ChromaNet: Deep neural networks for chromatin profile prediction

Adithya Ganesh **Behrooz Ghorbani** **Philip Hwang**

Wendi Liu **Armin Pourshafeie** **Karen Yang**

Abstract

In this work, we apply deep neural networks to the problem of *de novo* chromatin profile prediction. Our analysis takes two broad approaches. First, to model long-term dependencies, we train a purely recurrent neural network. In particular, a bidirectional-LSTM network was used directly on the sequence, which outperformed a logistic regression baseline. Secondly, we train a convolutional neural network adapted from the DeepSEA architecture [1], to analyze the benefits of multitask learning. We use principal component analysis to identify clusters of tasks, and give evidence that training a network on related tasks improves PR-AUC performance relative to randomly selected tasks.

1 Introduction

Understanding the functionality of non-coding genomic variants is of paramount importance, as approximately 93% of disease associated variants are in non-coding regions [2]. Various methods have been developed to score the pathogenicity of particular variants. One example is CADD [3], which uses a support vector machine model to estimate pathogenicity based on various metrics that include chromatin structure, conservation metrics and transcription information.

More recently, Zhou et al. [1] and Quang et al. [4] applied deep learning methods to predict chromatin profile *de novo*, using only sequence information. The predicted chromatin profile can then be used to model functional effects. The DeepSEA model [1] uses three convolutional layers which serve as spatial motif detectors. In their DanQ model, Quang et al. use a single convolutional layer followed by a bi-directional LSTM layer to model long-term interactions. Both methods are designed to predict the chromatin profile for a 200-bp sequence window with a 400-bp of additional context on each side.

Because both DeepSEA and DanQ employ convolutional layers, they require a fixed input length (1000-bp). Zhou et al. [1] have demonstrated that training DeepSEA models with inputs of larger context size results in increased prediction accuracy (see Figure 1). This observation is consistent with the fact that spatially distant sites are known to exhibit co-dependent behavior. For example, the ZRS regulator, which plays a role in modulating anatomical structure of tetrapod limbs, lies 1Mb from the target gene *Shh* [5]. In this research, we demonstrate a proof-of-concept fully-recurrent network, which can be extended to model these very distant dependencies.

The networks in both DeepSEA and DanQ are trained in a joint multi-task fashion to simultaneously learn to predict transcription factor (TF) binding, DNase I-hypersensitive sites, and histone marks across multiple cell types. However, multitask learning does not always improve performance [6]. We are interested in characterizing the performance of different multitask learning formulations, and in particular determining whether certain tasks are antagonistic to model performance.

To address the ideas above, we trained two different neural network models: a) a convolutional neural network, and b) a bidirectional-LSTM recurrent network. Using the convolutional network, we find

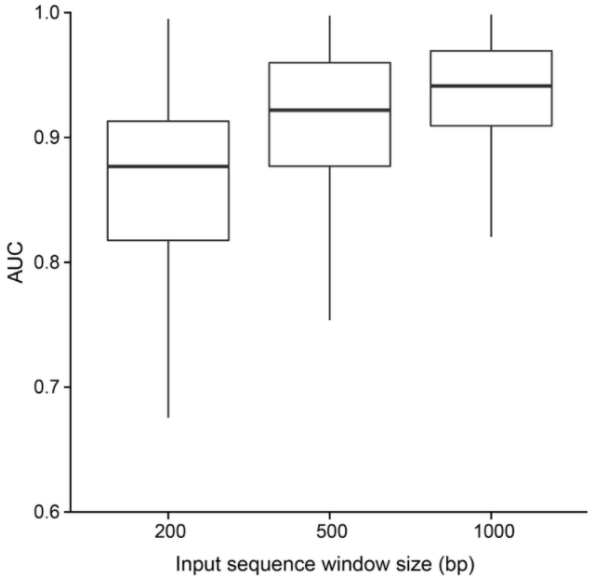


Figure 1: Prediction accuracy with different context length

that perhaps unsurprisingly, increasing the number of tasks increasing the performance. We also find preliminary evidence that suggests training similar tasks together improves test performance. Specifically, training DNase tasks together resulted in better performance than training randomly selected tasks.

We also trained a bidirectional LSTM network that outperformed a logistic regression baseline. This demonstrates an important proof of concept, as fully recurrent architectures are well suited to modeling long-term dependencies and variable-length inputs [7] [8] [9] [10].

2 Related Work

2.1 Chromatin profile prediction

Traditionally, analyzing genome sequences involves searching for common motifs or applying phylogenetic comparison. This often requires prior knowledge of motifs, expert annotations, or applying motif discovery algorithms [11] [12].

Several recent studies have adopted weighted k-mer scoring or gapped k-mer scoring to train a discriminative model to discover potential new motifs from known motif databases [13] [14] [15]. These studies use a set of k-mer features on ChIP-seq / DNase-seq peaks (positive examples) versus their flanks (negative examples), to predict the activity of sequence elements, typically ranging from 500 to 2000 bp in length. To eliminate the limitations of short k-mer frequency features in representing longer TB binding sites, gapped k-mers features are shown to significantly improve sequence classification accuracy.[13] Another approach that applies SVMs is the CADD algorithm [3], where Kircher et.al introduced C-scores that correlate with a range of features, including allelic diversity, disease severity, and complex trait associations.

2.2 Chromatin profile prediction with neural networks

A number of research efforts have applied deep neural networks to predict chromatin profile from sequence. While classical algorithms like support vector machines require manual preprocessing and feature selection, deep neural networks can adaptively extract features from the data during training [16].

Convolutional neural networks (CNNs) are one of the most successful deep learning architectures that have been applied to chromatin profile prediction, often out-performing classical methods such as gkm-svm [16] [1] [17]. For instance, DeepBind [17] uses 16 convolutional layers with window size of 24, a global max-pooling layer, and a fully connected layer of size 32 with a dropout layer. The DeepSEA model [1] uses a context sequence size of 1 kb, and uses a hierarchical architecture of three convolution layers with 320, 480 and 960 kernels, and adopts multi-task joint learning of diverse chromatin factors sharing predictive features in the final sigmoid output layer.

By contrast, recurrent neural networks (RNNs) are comparatively less studied. Bidirectional-RNNs [18] are capable of modeling interactions from future and past inputs. A variant of the BRNN model, the bidirectional Long Short-Term Memory (BLSTM) network, consists of gates which are added on each layer that control the degree of influence of information from the current step, and from the previous and next steps. This enables the model to capture the long-term temporal dependencies in sequential data with more control and flexibility. Quang et al. [4] used a combination of CNNs and bidirectional LSTMs to capture time dependencies on features extracted by convolutional operations.

2.3 Multi-task Learning

Multi-task neural architectures provide a joint learning framework for simultaneous feature sharing across many different but related learning tasks. In this setting, the network parameters are shared, with only the last layers of the network being task specific. It is expected that sharing deep layers would improve features produced by these deep layers, and thus improve generalization performance, as well as compensating for the imbalanced data distributions for one single task. Many recent empirical studies have shown the benefits of multi-task learning. Ramsundar et al. [19] found that multi-task networks can help in situations of imbalanced datasets that require special handling for effective learning. They found multi-task methods obtain predictive accuracies significantly better than single-task ones, and the presence of shared active compounds was correlated with multi-task improvement. However, multitask learning does not always improve performance [6], and thus it is often treated as a tool that requires testing on each application.

3 Dataset and pre-processing

Both of our frameworks use the same data used in the DanQ and DeepSEA models [4] [1]. To prepare this data, human reference genome (GRCH37) was segmented into non-overlapping 200bp windows. The binary labels for each task were determined by intersecting the windows with 919 ChIP-seq and DNase peaks determined by the ENCODE [20] and Roadmap Epigenomics [21] projects. Each region contains 400 flanking bases on each side for a total of 1000bp per region. Each window was one-hot encoded into a 1000×4 matrix. The dataset was augmented with the reverse complement of these regions. Training, validation and test sets were designed to contain strictly non-overlapping chromosomes. Due to computational limitations, we have sub-sampled the data as described in the corresponding section below.

4 Architecture

We performed experiments with a convolutional and recurrent neural network, which we discuss below.

4.1 Convolutional Network

In order to explore the performance characeristics of multi-task learning, we used a convolutional architecture similar to the DeepSEA model [1]. In particular, we use three convolutional layers with 320, 480 and 960 kernels from bottom to top. Each kernel has a window of length 8 with max-pooling of window size 4, and the final prediction is made after a fully connected layer.

While both [1, 4] used multi-task learning, the benefits of this approach for the problem of chromatin structure prediction has not been studied in detail. While the sheer number of proteins and cell-types requires a multi-task learning approach, the following problems are underexplored:

- The performance impact of adding tasks

- The performance impact of adding data
- The performance impact selecting tasks to be learned together

In this section of our project, we aim at investigating the performance impact of these choices.

We have used the data curated by DeepSEA [1], where 200bp sequences with 400bp flanking context, and their reverse complements are one-hot encoded. Due to computational limitations we used the first 1M regions for training (out of the total of 4.4M training regions used in previous work [1, 4]) and 200k random regions for testing (more than 400k regions provided). However, we keep the validation set as in the aforementioned works (8,000 regions).

4.1.1 Experiments with weighted cost function

Figure 2 shows the frequency of positive examples in our training data. Two facts are apparent from the figure:

1. Across all the tasks, the data is extremely unbalanced.
2. The number of positive examples is highly dependent on the nature of the task. For example, TF binding factors have extremely low number of positive responses while histone marks can have up to 20% positive examples.

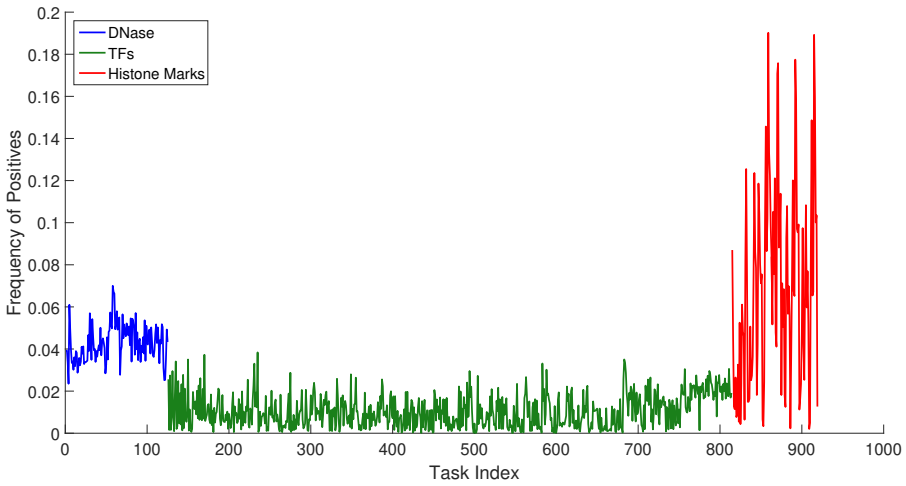


Figure 2: Frequency of positive labels for each task.

We experimented with the weighted negative-log-likelihood loss defined as

$$-\sum_{i=1}^{n}\sum_{j=1}^{m} \alpha w_j I_{\{y_{i,j}=0\}} \log(p_{i,j}) + (1-\alpha w_j) I_{\{y_{i,j}=1\}} \log(1-p_{i,j}) \tag{1}$$

where n is the number of observations, m is the number of tasks, and I is the indicator function. To normalize the loss, for task j, the following weight was chosen:

$$w_j = \frac{\text{Number of positive examples}}{n}$$

We added some regularization to this weighting scheme by adding a parameter α that was set to 1.1 after evaluating the performance of various choices on the validation set.

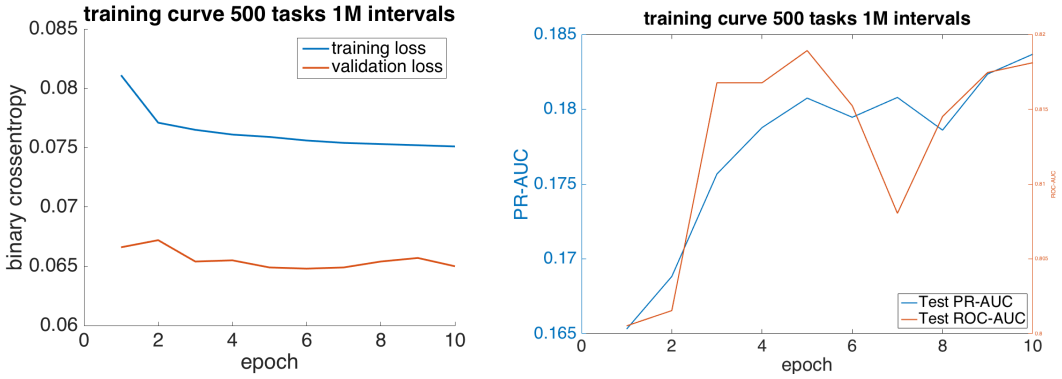
However, in our experiments, we were surprised to observe that adding the weights significantly hurts the results both in terms of ROC AUC and PR AUC. While further tuning of the weighting scheme could result in improved performance, for the remainder of the results of this paper, we have employed the un-weighted loss function.

4.1.2 Regularization

DeepSEA uses dropout as well as L_1 and L_2 regularization to avoid overfitting. We experimented with various choices of dropout, but we found that the model with no regularization performs the best in our case.

We trained each model for 10 epochs, while applying early stopping. As seen in Figure 3 the learning tapers off fairly quickly. Although we did not explicitly regularize our model, we do not see signs of overfitting. It must be noted that the surprisingly low validation loss could be a result of the size of the validation set. The only form of regularization we have used is early stopping and limiting the training to 10 epochs.

Figure 3 shows the learning curve for one such training. We have used the same validation set as in [1, 4].



(a) (b)

Figure 3: a) Training and validation loss for an example with 500 tasks and 1M regions. b) PR-AUC and ROC-AUC for the same example calculated on the test set

4.2 Bidirectional Long Short-Term Memory Network

Hyperparameters for our LSTM network were adapted from the choices made in the DanQ model [4]. Namely, we made the LSTM bidirectional in order to incorporate sequence data flanking the current input on each side. Then, we applied a dropout regularization of 0.5 to the output of this bidirectional LSTM layer. Finally, we added a fully connected layer with ReLU input and sigmoid output activation functions to yield our multi-task output. This model was trained subject to an unweighted binary cross-entropy loss function using the RMSProp optimizer. Due to computational limitations, 100K regions were used for training and testing.

5 Results

We will split the results into two sections. In the first section we present our analysis of the performance characteristics of multitask learning. Next, we will discuss our results training a Bidirectional LSTM directly on the input sequence.

5.1 Multi-task learning performance analysis with CNN

The results of runs from the model above are presented in the table 1. In all cases, validation loss did not show signs of over-fitting. The table presents the PR-AUC on the training test for the model at the 10th epoch. The training regions are constant across rows and the larger multi-task settings include all the tasks from smaller settings.

From these results, we see that the performance increases as the number of tasks considered increase. Further analysis could investigate 1) whether this increase tapers off after a certain number of tasks and 2) how much this increase changes with the amount of data available.

	100 tasks	300 tasks	500 tasks
750k regions	–	0.175	–
1Mk regions	0.168	0.191	0.184

Table 1: Precision-recall AUC on the testing set for various combinations of training sizes and tasks sizes

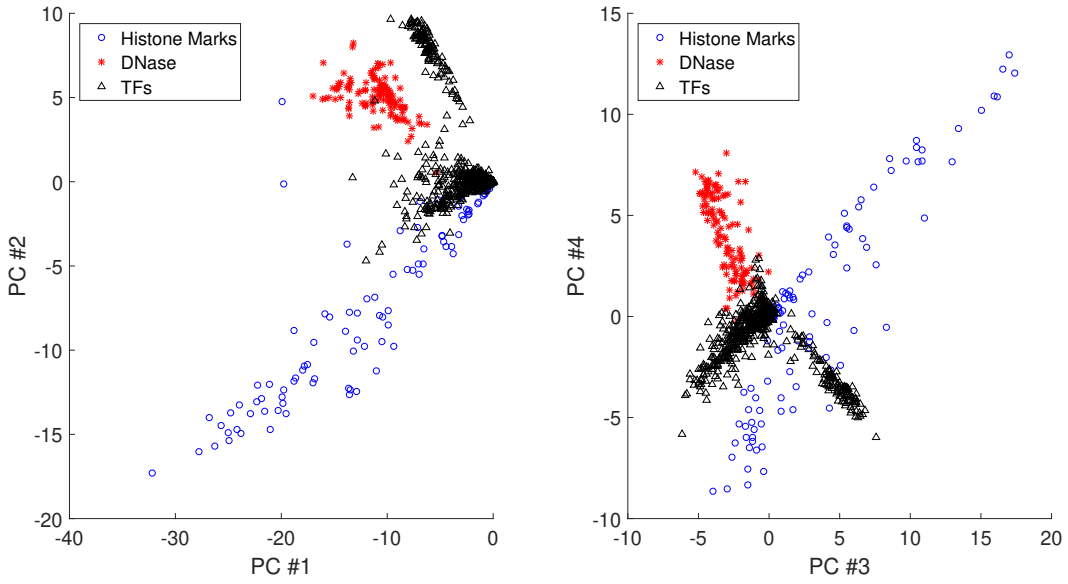


Figure 4: PCA plot of binding location of each task calculated from the first 10,000 regions. DNASE forms a clear cluster, TFs form two separated clusters and Histone markers do not cluster well.

Furthermore, as we discussed earlier, the extent of imbalance in these tasks is different. The presence of different clusters within the tasks can be further observed after dimensionality reduction. From Figure 4 we can clearly detect two separate clusters for TF's and one cluster for DNase. Histone markers do not seem to cluster very well.

We ran an experiment training our network on 100 DNase tasks. The PR-AUC results can be seen in Table 2. We have compared the average PR-AUC from the DNase-only training to 1) the average results from 100 random tasks training and 2) the average only on the shared subset (size = 11) between our random task training and our DNase-only tasks training. Note that in all cases the training task size is 100. The results can be seen in Table 2. This provides preliminary evidence that training on similar tasks is beneficial to performance.

	100 DNAse only	100 random tasks	100 rnd. tasks Eval. on DNase only
1M regions	**0.258**	0.168	0.222

Table 2: Average PR-AUC for different multi-task learning formulations with 100 tasks and 1M regions.

5.2 Bidirectional LSTM Network

To analyze the model of our bidirectional-LSTM network, we computed 919 receiver operating characteristic and precision-recall curves. As noted in [4], PR curves in particular often provide a more robust measure of performance when training data exhibits class-imbalance.

Below we report the distribution of the area under each ROC curve (Figure 5), as well as each PR curve (Figure 6). The AUC PR values were compared to task-specific logistic regression baselines mentioned in the DeepSEA paper [7]. Notably, the bidirectional LSTM model outperforms this baseline even when trained on a small subset of the available training data.

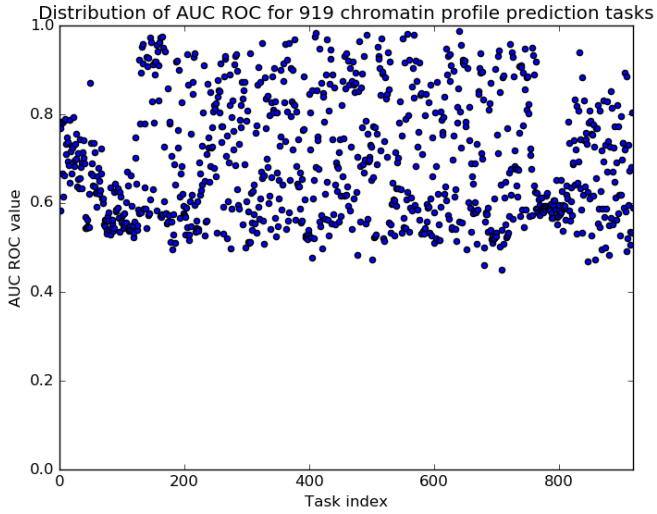


Figure 5: Distribution of AUC ROC for 919 chromatin profile prediction tasks

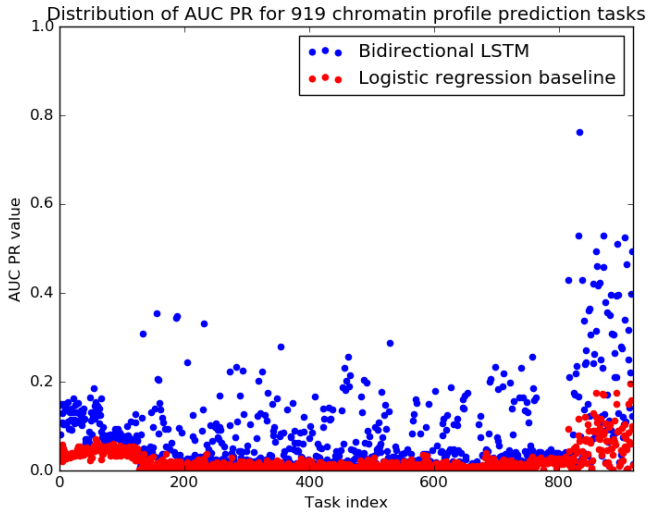


Figure 6: Distribution of AUC PR for 919 chromatin profile prediction tasks

We have also produced example ROC curves for particular tasks. In the figures below, we present the model's performance on task 639, predicting NELFe transcription factor binding on the K562 cell type, as well as task 493, predicting ZNF263 transcription factor binding on the HEK293-T-REx cell type. Task 639 was the best performing in terms of AUC ROC, while task 493 was a randomly selected task, closer to the average AUC ROC value.

Similarly, we present example PR curves for individual tasks. In the figures below, we display the model's performance on task 832, predicting H3K4me3 methylation on the H1-hESC cell type, as well as task 848, predicting H3K4me3 methylation from the K562 cell type. Task 832 was the best performing task in terms of AUC PR, while task 848 was a randomly selected task, more representative of average performance.

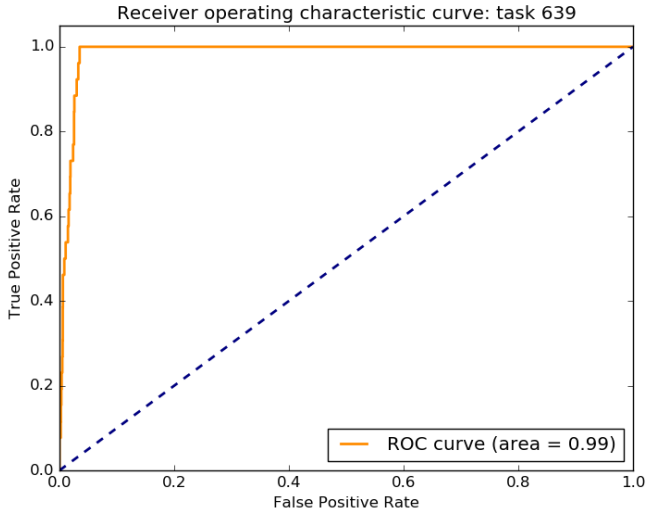


Figure 7: ROC curve: task 639, predicting NELFe transcription factor binding on the K562 cell type

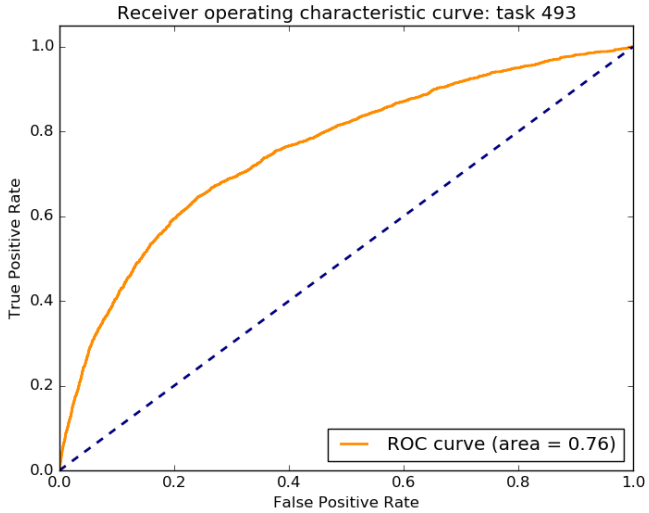


Figure 8: ROC curve: task 493, predicting ZNF263 transcription factor binding on the HEK293-T-REx cell type

6 Conclusion and Future Work

We have demonstrated a proof-of-concept for applying fully recurrent neural networks to the problem of multi-tasked chromatin profiling prediction. Namely, our promising results for the bidirectional LSTM network suggest that an intermediate convolutional neural network layer might not be necessary. By using extensions of this fully recurrent model, we can model input sequences of arbitrary length and model long range interactions.

Furthermore, in this work we have also analyzed the performance characteristics of multi-task learning. We see that multi-task learning and increasing the number of tasks does indeed improve the results. Furthermore, we studied the distribution of clusters that may be well suited for mutual learning. Specifically, we show that DNase, and TF tasks seem to form well defined clusters and we produce hints that learning based on these boundaries may be particularly beneficial.

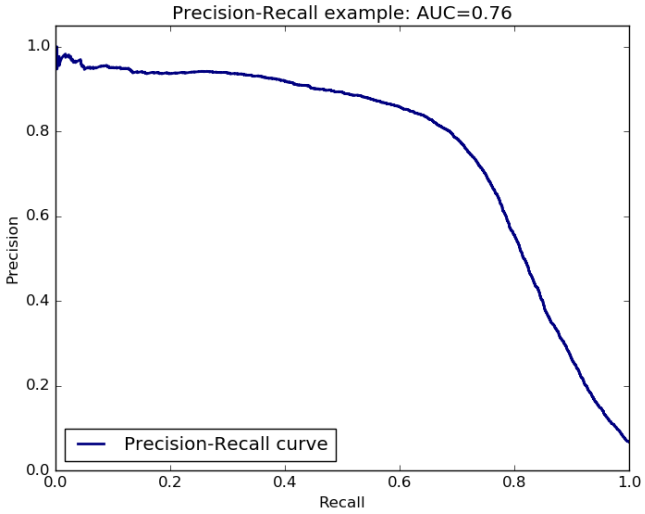


Figure 9: PR curve: task 832, predicting H3K4me3 methylation on the H1-hESC cell type

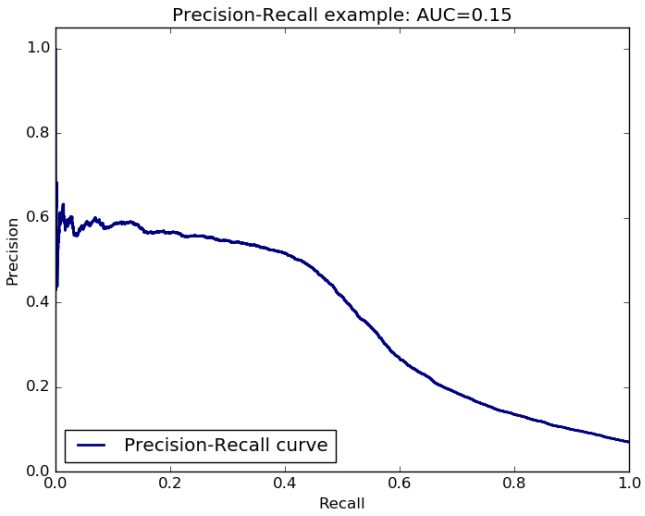


Figure 10: PR curve: task 848, predicting H3K4me3 methylation from the K562 cell type

In the future, we would like to further investigate the multi-task learning curve. Our results hint that after 300 tasks the benefits may be small however, more extensive experimentation is needed to account for the stochasticity of learning. Additionally, we would like to experiment with other sets of tasks that may be particularly well-suited for multi-task learning (e.g. tasks within a cell-type). We also need to further study our current hypothesis that multi-task learning segmented by the type of task is a beneficial formulation. More hyper-parameter search and repetitions would help quantify the extent/existence of a signal.

For our recurrent model, we would like to refine our performance by training on the entire data set and performing a more extensive hyper-parameter search. Additionally, it might prove useful to experiment more with our architecture by training a deeper model. For these experiments we have relied on data provided by previous works which contain a 1000bp context size. For future work, the efficacy of LSTM's in learning long range interactions can be examined by increasing this window size.

We would also like to extend our fully recurrent model to other variants of RNNs. Namely, we would like to investigate the suitability for Clockwork RNNs on the present problem. Clockwork RNN (CWRNN), a new variant of RNN architecture [22] have recently been introduced by Schmidhuber et al. CWRNN utilizes clocked module activation and divides hidden layers into modules with distinctive temporal granularity, and it embodies both short-term and long-term dependencies by specializing units that operate on a defined timescale. This reduces the number of overall RNN parameters, and the empirical results shows significant improvement and acceleration across several language modeling training tasks. In particular, they outperform standard RNNs and LSTMs on sequence generation form audio waveform, and spoken word classification experiments. We think this framework might be potentially suited for the task of chromatin profile prediction in which long-range interactions are present. While CWRNN has received some success in analyzing language waveform data, which have implicit periodic structure, its performance on gene sequential data has not been explored. More research needs to be conducted to study the optimal time scaling mechanisms as well its applications in the chromatin profile prediction tasks with genome data.

References

[1] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature methods*, 12(10):931–934, 2015.

[2] Matthew T Maurano, Richard Humbert, Eric Rynes, Robert E Thurman, Eric Haugen, Hao Wang, Alex P Reynolds, Richard Sandstrom, Hongzhu Qu, Jennifer Brody, et al. Systematic localization of common disease-associated variation in regulatory dna. *Science*, 337(6099):1190–1195, 2012.

[3] Martin Kircher, Daniela M Witten, Preti Jain, Brian J O'Roak, Gregory M Cooper, and Jay Shendure. A general framework for estimating the relative pathogenicity of human genetic variants. *Nature genetics*, 46(3):310, 2014.

[4] Daniel Quang and Xiaohui Xie. Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences. *Nucleic acids research*, page gkw226, 2016.

[5] Laura A Lettice, Simon JH Heaney, Lorna A Purdie, Li Li, Philippe de Beer, Ben A Oostra, Debbie Goode, Greg Elgar, Robert E Hill, and Esther de Graaff. A long-range shh enhancer regulates expression in the developing limb and fin and is associated with preaxial polydactyly. *Human molecular genetics*, 12(14):1725–1735, 2003.

[6] Rich Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998.

[7] Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. Deep recurrent models with fast-forward connections for neural machine translation. *arXiv preprint arXiv:1606.04199*, 2016.

[8] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[9] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *EMNLP*, volume 3, page 413, 2013.

[10] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[11] Anthony Mathelier, Oriol Fornes, David J Arenillas, Chih-yu Chen, Grégoire Denay, Jessica Lee, Wenqiang Shi, Casper Shyr, Ge Tan, Rebecca Worsley-Hunt, et al. Jaspar 2016: a major expansion and update of the open-access database of transcription factor binding profiles. *Nucleic acids research*, page gkv1176, 2015.

[12] Jie Wang, Jiali Zhuang, Sowmya Iyer, XinYing Lin, Troy W Whitfield, Melissa C Greven, Brian G Pierce, Xianjun Dong, Anshul Kundaje, Yong Cheng, et al. Sequence features and chromatin structure around the genomic regions bound by 119 human transcription factors. *Genome research*, 22(9):1798–1812, 2012.

[13] Mahmoud Ghandi, Dongwon Lee, Morteza Mohammad-Noori, and Michael A Beer. Enhanced regulatory sequence prediction using gapped k-mer features. *PLoS Comput Biol*, 10(7):e1003711, 2014.

[14] Dongwon Lee. Ls-gkm: a new gkm-svm for large-scale datasets. *Bioinformatics*, page btw142, 2016.

[15] Jeong-Hyeon Choi and Hwan-Gue Cho. Analysis of common k-mers for whole genome sequences using ssb-tree. *Genome Informatics*, 13:30–41, 2002.

[16] David R Kelley, Jasper Snoek, and John L Rinn. Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome research*, 2016.

[17] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 2015.

[18] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, pages 2673–2681, 1997.

[19] Bharath Ramsundar, Steven Kearnes, Patrick Riley, Dale Webster, David Konerding, and Vijay Pande. Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072*, 2015.

[20] ENCODE Project Consortium et al. An integrated encyclopedia of dna elements in the human genome. *Nature*, 489(7414):57–74, 2012.

[21] Anshul Kundaje, Wouter Meuleman, Jason Ernst, Misha Bilenky, Angela Yen, Alireza Heravi-Moussavi, Pouya Kheradpour, Zhizhuo Zhang, Jianrong Wang, Michael J Ziller, et al. Integrative analysis of 111 reference human epigenomes. *Nature*, 518(7539):317–330, 2015.

[22] Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A clockwork rnn. *arXiv preprint arXiv:1402.3511*, 2014.