Predicting protein coding boundaries using RNNs

Saket Choudhary
skchoudh@usc.edu

Abstract

We explore how recurrent neural networks (RNNs) can be used to predict protein coding domains in a gene. We first demonstrate that using long short term memory RNNs give with one-hot encoding resulted in a limited prediction power. Later, we demonstrate how a word embedding approach along with bi-directional LSTMs gives promising results using the entire pool of protein coding genes in human achieving an overall accuracy of 0.67. This model is then used to predict protein coding domains in a different species, mouse, and achieves an overall accuracy of of 0.70 when tested on non-orthologous genes(where orthogonality implies a gene in mouse shares significant sequence from a human gene owing to descent from a common ancestor).

1. Introduction and Related Work

The central dogma of biology describes the flow sequential information from nucleic acid to nucleic acid and nucleic acid to protein. A protein coding gene can be partitioned into three regions: 5' UTR, CDS and 3' UTR (UTR = Untranslated region; CDS = coding domain sequence) (Figure 1). While the CDS is responsible for synthesis of protein, the UTR regions act as regulators and stabilizers. The exact boundaries of CDS and UTR are more or less well characterized for humans and mouse but is not available for non-model organism such as *C. albicans* [1]. Determining these exact boundaries involves preparing biological assays capable of capturing the expressed parts of genome and is a resource and time consuming step. Instead, we wanted to use the existing annotation to come up with a predictive model that can be used to annotate regions in non-model organisms. The essential problem we are trying to tackle here is a toned down version of gene prediction problem, which has been addressed in literature earlier. Chris Burge and Samuel Karlin's GENSCAN [2] uses a fifth-order markov chain to to identify these boundaries in genes. However, RNNs do not make any markovian assumption and hence are capable of handling long term dependencies where markov chain approaches will often fail.

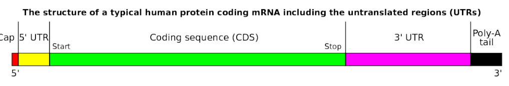
The structure of a typical human protein coding mRNA including the untranslated regions (UTRs)
Cap
5' UTR
Start
Coding sequence (CDS)
Stop
3' UTR
Poly-A tail
5'
3'

Figure 1. The three regions of interest in a gene. Image Credits : Wikimedia commons

2. Data

The human genome consists of $20,000$ protein coding genes with length ranging from 200-100,000. Sequences and their annotation were downloaded from ENCODE project's website (`https://www.encodeproject.org/`). Besides human, we also utilizes sequence from mouse and C. elegans as testing datasets.

3. Model and Results

The property of Long Short Term Memory (LSTMs) based RNNs to capture the sequential information is well suited for our problem. LSTMs have found extensive application in the field of natural language processing such as speech recognition [4] and text generation [3]. The human genome is 3×10^9 long made of 4 bases $\{A, C, T, G\}$ and as such re[resents a language in itself. Just like in speech or text prediction tasks where the goal is to learn the sequential pattern to predict the next occurrence of a word, our task here is to learn the sequential pattern of these bases to determine the boundaries of protein coding region in genes.

We iterated over different models since the initial few models had limited prediction power on the trainining dataset alone. The models are described in the following subsections.

3.1. Model 1: One-hot encoding with LSTM

Our first iteration of the model was a basic LSTM with 100 hidden units, followed by a fully connected layer with softmax activation (Figure 2). It was trained on 5000 human genes in human and was run for 40 epochs. Training was performed on a batch size of 1 with dropout rate set to 0.25. The input consisted of one-hot encoded sequences of length L such that each batch of size 1 was a $4 \times L$ matrix.

We use categorical cross entropy as the loss function. Though there are 20000 genes in total, we down-sampled

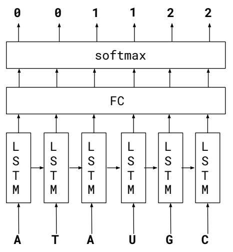

Figure 2. Model 1 : LSTM (100 hidden units) with fully connected (FC) layer with softmax activation. The input is one-hot encoded and is in the form of a $4 \times L$ matrix where L is the length of the sequence and $\{A, C, T, G\} = \{\langle 1,0,0,0\rangle, \langle 0,1,0,0\rangle, \langle 0,0,1,0\rangle, \langle 0,0,0,1\rangle\}$

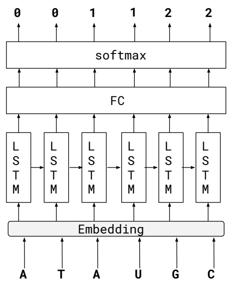

Figure 3. Model 2 : Embedding layer followed by LSTM (100 hidden units) with fully connected (FC) layer with softmax activation.

and used only 5000 genes for training. These 5000 genes were sampled proportionally from the entire pool of 20000 genes. Briefly, the genes were partitioned into 10 bins based on gene length ranging between 200 and 10000. Then we sampled genes from each bin proportional to the total number of genes in each bin such that the sum of sampled genes across all bins came out to be 1000. This approach has problems. In particular, a sample size of 1000 genes might be too little for RNNs to train. However, looking at the training curves for humans (Figure 8), the training error is upper bounded by around 0.54 which though is higher than 0.33 (worst case scenario for a 3 class classification problem), is too slow in learning. The testing accuracy curve (Figure 9) also shows saturation around 25 epochs. Though it is possible that the model will learn better by increasing the number of epochs, we decided to terminate the training at 40 epochs.

3.2. Model 2: Embedding layer with LSTM

Lee et al.[7] have argued that use of one hot encoding ($\{A, C, T, G\}$ are mapped to $\{\langle 1,0,0,0\rangle, \langle 0,1,0,0\rangle, \langle 0,0,1,0\rangle, \langle 0,0,0,1\rangle\}$) as in our Model 1 can result in limited generalization owning to the sparsity of the structure. They suggest using an embedding approach similar to *word2vec* [8] where the bases $\{A, C, T, G\}$ are randomly initialized to four dimensional dense vectors (say $A = \langle -0.1, -0.2, -0.3, 1\rangle$) and whose elements are in turn trained using gradient descent method.

Following Lee et al.'s suggestion, we modified our model to consist of a embedding layer followed by LSTM and the fully connected layer as in Model 1. This model resulted in improvement at both training (Figure 10) and testing steps (Figure 11).

3.3. Model 3: Embedding layer with bidirectional LSTM

LSTMs tend to 'memorize' input sequences that have passed through them using the hidden states and hence only preserve information about the past. Bidirectional LSTMs on the other hand follow a two pass strategy, where the first learning happens over sequences starting from the past and going to the future and the second pass involves learning the information starting from the future and going all the way back to the past. So using two hidden states, bidirectional LSTMs can learn information from both past and future. Such a property is useful in learning long term genomic dependencies arising in the transition from 5'UTR $\Longrightarrow$ CDS $\Longrightarrow$ 3'UTR.

Changing the LSTM units to bidirectional in Model 2 gives us Model 3 (Figure 4) which results in the highest training and test accuracies among the 3 models (Figures 12 and 13). Due to time constraints, we could only train Model 2 and Model 3 for around 10 epochs, which is nowhere close to saturation.

3.4. Prediction across species

Using Model 3 trained on human genes only, we used sequences from other species to asses its accuracy. Sequences were derived from mouse and C. elegans since the protein coding boundaries are close to completely annotated for these species. The model gave an average mean accuracy of 0.71 on entire pool of 20000 mouse genes and 0.70 on entire pool of 10800 protein coding genes in C.elegans. It is worth noting that this value is higher than the average mean accuracy of 0.67 that we obtained on the human

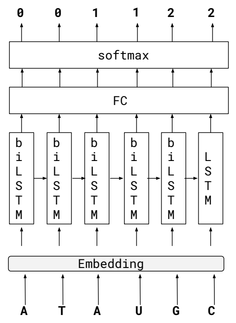

Figure 4. Model 3 : Embedding layer followed by LSTM (100 hidden units) with fully connected (FC) layer with softmax activation.

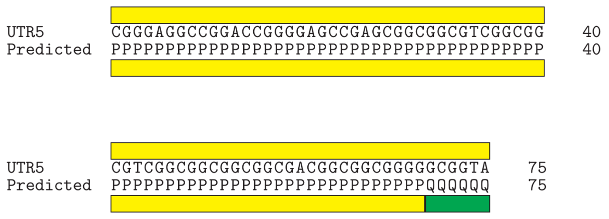
UTR5 CGGGAGGCCGGACCGGGGAGCCGAGCGGCGGCGTCGGCGG 40
Predicted PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP 40
UTR5 CGTCGGCGGCGGCGGCGACGGCGGCGGGGGCGGTA 75
Predicted PPPPPPPPPPPPPPPPPPPPPPPPPPPPPQQQQQQ 75

Figure 5. Human UTR5 prediction for ENSG00000107164. P=5'UTR, Q=CDS, R=3'UTR

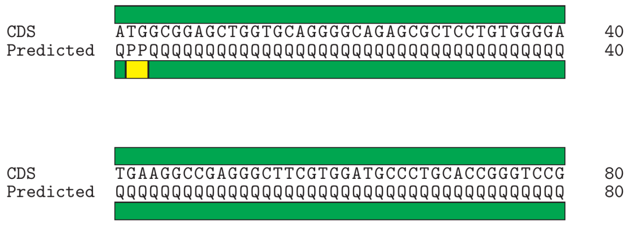
CDS ATGGCGGAGCTGGTGCAGGGGCAGAGCGCTCCTGTGGGGA 40
Predicted QPPQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQ 40
CDS TGAAGGCCGAGGGCTTCGTGGATGCCCTGCACCGGGTCCG 80
Predicted QQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQ 80

Figure 6. Human CDS prediction for ENSG00000107164. P=5'UTR, Q=CDS, R=3'UTR

dataset alone, given that the model was trained using human sequences only. Since humans have the most complex gene structure, it is possible that the mouse and C.elegans gene structure are subsets of those in human and hence result in higher accuracy as compared to the held out sequence in human, which is likely still different from the training set, even if it belongs to the same species.

The motivation to use these two species was to ensure that the model learns not just local, but global features that are useful for prediction across species when their common

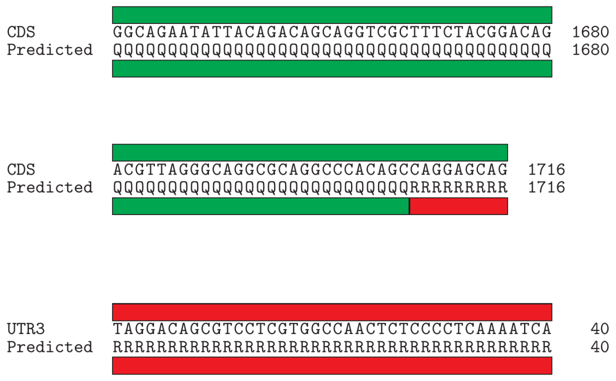
CDS GGCAGAATATTACAGACAGCAGGTCGCTTTCTACGGACAG 1680
Predicted QQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQ 1680
CDS ACGTTAGGGCAGGCGCAGGCCCACAGCCAGGAGCAG 1716
Predicted QQQQQQQQQQQQQQQQQQQQQQQQQQQRRRRRRRRR 1716
UTR3 TAGGACAGCGTCCTCGTGGCCAACTCTCCCCTCAAAATCA 40
Predicted RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR 40

Figure 7. Human UTR3 prediction for ENSG00000107164. P=5'UTR, Q=CDS, R=3'UTR

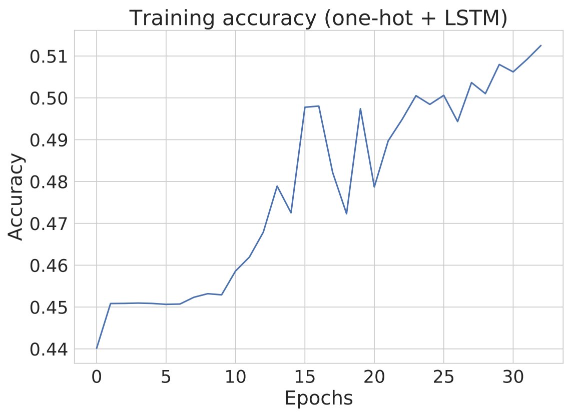

Figure 8. Model 1 training accuracy

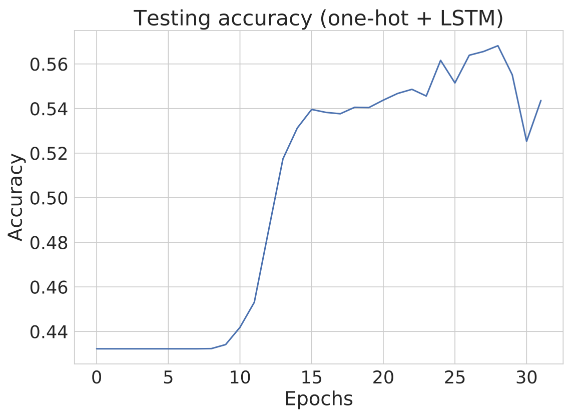

Figure 9. Model 1 testing accuracy

ancestors are too distant. Though human and mouse are close evolutionarily, the distance between C.elegans and hu-

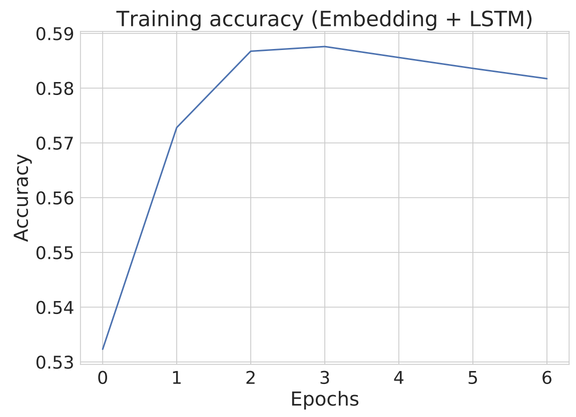

Figure 10. Model 2 training accuracy

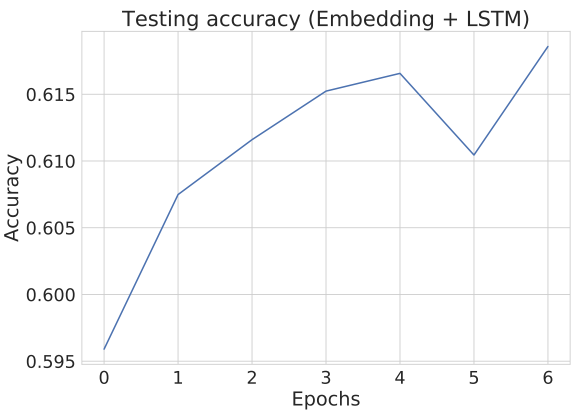

Figure 11. Model 2 testing accuracy

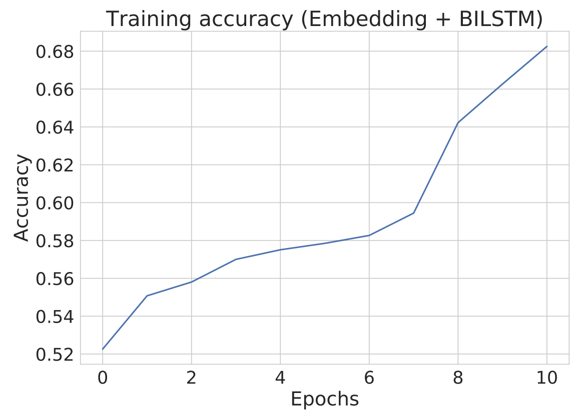

Figure 12. Model 3 training accuracy

mans/mouse diverged from a common ancestor 900 million years ago. Since the model gives comparable accuracies with all three species, it indicates, the model is capable of learning global features that determines the protein coding boundaries across species.

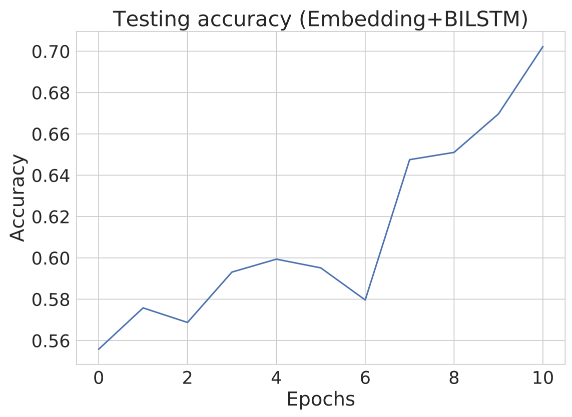

Figure 13. Model 3 testing accuracy

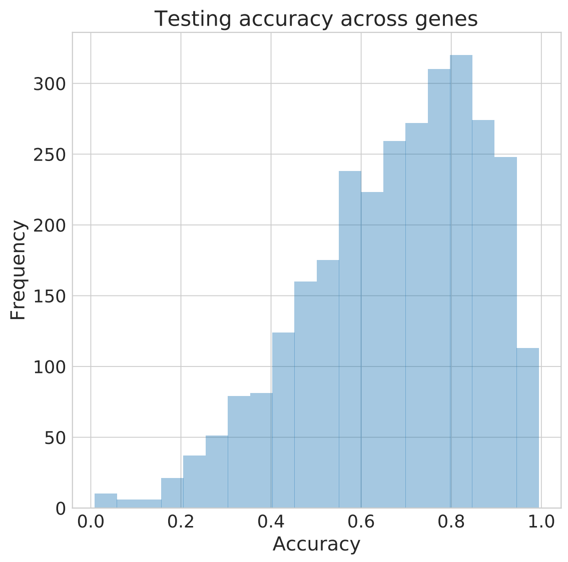

Figure 14. Model 3 testing accuracy distribution across genes in humans

4. Limitations and Possible improvements

LSTMS are time consuming to train. This resulted in us using only half of the entire pool of genes for training. Besides, the training was stopped at 10 epochs, even though the training curves did not indicate any saturation. One obvious improvement could have come through increasing the number of epochs. Besides this, we also came across other tricks that might result in a better accuracy, but could not be

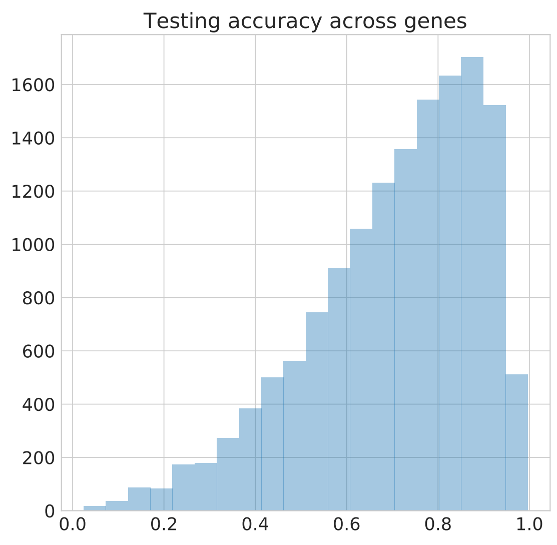

Figure 15. Model 3 testing accuracy distribution across genes in mouse

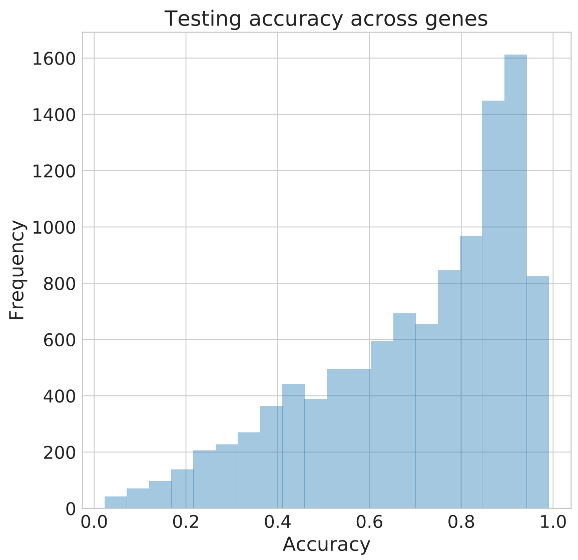

Figure 16. Model 3 testing accuracy distribution across genes in C. elegans

implemented due to time constraints. These are listed in the following subsections.

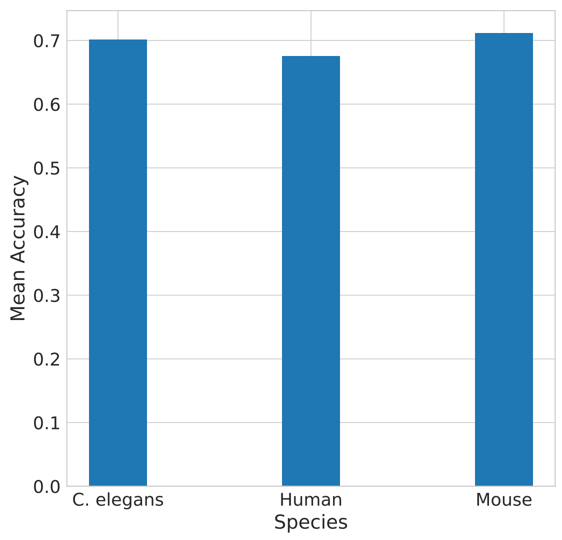

Figure 17. Model 3 average testing accuracy across species. Model was trained using 5000 genes in human only.

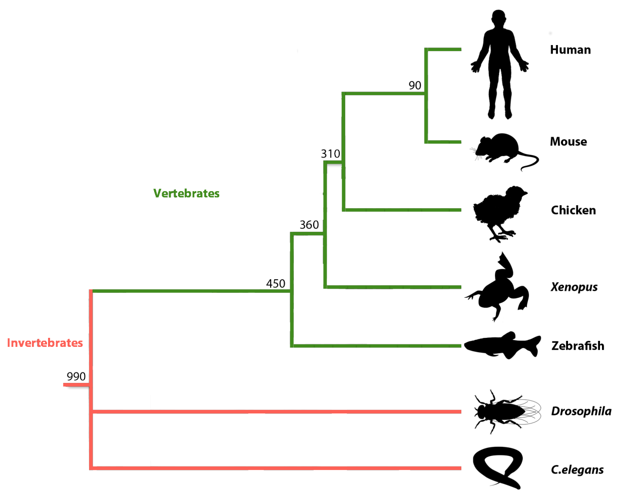

Figure 18. Phylogenetic tree indicating the divergence times in million years. Adapted from [10]

4.1. Using Autoencoders

Lee et al. [6] have made use of autoencoders to learn the inherent sequence over a stacked RNN model. The problem they are trying to address is similar in nature, where the idea is to learn which two patterns of sequence interact. Our current model (Model 3) might possibly still be learning features which are too local. An autoencoder based approach might likely help in learning these features in an unsupervised manner.

4.2. Dilated Convolutions

In a recent paper Gupta et al. [5] propose using dilated convolutions to model long term dependencies which claims to make the best out of convolution and bidirectional LSTM worlds, by ensuring a short path from input to output as in convolution and a large receptive field as in bi-directional LSTMs. the problem of handling long term dependencies is at the core of our problem definition. We might benefit from using this dilated convolution approach.

4.3. Conditional Random Fields

A recent paper by Wang et al. [9] combined Conditional Random Fields (CRFs) with deep neural networks to predict protein secondary structures. In spirit, the protein secondary structure prediction problem is similar to our problem where the objective is to identify class of structure given the protein sequence. The advantage of using CRFs over using HMMs to learn the interdependence between states is that it is discriminative approach and hence the independent assumption required by HMMs can be relaxed. CRFs seem the most promising improvement that could have potentially been included in the model.

5. Conclusion

We propose a bidirectional LSTM based approach to learn the boundaries of protein coding domains in gene. Our model achieved an overall mean accuracy of 0.67 on human dataset and 0.70 and 0.71 for mouse and C.elegans datasets respectively.

References

[1] V. M. Bruno, Z. Wang, S. L. Marjani, G. M. Euskirchen, J. Martin, G. Sherlock, and M. Snyder. Comprehensive annotation of the transcriptome of the human fungal pathogen Candida albicans using RNA-seq. *Genome Research*, 20(10):1451–1458, Oct. 2010. 00146.

[2] C. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *Journal of molecular biology*, 268(1):78–94, 1997. 03814.

[3] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013. 00820.

[4] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[5] A. Gupta and A. M. Rush. Dilated Convolutions for Modeling Long-Distance Genomic Dependencies. *arXiv preprint arXiv:1710.01278*, 2017. 00000.

[6] B. Lee, J. Baek, S. Park, and S. Yoon. deepTarget: End-to-end Learning Framework for microRNA Target Prediction using Deep Recurrent Neural Networks. pages 434–442. ACM Press, 2016. 00006.

[7] B. Lee, T. Lee, B. Na, and S. Yoon. DNA-level splice junction prediction using deep recurrent neural networks. *arXiv preprint arXiv:1512.05135*, 2015. 00005.

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 04660.

[9] S. Wang, J. Peng, J. Ma, and J. Xu. Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields. *Scientific Reports*, 6(1), May 2016. 00066.

[10] G. N. Wheeler and A. W. Brndli. Simple vertebrate models for chemical genetics and drug discovery screens: Lessons from zebrafish and *Xenopus*. *Developmental Dynamics*, 238(6):1287–1308, June 2009.