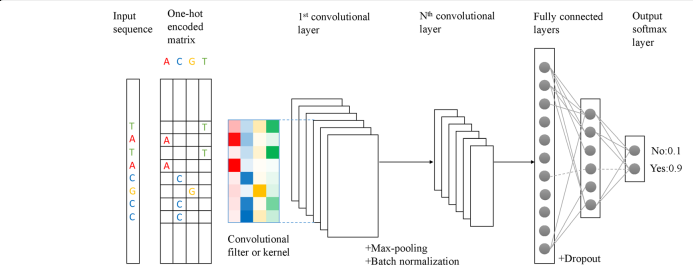
Report deep learning motif discovery

# Introduction

## A deep learning approach to motif finding

As shown in Fig1 a deep neural network takes the raw DNA sequences as input. No manually crafted dataset of features is needed, which is one of the major advantages of using a deep learning approach. After transformation to a 1-hot representation, the DNA sequences can be inputted into a first convolutional layer. Here a number of kernels, acting as motif scanners, will go over the sequences. Their width equals 4 (number of nucleotides) and their length corresponds to the length of the motifs we try to extract. Likewise, the number of kernels will correspond to the maximum number of motifs we will be able to find. This first convolutional layer is then followed by consecutive convolutional layers, each time transforming the input data into more higher level features. At certain points drop out layers and batch normalization are used to reduce the number of parameters, and as a counterweight to overfitting. At the end, these extracted featured are combined through some dense layers and eventually passed through a softmax layer which outputs a probability distribution over the different classes (in our case binary , P or I). During training, the network will adjust its weights so that classification is improved. This will lead to an adjustment of the weights in the kernels of the first layer to match the characteristic differences between the two types of DNA sequences (=the motifs).

In addition, some form of recurrent layers can be incorporated as well. A popular choice are Long Short Term Memory networks (LSTMs). These recurrent neural layers can pick up on sequential information and are therefore better suited to capture the context of the motifs (motif syntax).

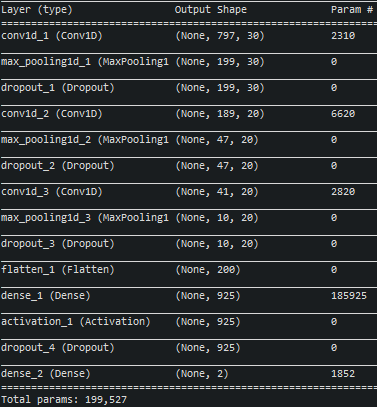
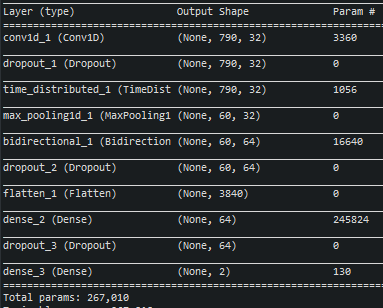


*Figure 1: Overview of a deep learning set up for motif detection. From Min, Xu et al (2017)*

# Results

## 2 model architectures

2 model architectures are compared (see fig 2). Model 1 uses 3 CNN’s, Model 2 uses only 1 convolutional layer, and incorporates a recurrent layer (LSTM’s).



**Model 1**

**Model 2**

Figure 2: The 2 model architectures.

(left) Model1 : using 3 convolutional layers. (right) Model2 : using 1 CNN and LSTM

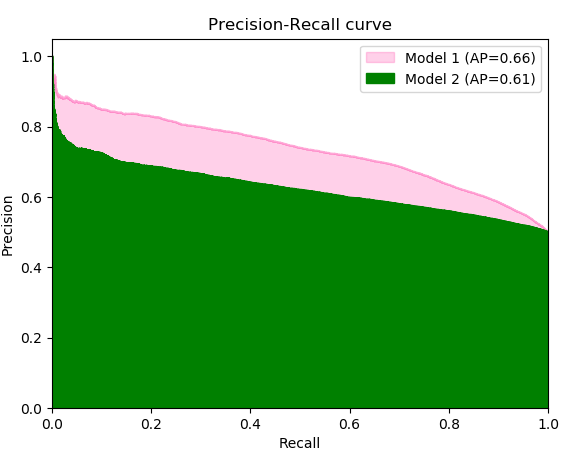
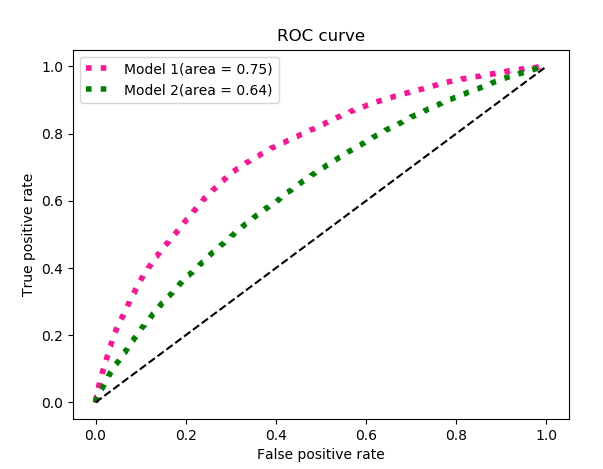
## Comparing classification performance

Both models are tested on a balanced test set of 45333 sequences (20% of the total). From Table1 and Figure 3 it is clear that model 1 outperforms model 2. The model using multiple convolutional layers achieves an accuracy that is 9% higher then the model using the recurrent layer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | accuracy | AUROC | AUPR | Epoch time |
| Model 1 | 0.69 | 0.75 | 0.66 | 10min |
| Model 2 | 0.60 | 0.64 | 0.61 | 15min |

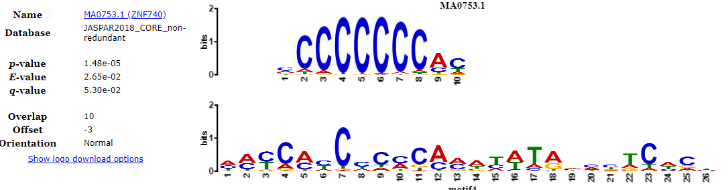
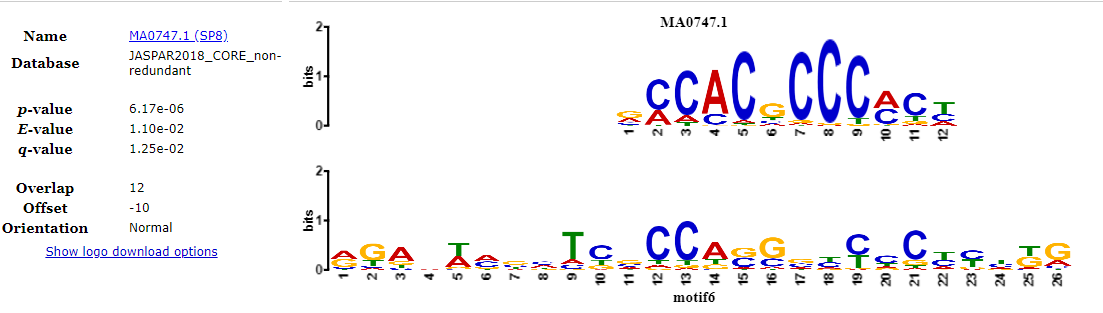
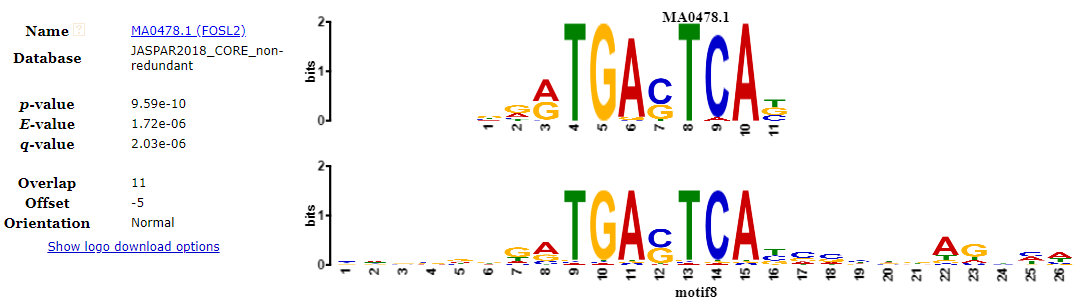
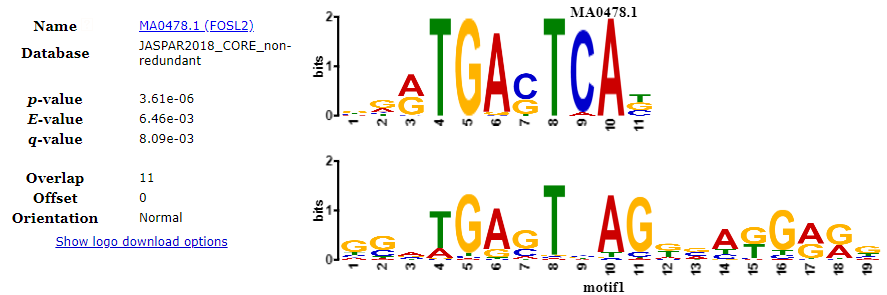
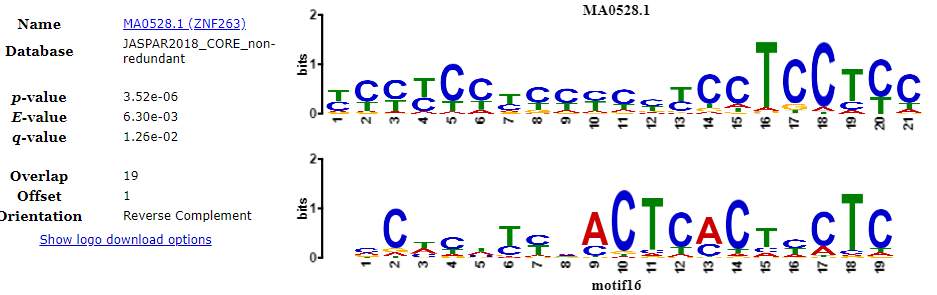
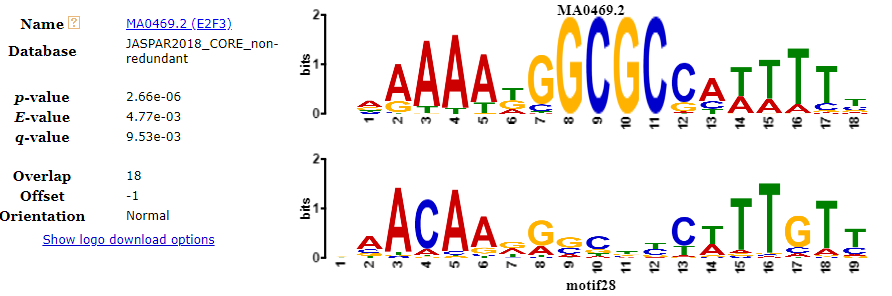
Table 1 Comparing classification performance of both models

Figure 3: ROC and Precision Recall curves comparing both models



## Comparing motif finding

After training, the kernels of the first convolutional layer, can be screened for motifs. To go from the kernel weights to a proper PSW matrix for a motif, each of the kernels are scored against each of the original DNA sequences. This score represents the best match of a motif when slided across a particular sequence. For each motif, the top 100 scoring sequences are stored, and these 100 sequences are then summarized into 1 PSWM. These PSWM’s are then queried against the JASPAR motif database using TOMTOM [2]. This way we can rank the motifs according to significance. The top 3 motifs for each model in shown in Fig4.



**Model 2**

**Model 1**

Figure 4:Top three most significant motif matches using TOMTOM for both models. The motif on top is the motif from the JASPAR database, the motif below it, the motif derived from the model. Mathces to very similar motifs are not taken up in this top three to avoidredundancy

# Methods

## Data augmentation

Even though the network can work with raw DNA sequences as input, there are still two issues that need to be overcome. First, the input to the convolutional layer needs to be of fixed length, and second, a deep neural network needs a vast amount of training data. To overcome these 2 issues, a moving window approach as in [1] (with stride = 20) is used ( fig 5).

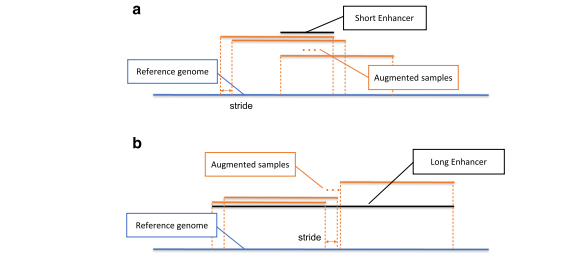


Figure 5:. Diagram of data augmentation. By using a sliding window approach, each enhancer sequence produces multiple sequences of equal length that can be used as input for the first convolutional layer. Taken from Min, Xu et al (2017)

# Discussion

…

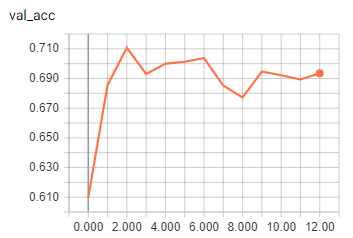
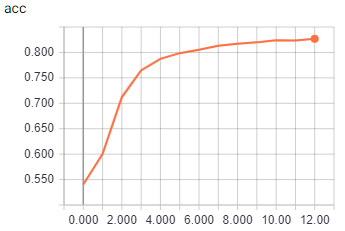
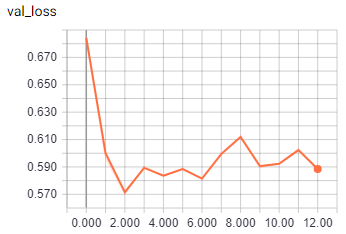
# References

1. Min, Xu et al (2017). Predicting enhancers with deep convolutional neural networks, *BMC Bioinformatics*, *18*(Suppl 13):478 – 490. DOI:10.1186/s12859-017-1878-3
2. Shobhit Gupta, JA Stamatoyannopolous, Timothy Bailey and William Stafford Noble, "Quantifying similarity between motifs", *Genome Biology*, **8**(2):R24, 2007
3. Verfaillie, Annelien et al.(2015), Decoding the regulatory landscape of melanomareveals TEADS as regulators of the invasive cell state, *Nature Communications, 6:*6683. DOI: 10.1038/ncomms7683

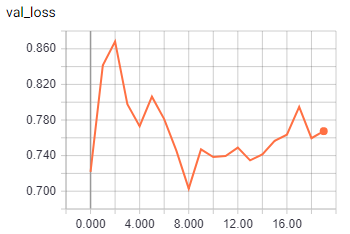
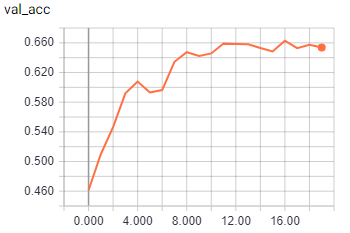
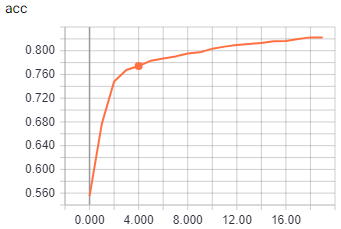
# Appendix

## training

Starting from 20122 enhancer sequences, we end up with 228051 sequences using the data augmentation approach (cfr Methods). The sequences are randomly split in a 70/20/10 fashion (train/test/val). The test and validation data are both balanced (equal amount of P and I labels).



**Model 1**

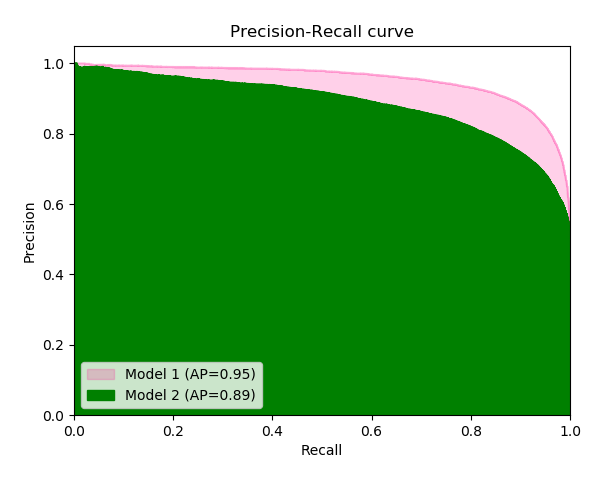
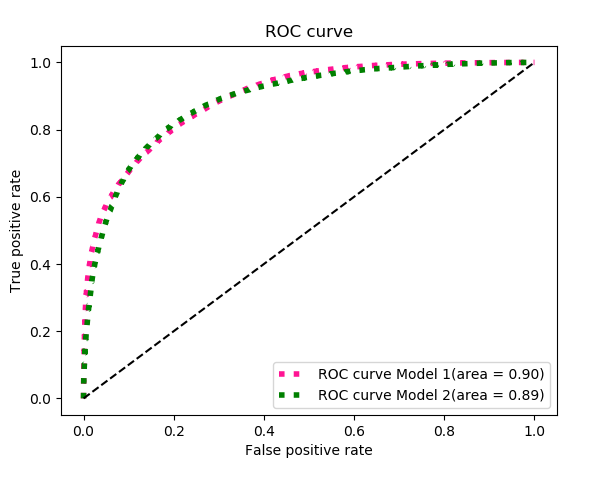


**Model 2**

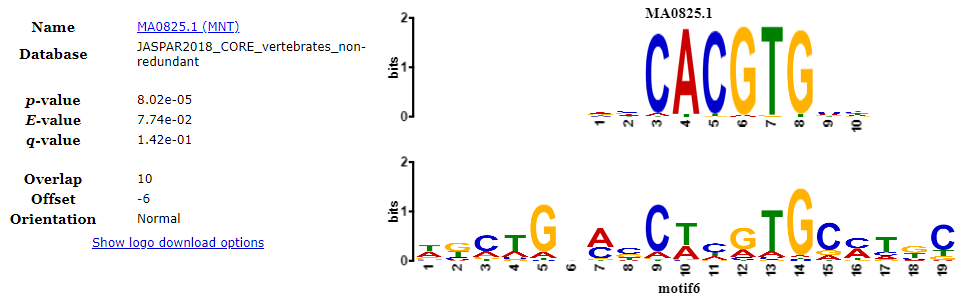
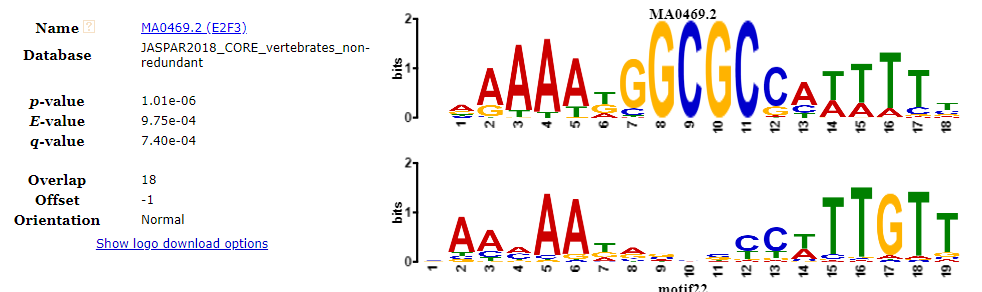
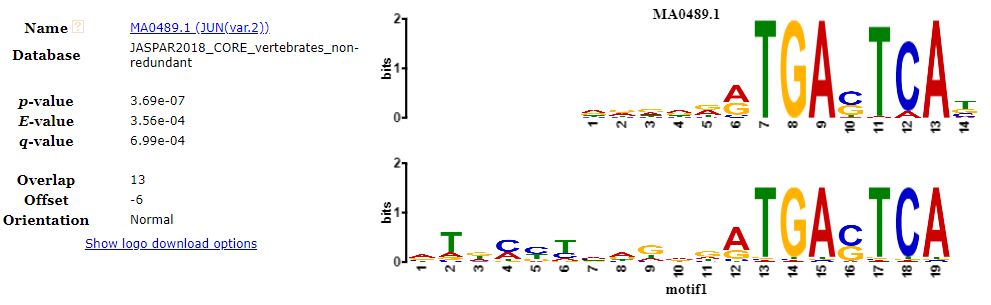
**Early stopping**

## Alternative approach

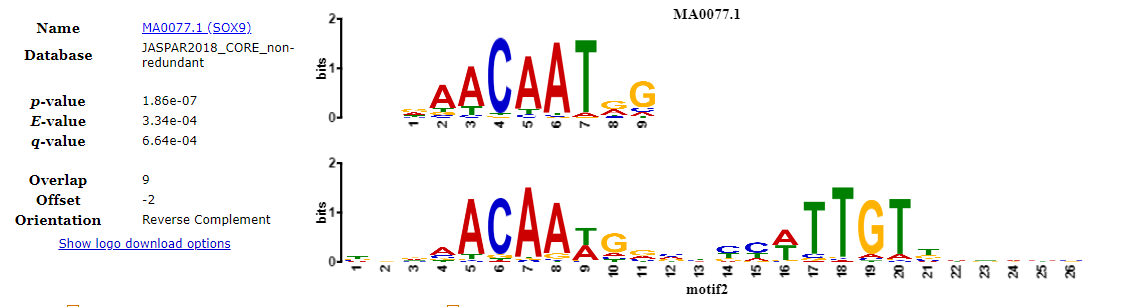
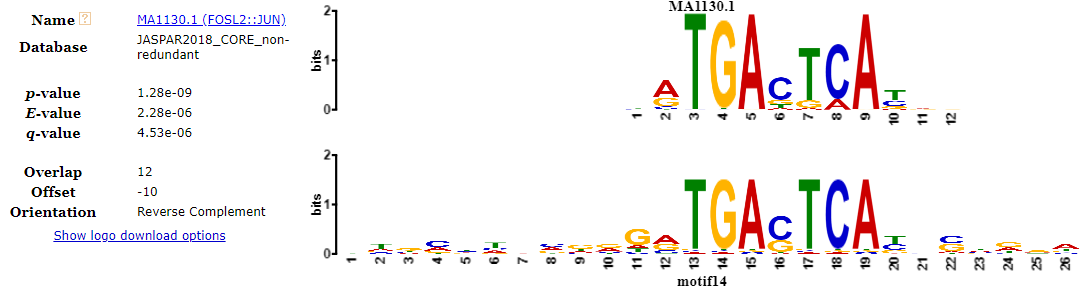
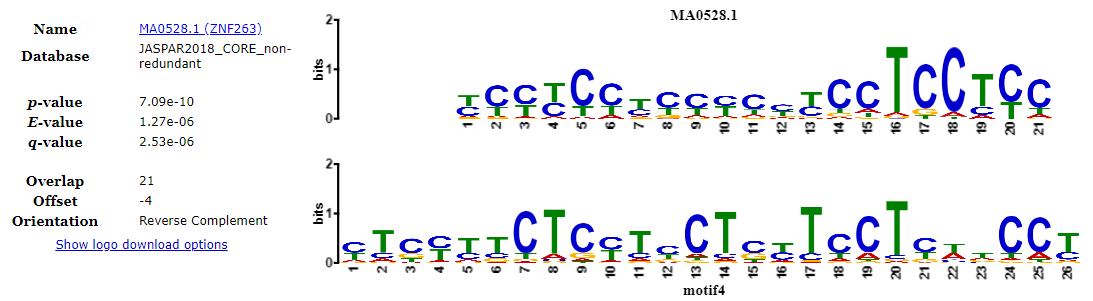
If the test and validation data are split after the data augmentation step (cfr Methods), there exists a correlation between test and training data. (the same sequence can occur in both test and train, although shifted in position). Even though the performance metrics are not reliable in this case (too optimistic), it is interesting to see that the motifs obtained from both models are actually a bit clearer then in the original set-up (using [3] as a reference).



**²x**



**Model 1**



**Model 2**

