Multiple Instance Learning for Aging Related Gene Identification

Ilze Amanda Auzina [*1] **Laurence Bont** [*2] **Jakub Tomczak** [*1]

Abstract

Many human diseases or health conditions have an underlying genetic component. In order to discover them the standard procedure is to proceed from gene level up, which requires detailed gene annotation as a starting point. However, gene annotation is a very difficult task, which is not readily available. Therefore, we propose to redefine the problem via multiple instance learning (MIL) setting, which allows to use a person level label, thus avoiding the need for detailed gene annotation. We show the first proof-of-concept implementation on a simulation data set consisting of only 160 people. The novel application succeeds to identify the relevant genes despite the small data set size and the high variance in gene lengths utilized.

1. Introduction

Genes are coding DNA sequences consisting of varying combinations of 4 nucleotides (ACTG). Within genes you can find motifs: nucleotide sequence patterns that are widespread and have, or is conjectured to have, a biological significance. Consequently, motif identification is a relevant task, as it suggests the biological role of a gene. Nonetheless, it is a difficult task as one motif often consists only of around 11 base pairs, where a 'mean' human coding gene sequence is of 67'000 bp long (Piovesan et al., 2016). Thus, even if a motif is present multiple times within a gene, its presence ratio is still very minor in comparison to the entire sequence length. Furthermore, multiple motifs can be present within the same sequence, hence, creating a difficult task for standard statistical approaches and the need for a large amount of data. On the contrary deep learning has shown to be very successful for fine-grained detail detection in the domain of image recognition (Krause et al., 2014), which presents a similar problem as motif identification: out of large amount of features the model has to able to identify few patterns relevant for the task at hand. However, where in imaging data there is not a strong sequence order effect, it does play an important role in genomics as it represents the information carried by a gene. Similarly to how word order plays role in human language, where deep learning approaches have been applied with success (Kim, 2014). Hence, all of this together strongly suggests that genomic data represents a very promising domain for deep learning applications.

Indeed, in recent years there have been multiple publications in the domain of deep learning applications for genetic data (Zitnik et al., 2019). However, most of these studies are based on very large amounts of annotated DNA sequences, where the focus has been on protein-binding classifications problems. Even though, whole-genome-sequencing has become more accessible, allowing to easily obtain the entire genome of a person, often in real-life settings such data is still limited to 100-200 patients. Therefore, in the present paper we aim to create a pipeline that is able to work with a limited set of patient cases. To achieve this an alternative form of standard convolutional neural network (CNN) is implemented, a Bayesian CNN, which is better suited for tasks with limited data (Gal & Ghahramani, 2015). Furthermore, gene level annotation is a very time consuming and high-expertise requiring task. Even more so, often the actual outcome we care about is a subject's level physiological characteristics, which can easily be measured and observed. Hence, using a subject level label would, firstly, be more easily obtainable, and, secondly, would provide insights of gene differences identified between healthy people and people with diseases or other physiological changes. In the present study the focus is especially on 'other' physiological changes that are not related to any underlying diseases, such as human aging.

Aging is a complex process, however, it has a clear genetic component. A study on twins has reported that heritability accounts for 26 percent in males and 23 percent in females for the expected lifespan (Herskind et al., 1996). However, the identification of genes related to aging so far has been limited to going from the gene level up: we known that a

[1]Department of Computer Science, Vrije Universiteit, Amsterdam, The Netherlands [2]Department of Computer Science, University of Amsterdam, Amsterdam, The Netherlands. Correspondence to: Ilze Amanda Auzina <ilze.amanda.auzina@gmail.com>, Lauence Bont <laurencebont@gmail.com>, Jakub Tomczak <jakubmkt@gmail.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

given gene plays a role in the DNA repair processes, hence, we assume that it might be involved in human aging, thus we test this assumption. This approach does allow to evaluate whether a given gene is involved in aging, however, it fails to identify new patterns. It rather represents a proof of concept than new discoveries. Therefore, what we propose is in order to achieve such insights is to go from the other way around. First, create a subject level label that indicates whether the person is 'fast-aging' or 'slow-aging' based on his/her physiological features. Second, for each subject gather a set of genes/dna sequences one might be interested to investigate. Thus, creating a dataset where for multiple instances (multiple genes), there is one real label given. Such a scenario is known in machine learning as *multiple instance learning* (MIL) (Dietterich et al., 1997). The benefits of this approach is that it does not require fine detailed annotations, yet with appropriate pooling method it is still able to identify the relevant instances, in this case genes.

Therefore, in this paper, we propose a novel MIL application for relevant gene identification. We design a pipeline based upon the work of (Ilse et al., 2018) that includes a CNN network which extracts relevant features from the data in combination with a MIL pooling, where the resulting output is passed into a standard classifier (neural network), allowing the entire model to be trained in an end-to-end manner (Appendix A fig 4). We suggest to use Monte-Carlo dropout for the convolutions in order to prevent the CNN on overfitting the training data, and to use gated-attention mechanism for MIL pooling in order for the model to be interpretable. Furthermore, we intend to use the entire sequence length in combination with padding. This allows to keep the gene order information intact, as well as to obtain the same sequence length across all genes. In the experiments we show that the designed approach is able to identify the relevant genes, which is the first known example in the present literature for MIL application on genetics data.

2. Methodology

2.1. Multiple Instance Learning (MIL)

MIL represents a special use-case of a traditional supervised learning problem where instead of having a single instance for a target variable, there is a bag of instances X = $\{ x_1, ..., x_k \}$ that are independent of each other. The underlying assumption is that individual labels exist for all the instances within a bag, however, these labels are unknown. The bag label distribution can be represented according to the Bernoulli distribution given some parameter: Θ (X)$\in$[0,1]. This bag probability must be permutation-invariant since no dependency is assumed between the instances in the bag. Following the implementation of (Ilse et al., 2018), bag probability can be formally defined as:

$$\Theta(X) = g(\sigma(f(x))) \tag{1}$$

Where (i) a transformation of instances is performed using a function *f*, followed by (ii) a combination of transformed instances using a symmetric (permutation-invariant) function σ, referred to as MIL pooling, and (iii) a transformation of combined instances using a function g. For the MIL pooling function an embedding-level approach with an attention mechanism is selected. The benefits of this approach are as follows: it is indifferent to the fact that individual labels are unknown, as well as the attention mechanism allows the approach to be interpretable.

2.1.1. Transformation function f

A convolutional neural network (CNN) is selected as the transformation function *f*. In order to avoid the CNN on overfitting the limited training data a Bayesian CNN is designed, also known as Monte Carlo dropout (Gal & Ghahramani, 2015). This means that a dropout is applied after all convolution and inner-product layers. The exact model setting are described in the section below.

Number of layers It is chosen to limit the CNN to only one layer. The decision is based upon the findings by Zeng et al. (Zeng et al., 2016). In their work they investigated multiple CNN architectures for a motif discovery task and discovered that adding additional layers did not improve the model's performance. This is contrary to other known implementations where a more deep CNN network was presented (Kelley et al., 2018),(Zhou & Troyanskaya, 2015). However, these studies had a large data corpus, while for the present use case the data is limited to 160 participants, therefore, implementing a shallow CNN might be advantageous to avoid over-fitting.

Number of kernels and kernel size The number of output kernels is set to 100 with a filter size of 11. These kernel settings are based on previous implementations, where it was found that high kernel count allows to capture more motif variants and filter size of 11 has repeatedly been reported as the filter size of choice in multiple studies for DNA pattern identification (Zeng et al., 2016), (Shrikumar et al., 2017), (Tampuu et al., 2019).

Pooling operation In order to extract the salient patterns k-max pooling operation is applied. This operation allows to extract the k most active features of a given input sequence (Kalchbrenner et al., 2014). The advantage of using max pooling over average pooling is that max pool operation is indifferent to the padding of the DNA sequences as it only takes the maximum value. This is of great importance for the designed approach, as the entire gene sequences are

used, meaning that for the shorter sequences as much as 96% of the sequence is consisting of padding.

2.1.2. Symmetric function σ

A gated attention-based MIL pooling is selected as the symmetric function σ (Ilse et al., 2018). The main benefit of this approach is its ability to assign different weights to instances within a bag, which allows to subsequently identify the key instances. The varying weights make the model interpretable for the user, allowing to draw conclusions about the instances investigated. Furthermore, the gated attention-based MIL pooling allows to capture more complex relations due to its gating mechanism, represented by the sigmoid non-linearity, $sigm(\cdot)$:

$$a_k = \frac{exp\left\{w^T(\tanh(Vh_k^T) \odot \text{sigm}(Uh_k^T))\right\}}{\sum_{j=1}^{K} exp\left\{\mathbf{w}^T(\tanh(Vh_j^T) \odot \text{sigm}(Uh_j^T))\right\}}, \quad (2)$$

where $\mathbf{U} \in \mathbb{R}^{L\times M}$, $\mathbf{V} \in \mathbb{R}^{L\times M}$, and $\mathbf{w} \in \mathbb{R}^{L\times 1}$ are parameters, $\odot$ is element-wise multiplication, and $tanh(\cdot)$ is hyperbolic tangent function to include both negative and positive values. In particular, the gating mechanisms removes the problematic linearity of the tangent function in the interval $x \in [-1, 1]$. Consequently, the final model is fully differentiable and adaptive, allowing the model to be trained with standard gradient descent optimization algorithm.

2.1.3. Transformation function g

Lastly, a simple feed-forward neural net is implemented. Where, the input is the created person-level embedding, and the output is the the predicted target label, creating a simple binary classification problem. Hence, Cross-Entropy loss is selected as the loss function. For a complete overview of the model settings see Appendix A table 4 and table 5 or visit the github page (https://github.com/LaurenceBont/Bayesian-CNNs-and-DNA).

2.2. Evaluation

In order to evaluate the performance of the model a 5-fold cross validation is selected as the evaluation method. In particular, for every fold 90% of the data is selected for training, while the remaining 10% is left for validation. The final result is computed by taking an average across the folds.

3. Related Work

Aging studies Most studies that have tried to identify patterns between the aging process and changes in DNA have focused on a specific set of genes. In particular, the focus has been on supervised classification models that try to discriminate between aging-related and non-aging related DNA repair genes, where rather than using the nucleotide sequence, multiple features are constructed to represent some meaningful DNA properties (Freitas et al., 2011). These features are often created manually, by annotating the DNA sequence, and then they are passed trough a standard machine learning algorithm such as Support Vector Machines (SVM), Random Forest (RF) or Naive Bayes (NB). Some of these models allow to draw inference of which one of the constructed features was most relevant (Breitbach et al., 2019), as well as to identify novel genes that might be important, given that the feature representation is created (Jiang & Ching, 2011). However, creating these features requires expert knowledge, their creation is time consuming, but even more some of the relevant information might be lost during this process. Of course, one of the reasons why feature extraction is done is because the utilization of the entire sequence increases the amount of data, while for such large amount of data the performance of standard machine learning models drops (Tampuu et al., 2019). However, if the data is analyzed with a deep machine learning model, an improvement in the performance can be observed with utilising the raw data sequence versus the extracted features (Tampuu et al., 2019). Therefore, in this paper the raw DNA sequence is used, which allows to retain all the information present in the sequence, as well as to draw inference about the possible gene motifs.

Deep machine learning model applications with genomic data In recent years there has been a growing trend for deep learning model applications to genomic sequence data. Most models have an underlying CNN architecture, but instead of the standard image processing pipeline (2D images with 3 color channels (R,G,B)), now we consider a 1D sequence with 4 channels ('A', 'C', 'G', 'T'). Most of these models have either outperformed existing models or provided new insights in the domain (Kelley et al., 2018), (Zhou & Troyanskaya, 2015), (Kelley et al., 2016). Consequently, here, we want to take the advantage of the observed capability of CNN to extract relevant patterns of the input data, but apply the model in a MIL setting.

MIL for medical applications The reason why MIL is so appealing for the medical domain is because it reduces the need for very detailed annotations, which are often hard to obtain to the extent that most machine learning models need. The main medical domain where MIL has been applied so far is medical imaging data. In particular, the application has been successful for computational histopathology (Ilse et al., 2018), mammography (nodule) classification (Zhu et al., 2017), and microscopy images (Kraus et al., 2016). This type of data provides a perfect setting for MIL: often there is one label per image (benign versus malignant), however, the image itself contains multiple cells contributing to the final diagnosis. Applying attention based MIL allows to

correctly classify the image as a whole, as well as to extract which cell was contributing to the final diagnosis. The second aspect is of great importance for medical data, as the expert has to be able to trace back the underlying cause of the diagnosis. Thus far there have been no examples in literature for applying MIL for genetic data, while it's analysis posses similar problems as medical imaging data: gene annotation is a very cumbersome process, however, we can relatively easily obtain a category/label on a person level. Therefore, it would be of great interest to see whether we can use a person-level labels to identify some gene level differences between individuals.

4. Experiments

4.1. Data

4.1.1. Simulation Data Generation

In order to validate whether the designed model performs as intended simulation DNA sequences are generated. In particular, 19 genes with varying sequence lengths are simulated for 160 people. For each person a label of 0 or 1 is assigned, creating two balanced groups each consisting of 80 people. For a complete overview of the exact gene lengths used see Appendix A table 6.

Gene sequence generation Firstly, for each gene a background sequence is generated by sampling a letter (ACGT) at each position with equal probabilities, 0.25 respectively. The generated sequences are used as the reference genome above which for each individual some individualistic differences are inserted. Secondly, the gene pool is subdivided in two categories: *relevant* and *irrelevant* genes. The *relevant* category represents genes in which DNA motifs are inserted for the people with label 1, while the *irrelevant* genes are modified across the participants with random mutations (table 1)

	alteration	
gene number	label 1	label 0
0,1,2	motif NRF1	random mutation
3,4,5	motif TFAP2A	random mutation
6,7,8	both motifs	random mutation
9-18	random mutation	

Table 1. The simulation dataset schematic representation

Irrelevant Genes: According to (Consortium et al., 2015) the observed genetic variation across humans is around 1%. Therefore, mutations are inserted in genes labeled as *irrelevant* based on a value estimated via *gene length* · 0.01. Hence, for longer genes the mutation rate is higher than for short. The mutation location is chosen at random, thus, avoiding the generation of any patterns.

Relevant Genes: Based on a meta-analysis about transcription factor motifs involved in human aging (Alfego et al., 2018), two motifs, NRF1 and TFAP2A, are selected as possible insertions in the relevant genes. In particular, the procedure is designed in similarity to the work of (Shrikumar et al., 2017). Firstly, for each motif its position probability matrix (PPM) is estimated. This matrix represents the probability of each letter at a given position of a motif. During each motif insertion the actual sequence pattern is sampled from this probability distribution. Secondly, the number of motif insertions in a gene are calculated. This number is sampled from a Poisson distribution, where the mean is estimated by formula (3) and the distribution is adjusted for longer sequences:

$$mean = \frac{s_{length} \cdot \alpha}{m_{length}} \tag{3}$$

where s_{length} is the length of the gene sequence, m_{length} is the length of the motif, and α is a hyperparameter set to 0.055, which is estimated by the ration used by (Shrikumar et al., 2017).

Subsequently, for people with label 1 the sampled motifs are inserted in the background sequence at random non-overlapping positions, while for people with label 0 random mutations are inserted according the same ratio as mentioned for *irrelevant* genes. Consequently, the resulting motif distribution for the relevant genes is as follows: 3 genes only containing motif NRF1, 3 genes containg motif TFAP2A, and 3 genes containing a mixture of both motifs.

4.1.2. Data Pre-processing

The input sequences are converted to a binary encoding, where each gene is represented as a 1D sequence with 4 channels (ACGT). As each gene sequence is of different lengths, all shorter sequences are padded with 0s to match in length of the longest sequence. Consequently, a matrix is created for each person representing a single bag, where in each bag there are 19 sequences.

4.2. Results and Discussion

4.2.1. Learning Rate

In total 6 learning rates were examined, ranging from 0.01 to 1e-07. Initially, only the standard learning rates were tested, 0.01 and 0.001 respectively. However, neither one of them returned any results, therefore, the learning rate was decreased. From the remaining ones learning rate of 1e-04 was chosen as the most optimal, as it had the best trade-off

k-max	mean validation loss (standard deviation)
1	**0.0165** (0.0051)
5	0.0356 (0.0068)
10	0.0258 (0.0199)
15	0.0615 (0.0179)

Table 2. Mean Validation Loss across 5 folds for each possible k-max setting

between learning speed and the loss obtained. Once the best learning rate had been selected, we shifted towards investigating hyper-parameters that could influence what kind of features are extracted from the model.

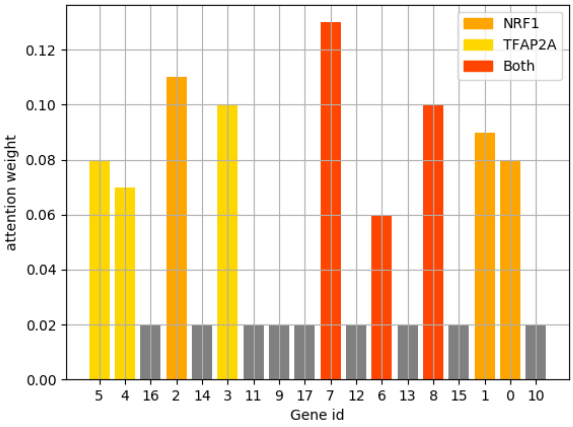
4.2.2. K-MAX VALUE

The kmax value in the model determines how many 'features' from the output of the CNN are actually passed further on, in order for the classifier to determine a correct label for an individual. It was chosen to investigate 4 possibilities, 1,5, 10 and 15. The value 1 was selected based on the successful implementation of (Shrikumar et al., 2017). However in their study the sequence length was limited to only 200 base pairs, while in the present case the shortest sequence is of length 2980. Thus, investigating higher k-max values was suspected to be beneficial as it would allow to capture more information.

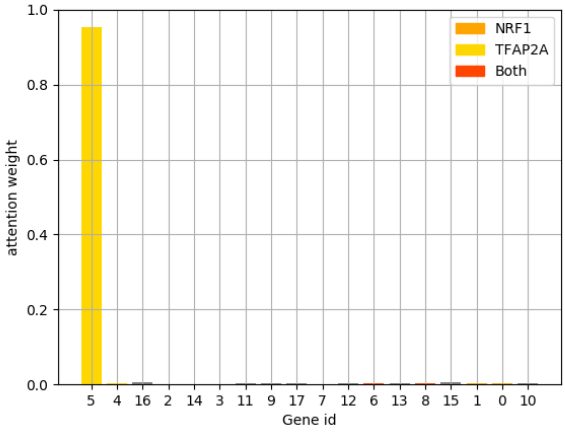
Training and Validation Loss The results revealed that k-max value of 1 obtained the lowest loss on the validation set, while k-max value of 15 obtained the highest value (table 2). However, this is in part could be due to the experimental setup: a lower k value results in fewer hidden units, hence, the models needs to estimate a smaller amount of weights. Because all models were trained for the exact same time, 180 epochs, the more simpler models could have obtained a lower loss, because there were fewer parameters to tune. This comes in line with also the observed variation in standard deviation across models: models with lower k value obtained a lower standard deviation. Furthermore, in Appendix B fig. 5 the obtained training loss can be seen overtime for each model across different folds. What becomes apparent by comparing the different plots is that because the training data is very limited (144 patients), when more features are extracted from the input (higher k value), the training loss becomes more varied across the folds. Thus, purely based on the obtained loss on the validation data, kmax value of 1 seems as the most optimal as it attains the lowest loss with the fastest convergence. However, as the final goal of the project is not accurate classification but rather correct gene identification, the obtained attention weight matrices were examined to draw more conclusive results.

Attention Weights The obtained mean validation loss suggested that a lower k-max value is more optimal, however, the resulting attention weight matrices suggest an alternative explanation. In particular, varying degree of differences can be observed across folds for all kmax settings, having said that, the extent of variation is k-max value specific. For a complete overview of results see Appendix C, in the section below we present only part of the results for descriptive purposes.

K-max set to 1: In simple terms one could say that the produced attention weights by kmax 1 were either superb or completely incorrect (fig 1). For fold 1 the best model according to the loss obtained on the validation data identified all the genes with an inserted motif, however, for fold2 the best model failed to draw any logical conclusions. This clearly shows a large discrepancy between the results obtained on different folds, which suggests that the obtained results are greatly influenced by the data loaded (Appendix C fig 6).



Successful attention weights fold 1



Unsuccessful attention weights fold 2

Figure 1. Weight matrix for kmax 1

K-max set to 5: Results of k-max 5 had the highest variance among the different folds. The obtained outcome would range from similar outcomes as for kmax1, as well as for kmax 10 (Appendix C fig 7), suggesting that kmax value of 5 is not optimal for the present data.

K-max set to 10: With k-max set to 10 the same genes were successfully identified across the different folds. Thus, having a higher consistency as compared to the previous two settings. However, out of the 9 genes with motif insertions only 6 of them were identified (fig 2). In particular, genes containing motif NRF1 were found, while genes containing only motif TFAP2A were not. The reason for this may be found by examining the expected base pair probability at each position of the two motifs (fig 3). For motif NRF1 the sequence contains on average more conserved base pairs, hence when sampling from the probability mass matrix, the resulting insertions would be more alike. While for motif TFAP2A there would be more variance across the different insertions, hence, its detection presents a more difficult task. Nonetheless, even though the same genes were identified, the weights computed did exhibit variation (Appendix C fig 8).

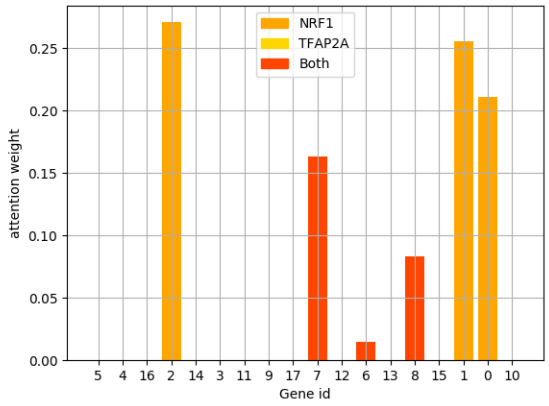


Figure 2. Weight matrix for kmax 10

K-max set to 15: The outcome of kmax set to 15 was similar to the one observed with kmax 10 (Appendix C 9): the same set of genes was consistently identified across the different folds. Even more so, the variation across the different folds with respect to the obtained weight values was lower, and, apart from fold1, for all remaining folds genes only containing motif NRF1 obtained a higher weight. This comes in line with the expectation that if the model's classification performance is based on the successful identification of motif NRF1 then for genes only containing this motif you would expect to see a higher attention weights, as these genes contain the highest amount of the motif. An exception of this rule sometimes can be observed for gene 7, but this is due to the fact that gene 7 is the longest gene, its length is 33 times longer than the shortest gene in the dataset. Therefore, it also has the most motif insertions as compared to other genes.

Sequence NRF1

Sequence TFAP2A

Figure 3. Base probability distributions for both motifs

Even though, all models exhibit some variation as discussed above, it was most prominent for models with a smaller k value. This might be due to the fact that if very few values are extracted then the model is very biased towards the exact data it sees, and since the dataset is very limited, for models with low k value you can see a massive discrepancy between the results. Support for this observation also comes from re-examining the produced training loss plots (Appendix A fig 5). Even though, kmax 1 has the fastest convergence, it also exhibits higher fluctuations than k-max 15. Furthermore, another possible explanation for the observations made comes from the fact that, even though, the random shuffle of the data was initialized with a given seed for every fold to keep the models comparable, the resulting validation sets did exhibit different ratios of positive and negative labels (table 3), perhaps leading to the very high variation observed for kmax 1 and 5. Hence, the cause might be two fold, having very few training data points makes the model more biased towards the data seen, while having variance in the validation set might indicate a different model as best, as the performance is reported for the model that performed best on the validation data. Nonetheless, the robustness of models with higher k-max value does suggest that using a higher k value might be beneficial when working with limited datasets as the results are more consistent across, despite the fact that not bot motifs were identified.

fold	ratio (1/0)
1	0.62/0.38
2	0.38/0.63
3	0.75/0.25
4	0.62/0.38
5	0.56/0.44

Table 3. Label ratio of validation set across different folds

5. Conclusion

In the current paper we present the first interpretable MIL approach for genomic data. As real-life data was not readily available for the current project, a simulation data set was generated which would mimic real genomic data. In particular, motifs related to aging were inserted to show whether the model would be able to detect patterns related to human aging process. We show a proof-of-concept implementation. The designed model was able to successfully identify genes with motif insertions, where only one or multiple motifs could be present within a gene. Furthermore, the designed model obtained a very low validation loss, despite utilizing a very small dataset. The success of the model given the data limitations, the small dataset size, whilst a very large variation of gene lengths, suggests that the current pipeline has the ability to identify relevant genes based on their underlying motifs, by simply using human level labels. Nonetheless, these results have to be interpreted with caution as the results showed that different value of kmax greatly influences not only the amount of genes detected, but also the consistency across different models. Therefore, future research should re-confirm the obtained results by two approaches: 1)by assuring that the validation data ratio does not vary across folds or 2) by increasing the number of folds performed by cross-validation. Alternatively, also increasing the amount of data might bring beneficial effects. However, the aim of the present paper was to simulate a real-life situation as much as possible, hence, the simulation data was not increased to a larger amount. Furthermore, the given implementation was limited to the investigation of only two motifs, in reality many more motifs might be present. Thus, working with very limited data might bring additional issues if the number of motifs increases significantly. Moreover, in the present paper the focus was on the investigation of different k-max values, however, the model has many more settings that can be tunned and optimized. Thus, future research should explore these settings as it might reveal more insights about different model settings when working with DNA data.

Acknowledgements

We would like to thank Meike Morren for the provided opportunity, as well as (eScience Center et al.) for allowing to use the Distributed ASCI Supercomputer 5 (DAS-5) for running the analysis of the model.

References

Alfego, D., Rodeck, U., and Kriete, A. Global mapping of transcription factor motifs in human aging. *PloS one*, 13 (1), 2018.

Breitbach, M. E., Greenspan, S., Resnick, N. M., Perera, S., Gurkar, A. U., Absher, D., and Levine, A. S. Exonic variants in aging-related genes are predictive of phenotypic aging status. *Frontiers in Genetics*, 10, 2019.

Consortium, . G. P. et al. A global reference for human genetic variation. *Nature*, 526(7571):68–74, 2015.

Dietterich, T. G., Lathrop, R. H., and Lozano-Pérez, T. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997.

eScience Center, R. v. N. N., ASTRON, J. R., and eScience Center, F. S. N. A medium-scale distributed system for computer science research: Infrastructure for the long term.

Freitas, A. A., Vasieva, O., and de Magalhães, J. P. A data mining approach for classifying dna repair genes into ageing-related or non-ageing-related. *BMC genomics*, 12 (1):27, 2011.

Gal, Y. and Ghahramani, Z. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015.

Herskind, A. M., McGue, M., Holm, N. V., Sörensen, T. I., Harvald, B., and Vaupel, J. W. The heritability of human longevity: a population-based study of 2872 danish twin pairs born 1870–1900. *Human genetics*, 97(3):319–323, 1996.

Ilse, M., Tomczak, J. M., and Welling, M. Attention-based deep multiple instance learning. *arXiv preprint arXiv:1802.04712*, 2018.

Jiang, H. and Ching, W.-K. Classifying dna repair genes by kernel-based support vector machines. *Bioinformation*, 7 (5):257, 2011.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

Kelley, D. R., Snoek, J., and Rinn, J. L. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome research*, 26(7): 990–999, 2016.

Kelley, D. R., Reshef, Y. A., Bileschi, M., Belanger, D., McLean, C. Y., and Snoek, J. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome research*, 28(5):739–750, 2018.

Kim, Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

Kraus, O. Z., Ba, J. L., and Frey, B. J. Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics*, 32(12):i52–i59, 2016.

Krause, J., Gebru, T., Deng, J., Li, L.-J., and Fei-Fei, L. Learning features and parts for fine-grained recognition. In *2014 22nd International Conference on Pattern Recognition*, pp. 26–33. IEEE, 2014.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Piovesan, A., Caracausi, M., Antonaros, F., Pelleri, M. C., and Vitale, L. Genebase 1.1: a tool to summarize data from ncbi gene datasets and its application to an update of human gene statistics. *Database*, 2016, 2016.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3145–3153. JMLR. org, 2017.

Tampuu, A., Bzhalava, Z., Dillner, J., and Vicente, R. Viraminer: Deep learning on raw dna sequences for identifying viral genomes in human samples. *PloS one*, 14(9), 2019.

Zeng, H., Edwards, M. D., Liu, G., and Gifford, D. K. Convolutional neural network architectures for predicting dna–protein binding. *Bioinformatics*, 32(12):i121–i127, 2016.

Zhou, J. and Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015.

Zhu, W., Lou, Q., Vang, Y. S., and Xie, X. Deep multi-instance networks with sparse label assignment for whole mammogram classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 603–611. Springer, 2017.

Zitnik, M., Nguyen, F., Wang, B., Leskovec, J., Goldenberg, A., and Hoffman, M. M. Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. *Information Fusion*, 50:71–91, 2019.

A. Model Settings

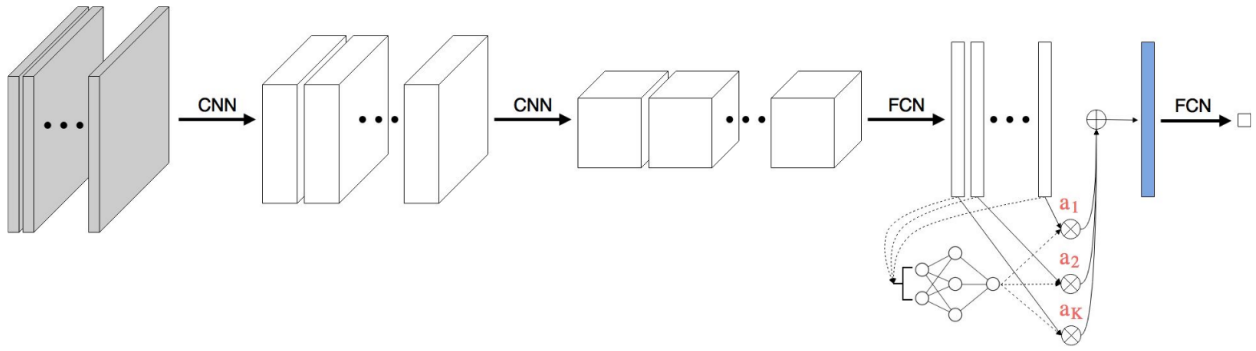


Figure 4. Model Schematic Design (Ilse et al., 2018)

Layer	Type
1	CONV1D(11,1,0)-100 + ReLU
2	Dropout(0.5)
3	k-maxpool(10)
4	fc-1000 + ReLU
5	Dropout(0.5)
6	MIL-gated attention-128
7	fc-1 + sigm

Table 4. Model Architecture

Experiment	Optimzer	β_1, β_2	Learning rate	weight decay	epochs
All	Adam	0.9, 0.999	0.0001	0	180

Table 5. Model optimization settings

gene0	gene1	gene2	gene3	gene4	gene5	gene6	gene7	gene8
25370	47402	84651	33734	30160	2980	18541	98820	32188
gene9	gene10	gene11	gene12	gene13	gene14	gene15	gene16	gene17
3353	5415	3736	3180	7038	21693	3065	3108	4176

Table 6. Gene sequence lengths

B. Training Loss Plots

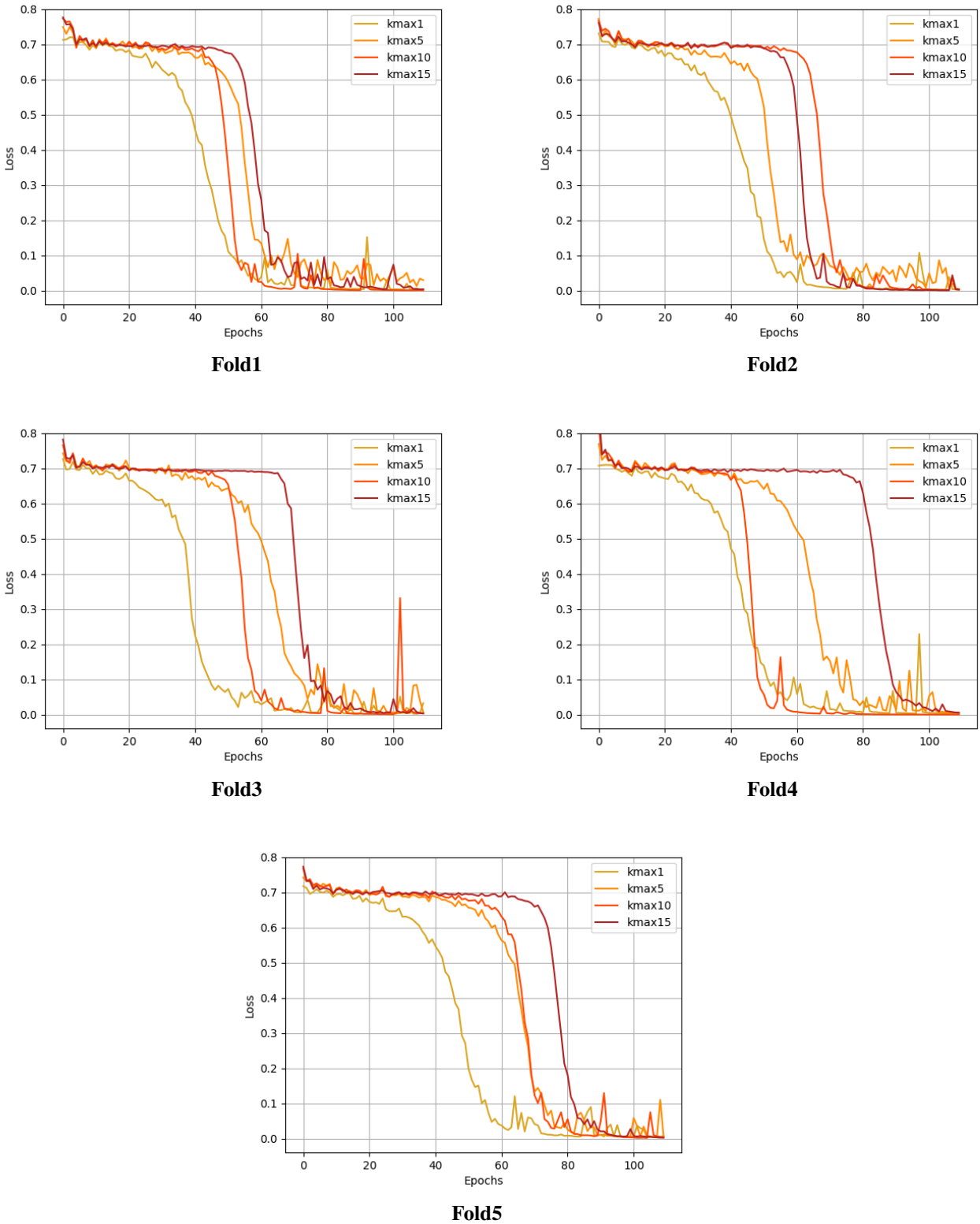
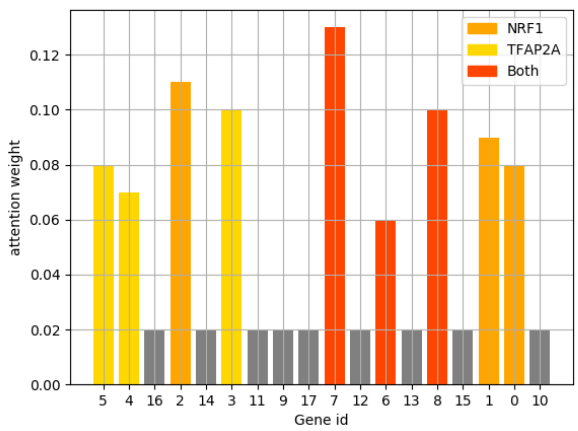
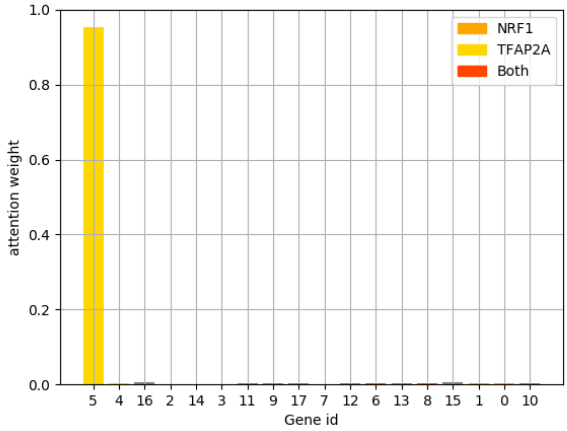


Figure 5. Training Loss for different model settings across different cross-validation folds

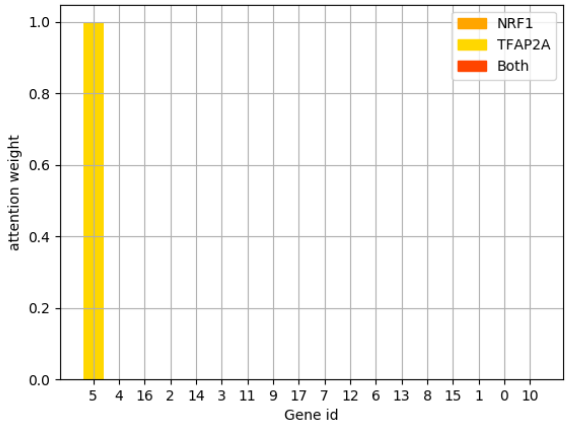
C. Attention Weights



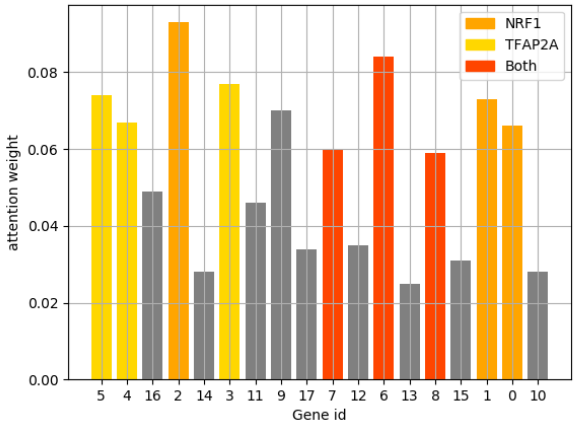
Fold1



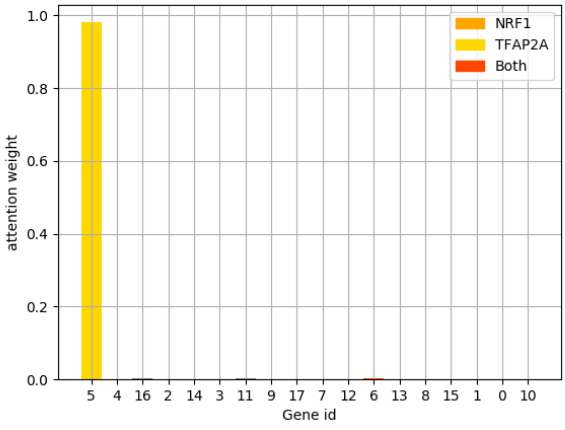
Fold2



Fold3

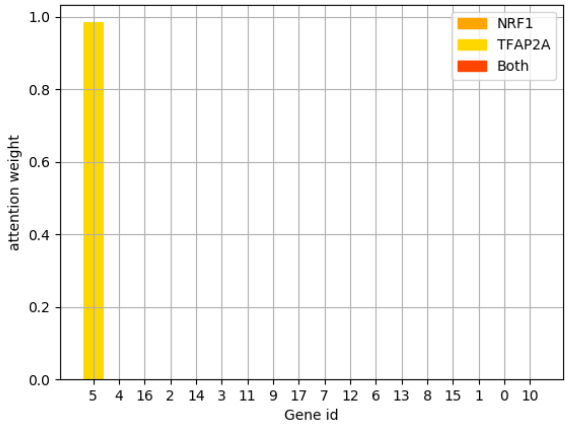


Fold4

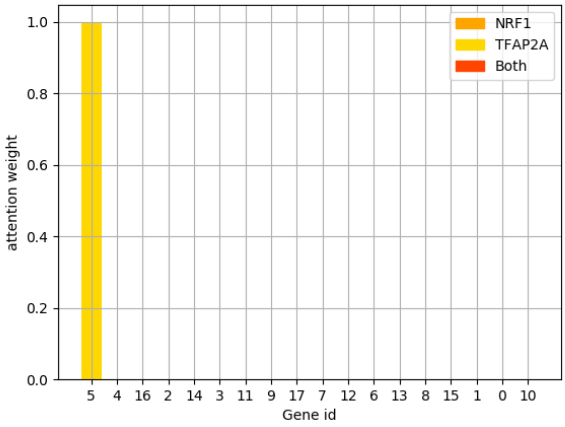


Fold5

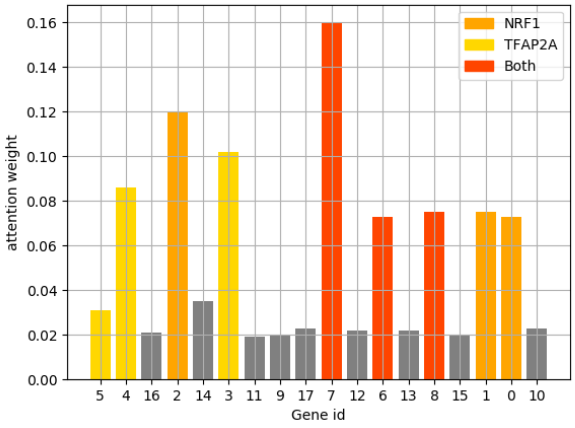
Figure 6. Attention weights for model with kmax-1



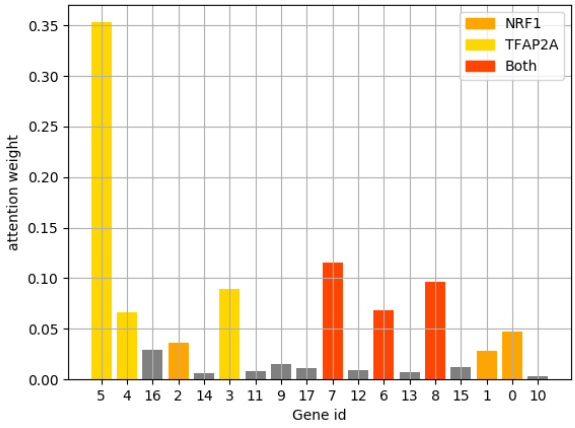
Fold1



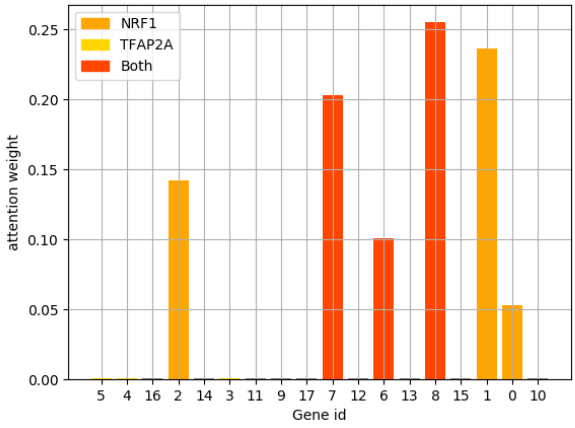
Fold2



Fold3

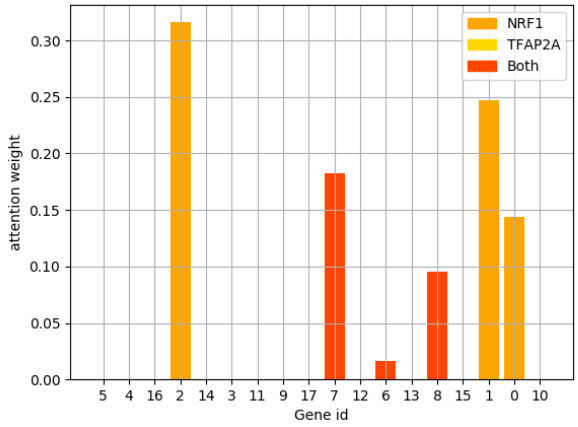


Fold4

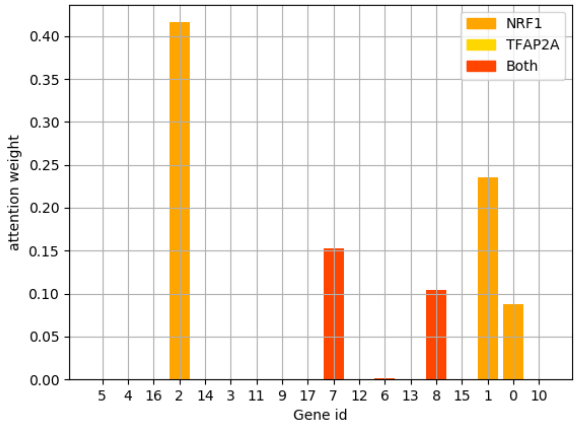


Fold5

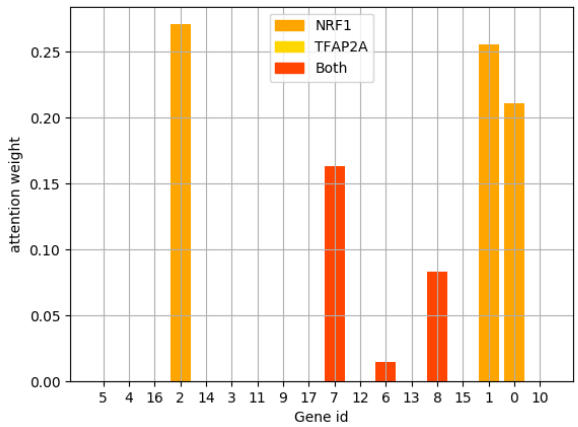
Figure 7. Attention weights for model with kmax-5



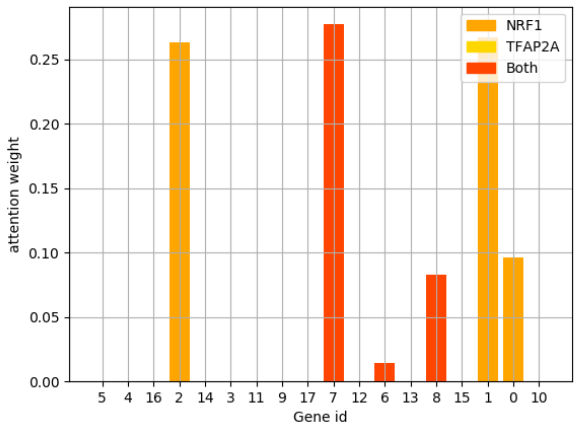
Fold1



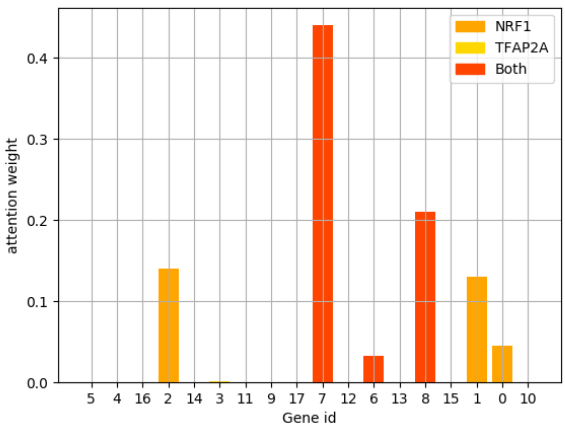
Fold2



Fold3

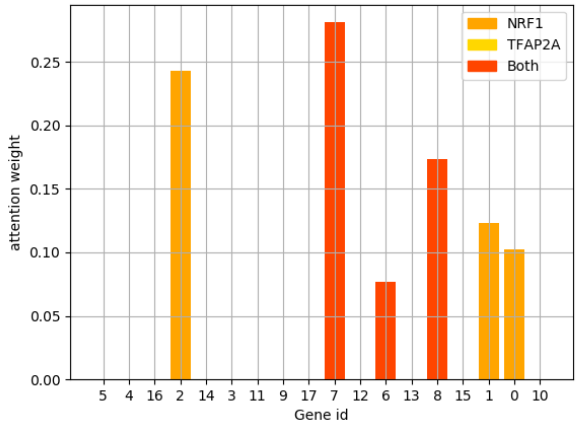


Fold4

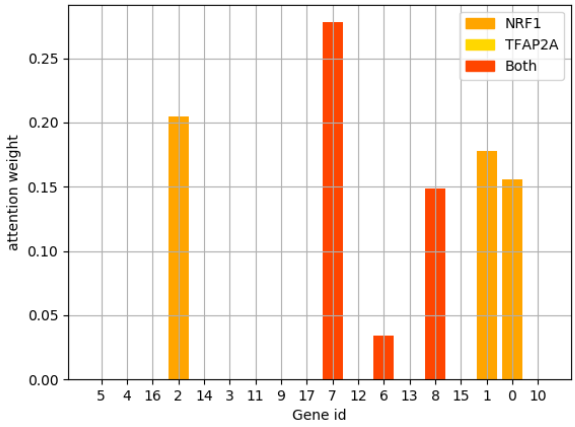


Fold5

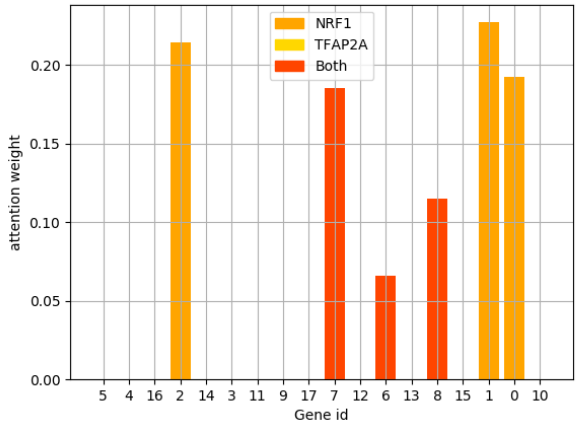
Figure 8. Attention weights for model with kmax-10



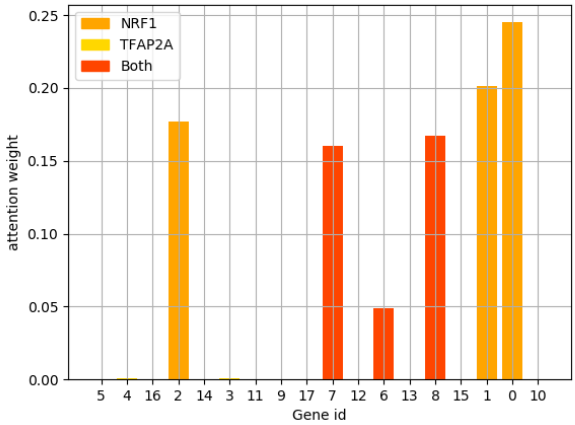
Fold1



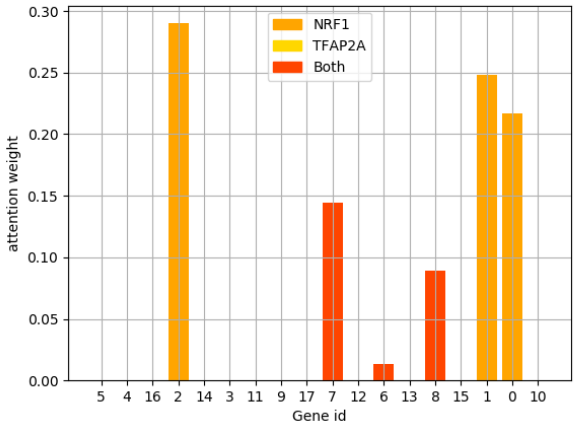
Fold2



Fold3



Fold4



Fold5

Figure 9. Attention weights for model with kmax-15