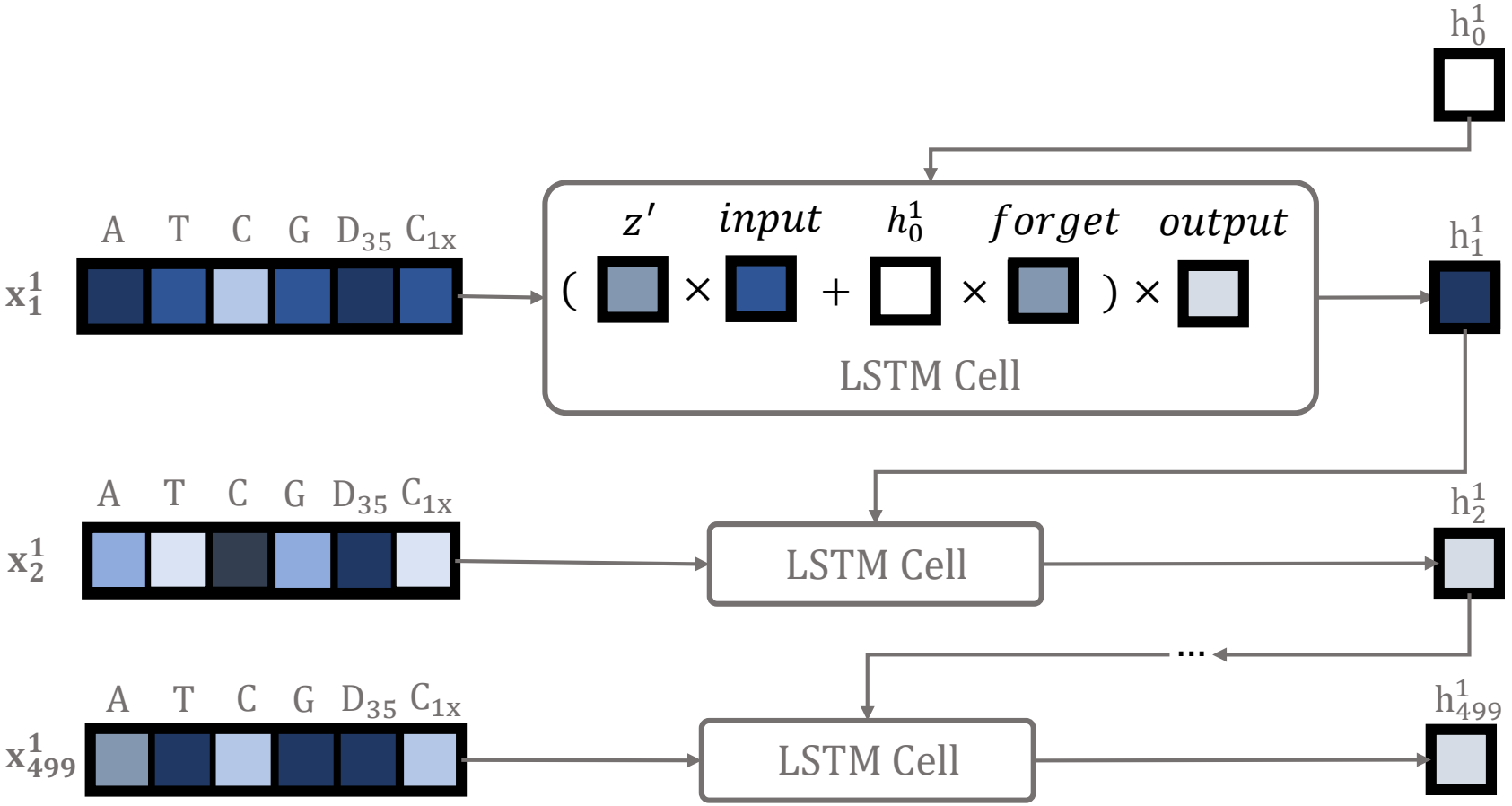
Long short-term memory



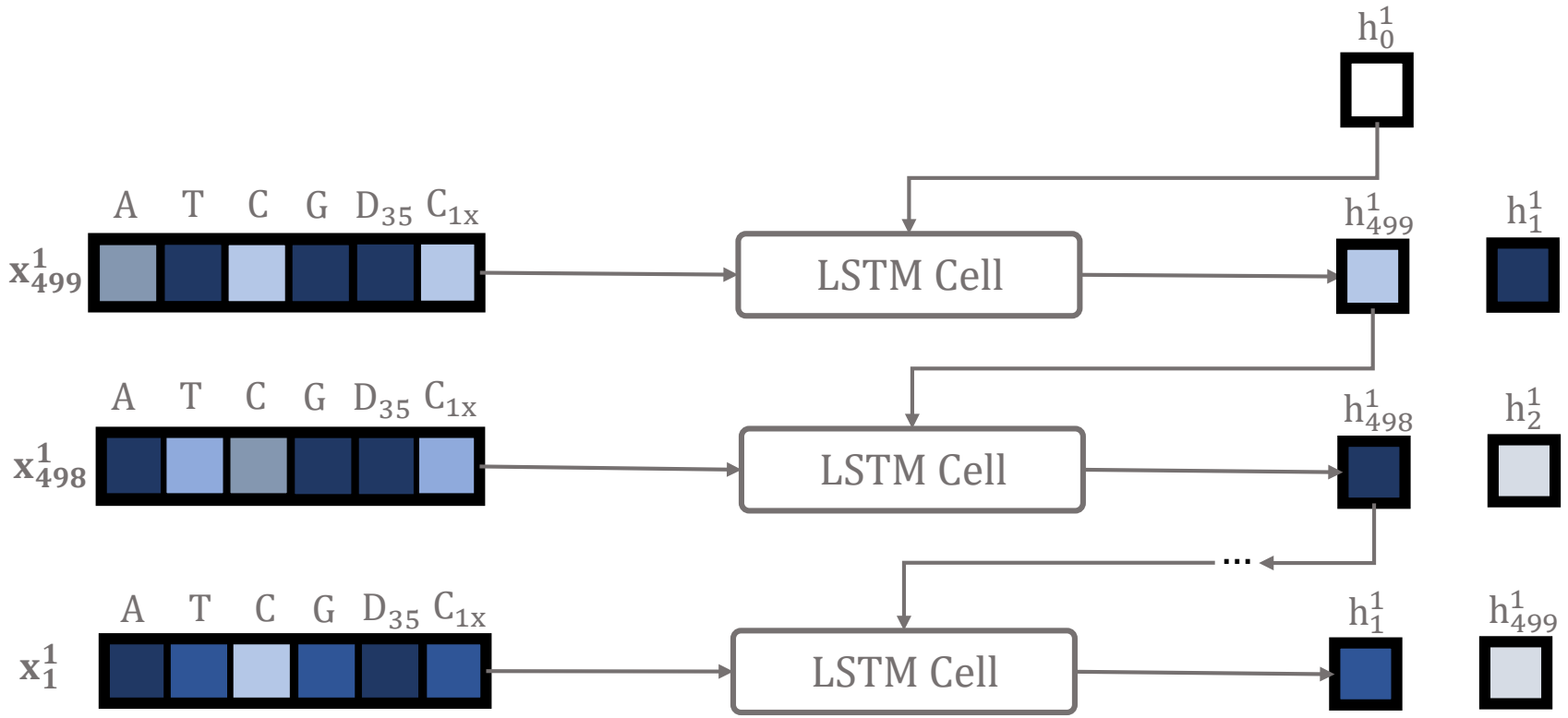
Long short-term memory cell



Long short-term memory cell

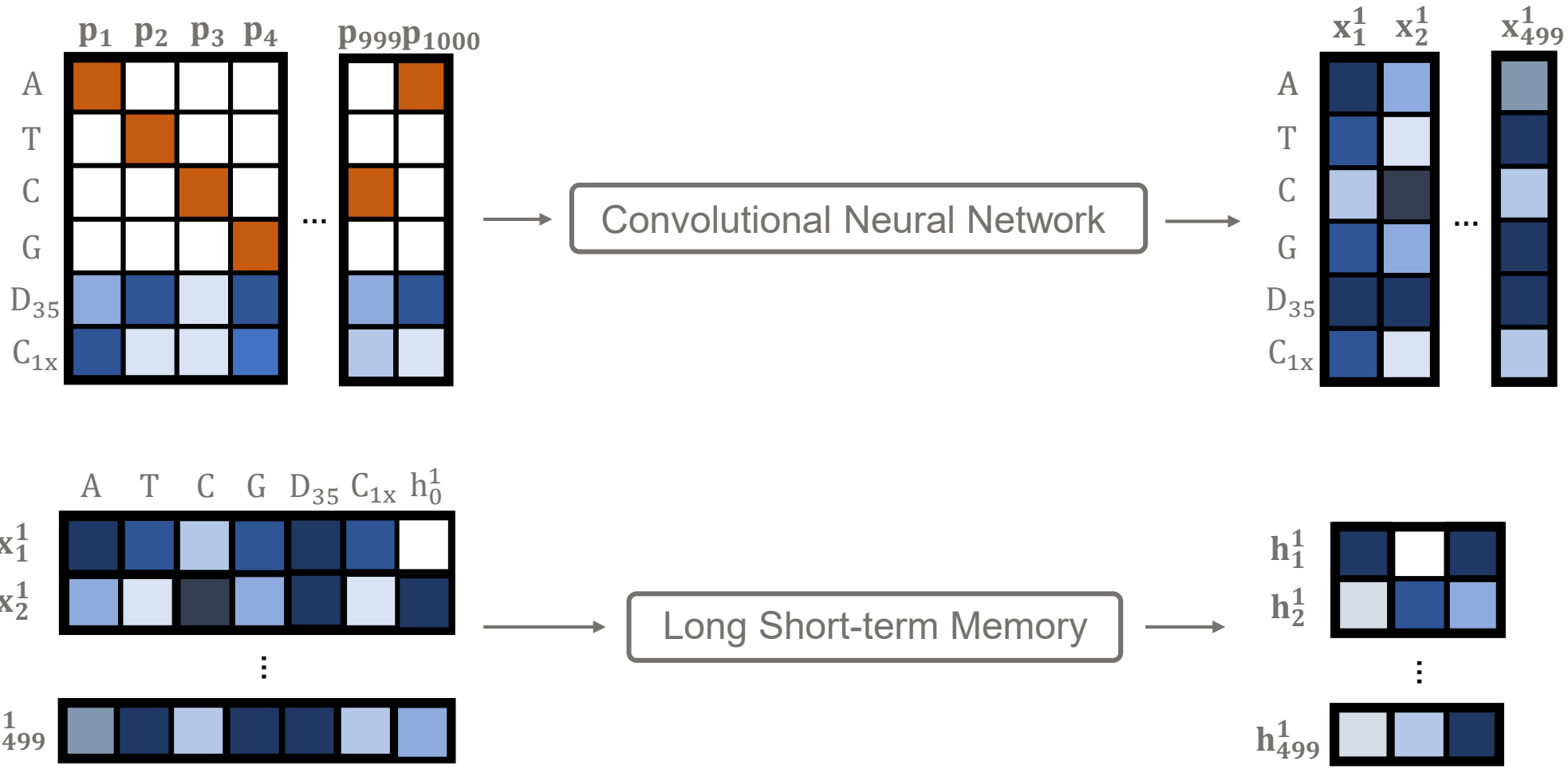


Bi-directional LSTM



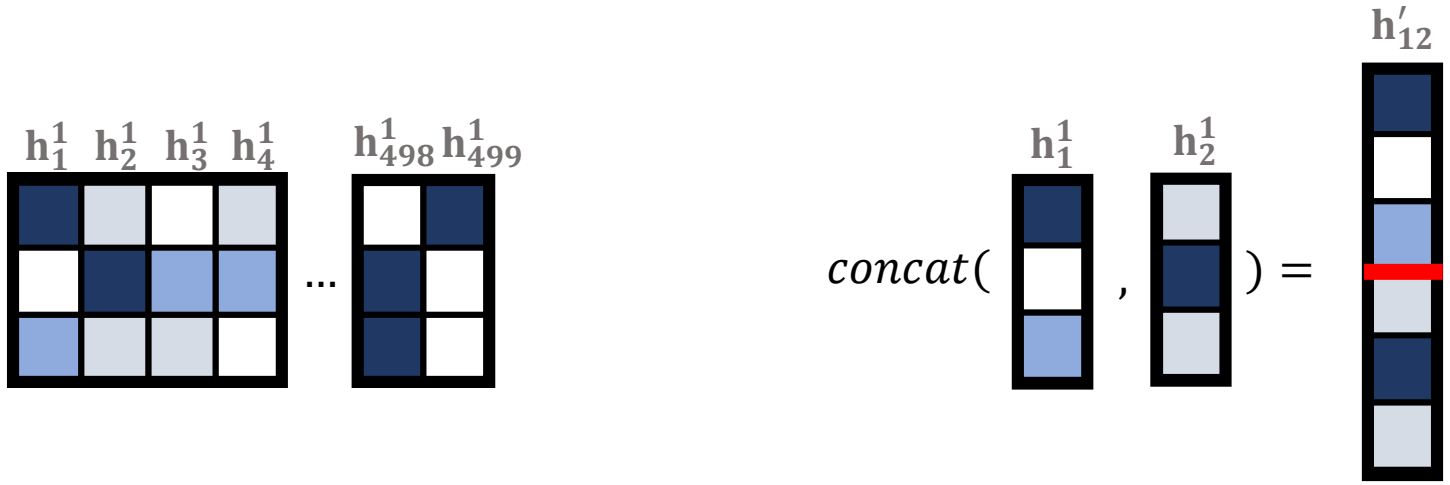
concatenate the two output

Input for Recurrent Neural Network



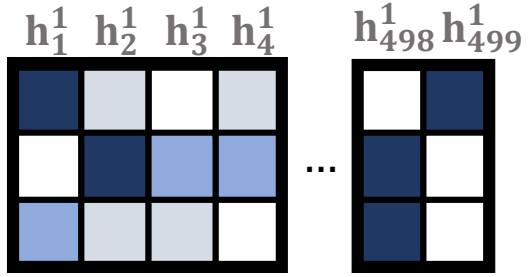
(Self) Attention mechanism

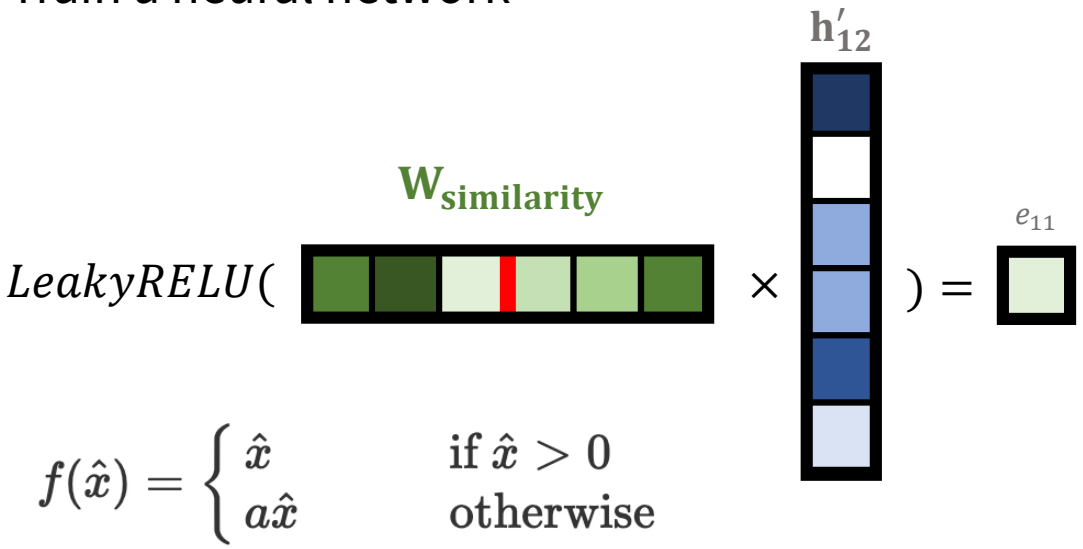
1. Concatenation



(Self) Attention mechanism

1. Concatenation
2. Train a neural network

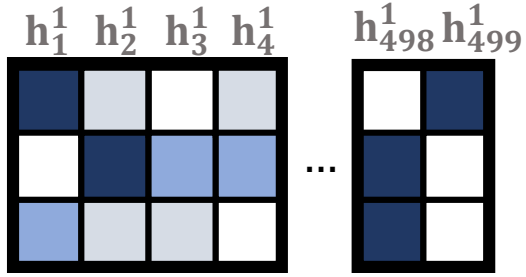


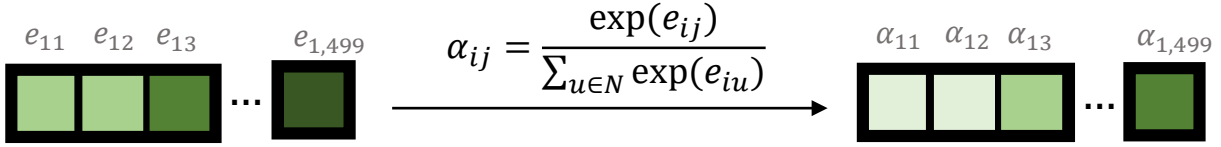


$$f(\hat{x}) = \begin{cases} \hat{x} & \text{if } \hat{x} > 0 \\ a\hat{x} & \text{otherwise} \end{cases}$$

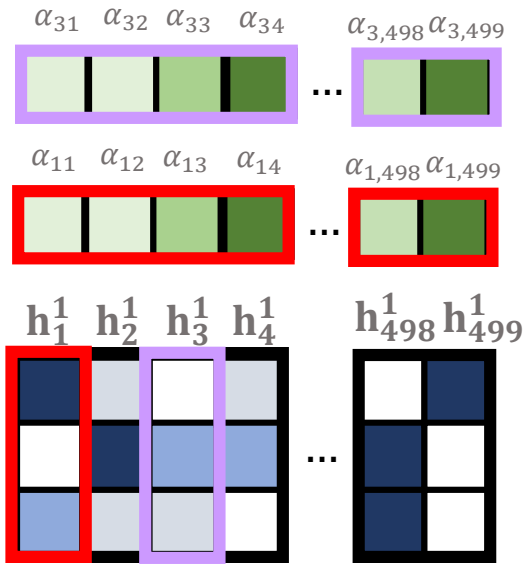
(Self) Attention mechanism

1. Concatenation
2. Train a neural network
2. Apply softmax

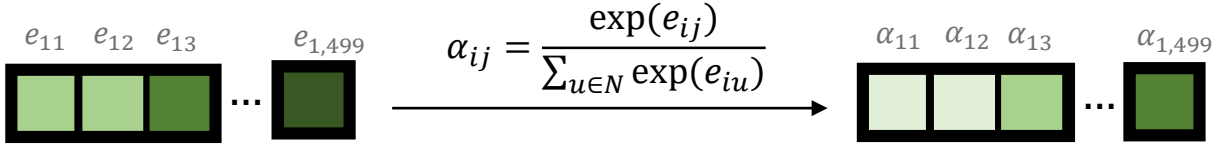




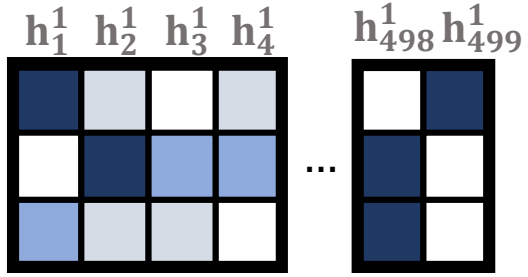
(Self) Attention mechanism



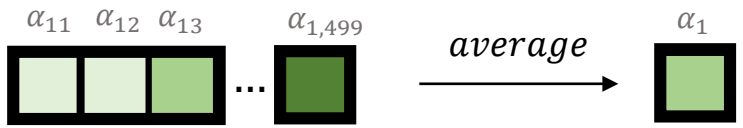
1. Concatenation
2. Train a neural network
3. Apply softmax



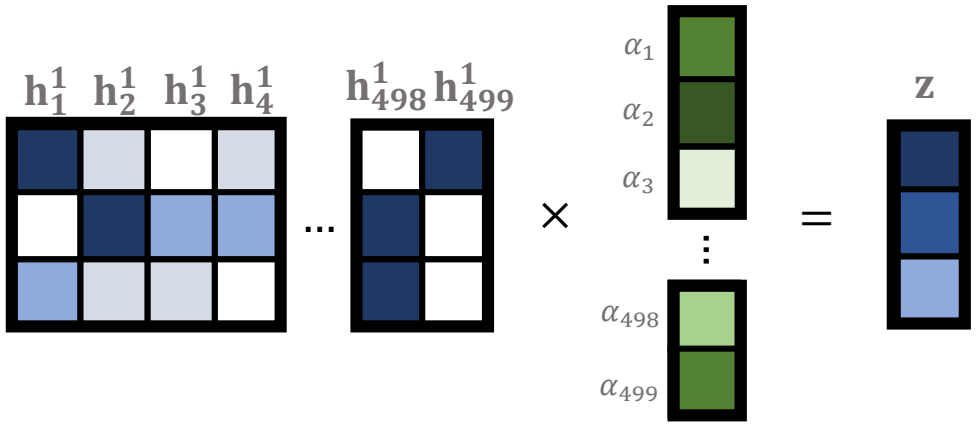
(Self) Attention mechanism



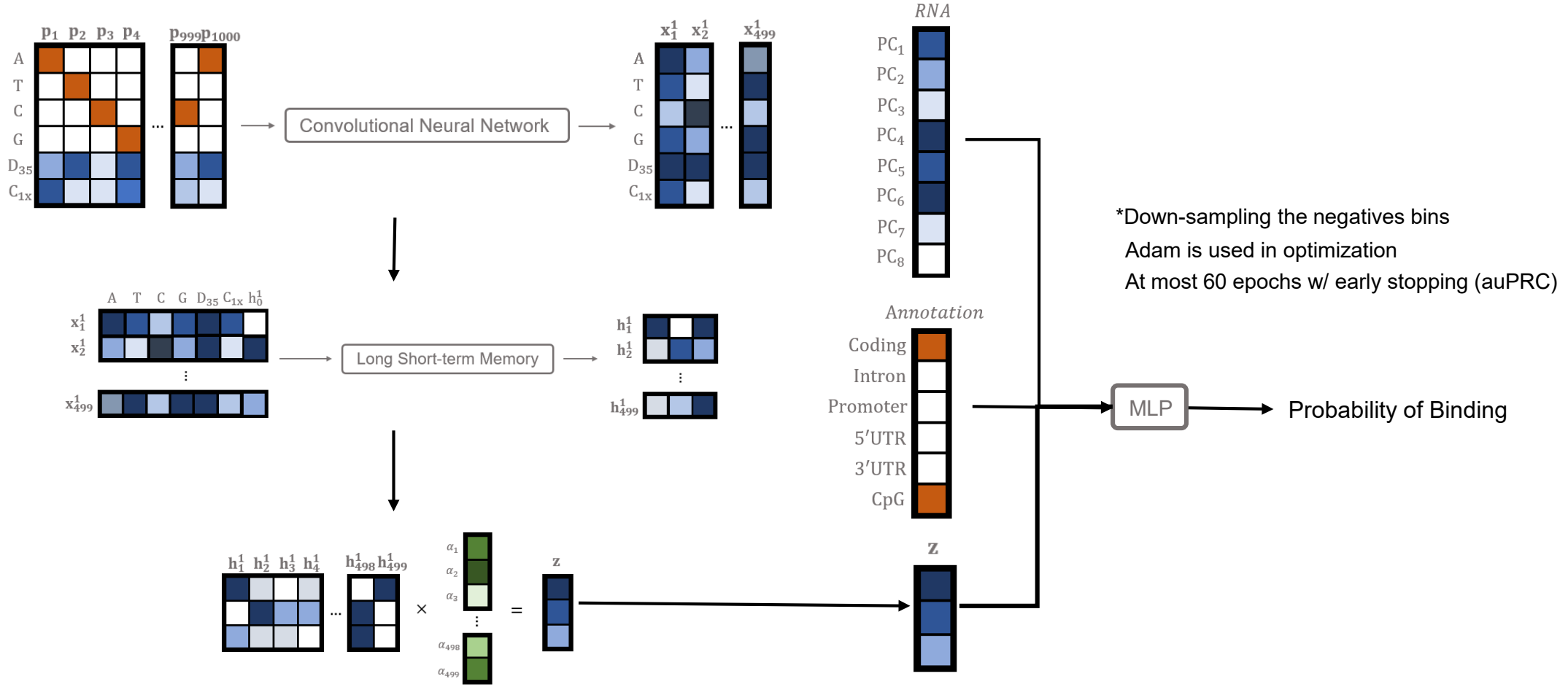
1. Concatenation
2. Train a neural network
3. Apply softmax
4. Aggregate



(Self) Attention mechanism



Overall network structure



Results

Benchmarking on Evaluation Data

- DeepGRN attention model better performed on:
  - 69.23% of prediction targets than **Anchor**
  - 69.23% of prediction targets than **FactorNet**
  - 72.92% of prediction targets than **Catchitt**
  - 92.31% of prediction targets than **Cheburashka**
- Highest auPRC on six targets

Table 1. The results of our attention model and top four algorithms in the DREAM Challenge (Bold denotes the highest auPRC among all methods).

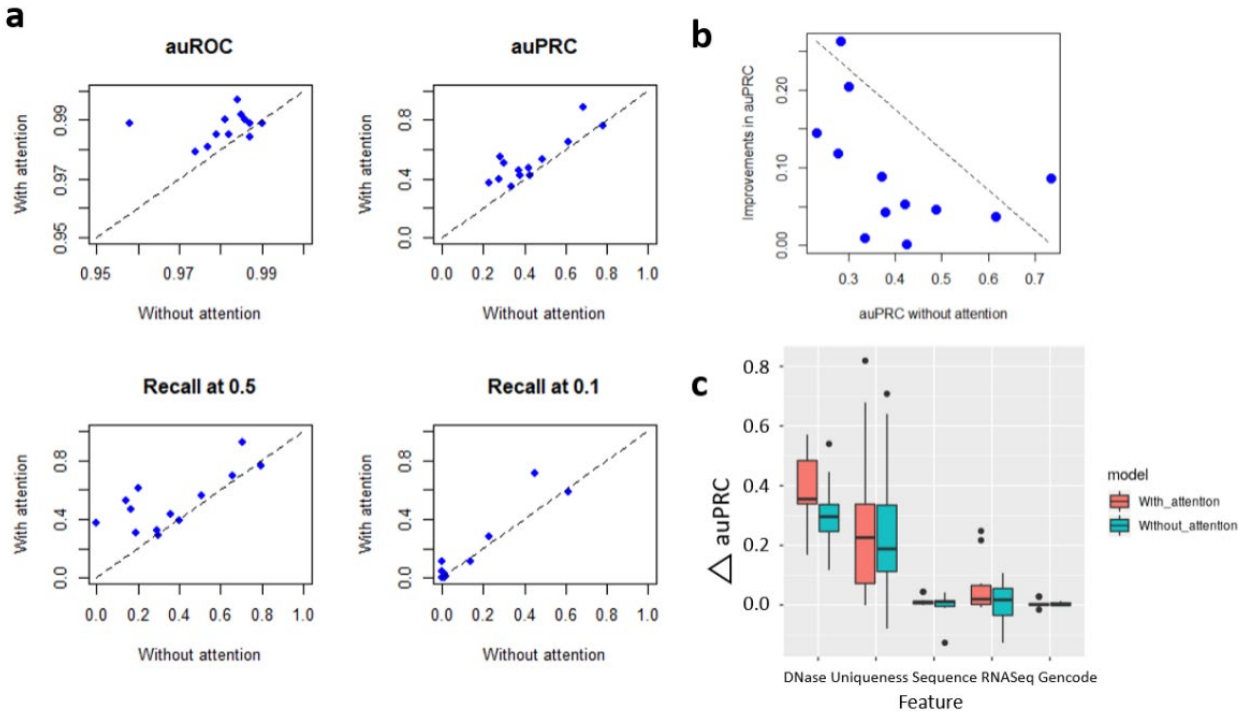
TF Name	Cell Type	Anchor	FactorNet	Cheburashka	Catchitt	DeepGRN
CTCF	induced pluripotent stem cell	0.755	0.861	0.780	0.816	**0.883**
CTCF	PC-3	0.608	**0.783**	0.480	0.753	0.756
E2F1	K562	0.352	0.241	0.353	**0.427**	0.375
EGR1	liver	**0.429**	0.317	0.364	0.399	0.395
FOXA1	liver	0.376	0.492	0.310	0.488	**0.534**
FOXA2	liver	0.461	0.217	0.315	0.387	**0.505**
GABPA	liver	0.470	0.442	0.444	0.423	**0.473**
HNF4A	liver	0.586	0.619	0.519	0.596	**0.652**
JUND	liver	**0.599**	0.265	0.447	0.422	0.546
MAX	liver	**0.535**	0.422	0.474	0.448	0.422
NANOG	induced pluripotent stem cell	0.309	**0.354**	0.195	0.315	0.344
REST	liver	0.398	0.412	0.324	0.264	**0.46**
TAF1	liver	**0.437**	0.428	0.375	0.413	0.426

Results

Performance improvement of attention mechanism with a CNN-RNN model

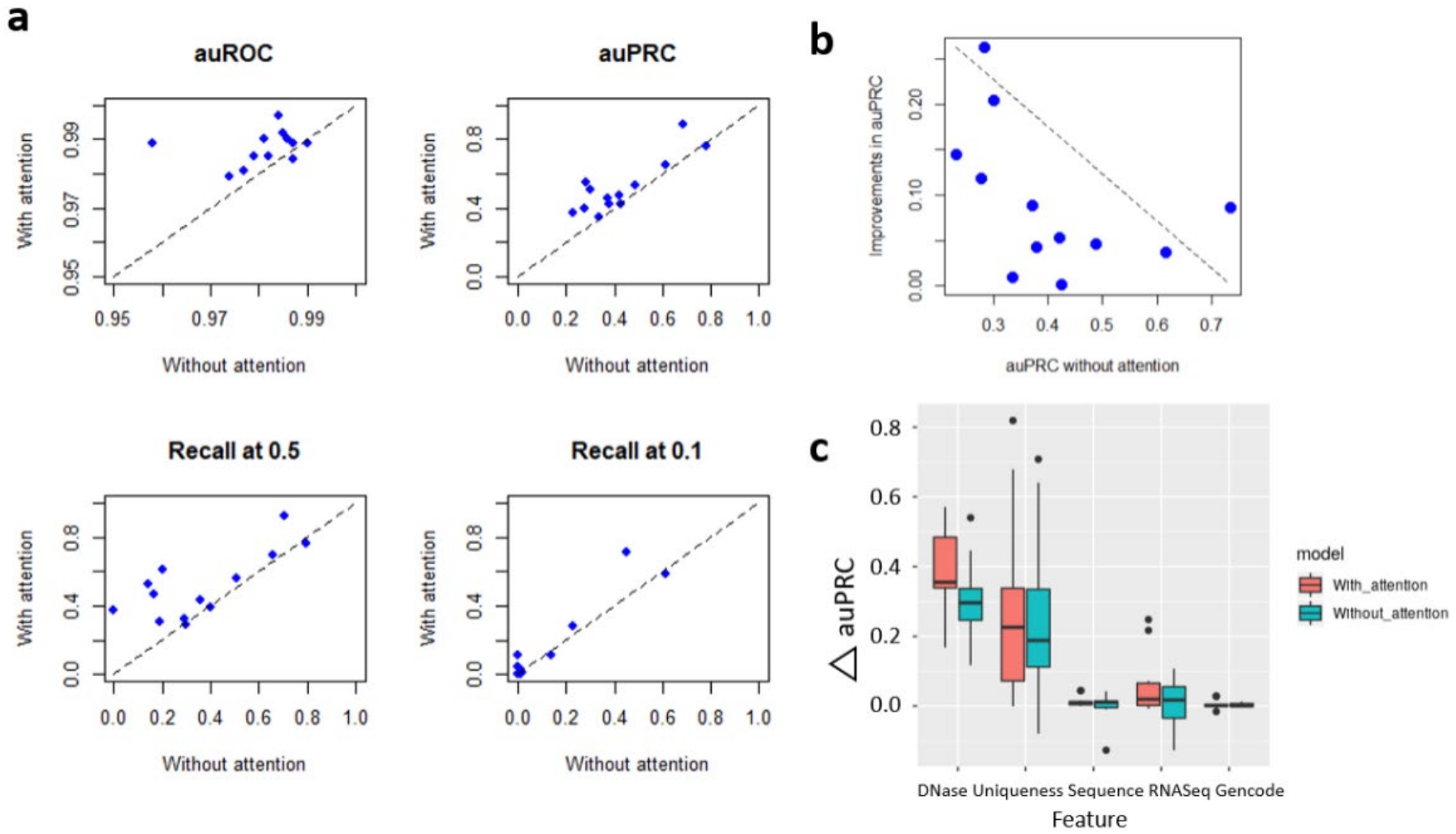
- DeepGRN attention model achieved better or equal performance on:
  - 7 targets with all scoring metrics
  - 10 targets with 3 metrics
    - Except for recall at FDR 0.1

- Largest auPRC improvement
  - JUND (0.262), FOXA2 (0.204), E2F1 (0.144), and EGR1 (0.118)

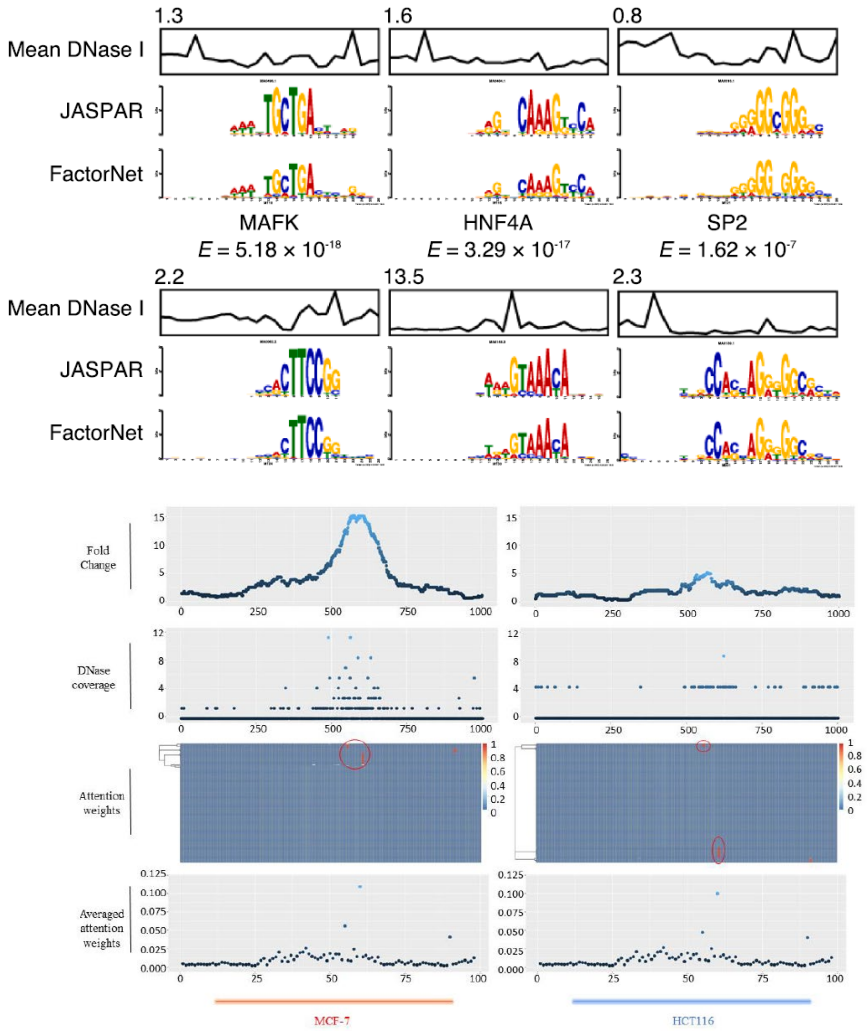


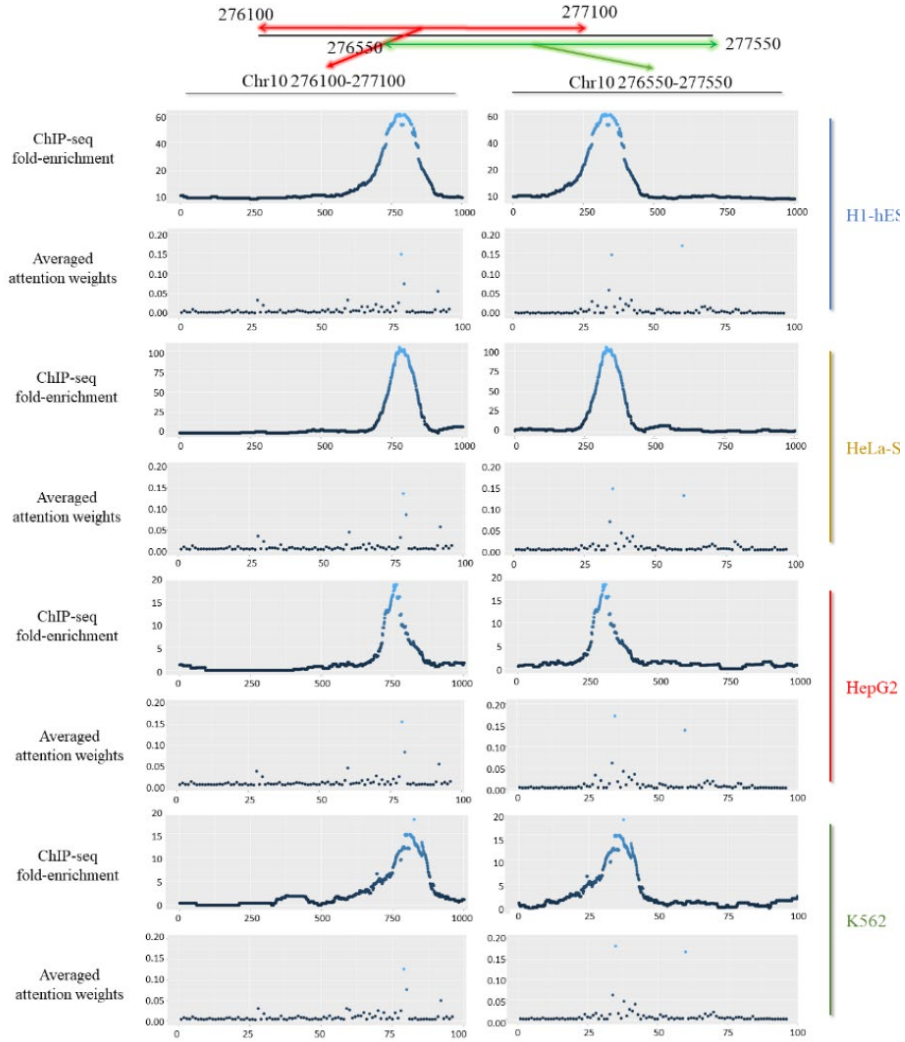
Results

Performance improvement of attention mechanism



Interpretation of the Model





Discussions

- The manuscript is short of visual representation specially for simplified explanation of the structure of the model
- They refer to the FactorNet model to describe their input structures
  - It is less convenient for the reader to refer to another paper and try to make the connection with this paper
  - The manuscript would benefit from a comprehensive visual representation (diagram) of structure of input features (as in FactorNet) & their of DNN architecture
- Visual report of the results of their benchmarking instead of table(s)
- Figure 2b caption refers to Pearson and Spearman correlation coefficients but these are missing from the figure
- The discussion section is very thin.
  - They are rephrasing the results part and put it in discussion
- Annotation feature ‘CpG islands’, may not be optimal
  - ~70% of human promoters contain CpG islands

Thanks