**Effective Gene Expression Prediction from Sequence by Integrating Long-Range Interactions**

The way the non-encoding DNA determines the gene expression in different cells is a major issue that this paper tries to solve. The prediction of thousands of epigenetic and transcriptional datasets in a multitask setting across long DNA sequences is the problem that is being solved. In order to capture extended sequences and determine whether the regulatory DNA elements affect expression at greater distances, the paper expressed a need for a fundamental architectural modification. A deep learning architecture called the Enformer is developed to improve the accuracy of gene expression prediction from the DNA sequencing. The main application of this model is that it predicts the changes in genetic variants, which will alter the gene expression. Earlier studies on gene expression used CNNs as building blocks, example, Basenji2 used dilated convolutions. Due to their local receptive field, the convolutional layers require many successive layers to reach distal elements. However, these are less effective, and the accuracy and usefulness has been hampered because of the effect distal enhancers have on the gene expression.

The enformer is a neural network architecture based on self-attention and is basically formed by combining the qualities of the enhancer and the transformer. The enformer integrates the information from long range interactions in the genome and the predictions were made directly from DNA sequences. The model architecture describes the influence of enhancers on gene expression from further away within the DNA sequence. The Enformer architecture consists of three parts: 7 convolutional blocks with pooling, 11 transformer blocks, and a cropping layer followed by final pointwise convolutions. The convolutional block with pooling reduces the spatial dimension. The transformer layers increase the receptive field, which expands the number of relevant enhancers seen by the model and capture long-range interactions across the sequence. The convolution and the transformer layers basically summarize and compress information in the DNA sequence. The transformer layers' byproduct called the attention matrix, which shows the information routing of the transformer. The cropping layer trims to avoid computing the loss on the far ends on each side. Output heads predict organism-specific tracks. Relative positional encodings are added to inject positional information. They provide a parameterized baseline for how actively two positions in the sequence should influence each other during the layer’s transformation as a function of their pairwise distance. Only DNA sequence as input to the enformer and it is one-hot encoded. Three different scores - namely gradient × input, attention, and in silico mutagenesis, are computed to prioritize enhancer–gene pairs with sequence-based models. Contribution scores were used to highlight which parts of the input sequence were most influential for the prediction in order to better understand how Enformer interprets the DNA sequence to make more accurate predictions. The attention matrix indicates that regions with the most attention are the known promoters and topologically associated domains (TADs), which are known to influence gene expression.

Enformer beats earlier versions in predicting how genetic variations, both natural and synthetic ones that alter critical regulatory areas, affect gene expression. The accuracy of the predictions of mutation effects is more. When making predictions, the enformer pays attention to biologically significant regions, such enhancers, and that gene expression contribution scores can be used to prioritize essential enhancers, with a high accuracy. Insulator elements also were learnt by the enformer. In addition to this, the enformer has several other advantages including that fact that does not rely on evolutionary conservation statistics and its capability of signed prediction of activating or repressive mutations. Also, the predictions are not just limited to conserved enhancers, which comprise a small proportion of all enhancers. The model is not trained to explicitly locate enhancers, so the enformer can also be used for arbitrary sequence variations lacking experimental data. However, this approach has the drawback of being unable to generalize to new cell kinds or assays and can only model and forecast for cell types and assays in the training data. By using the increasing amount of functional genomic datasets, to train the model, the sensitivity of the model towards genetic variants can be further enhanced. Further, Enformer can also be used to include prioritizing rare or de novo mutations seen in rare illnesses, fine-mapping current GWAS data, and imputing regulatory activity across species to examine cis-regulatory evolution.

**Which Tasks Should Be Learned Together in Multi-Task Learning?**

Using multi-task learning, a neural network can be trained to solve multiple tasks at once. However, task objectives can compete, which frequently results in subpar overall performance. As a result, when using multi-task learning, which tasks should and shouldn't be taught together in one network is a question that is trying to be solved. The task cooperation and competition in a variety of learning settings is being studied. A framework for assigning tasks to a few neural networks is proposed so that cooperating tasks are computed by the same network while competing tasks are computed by different networks. The main objective is to identify a set of networks, each trained on a subset of the tasks, that produces the lowest total loss given a certain computational budget. An empirical study of the number of factors influencing multitask learning such as the dataset and network size and how the tasks when learned together influence each other is being provided in this paper.

The multi-task learnability is studied by task groupings. The paper found that they are inversely correlated. Given the initial candidate set of a fully trained network, to solve a subset of the task T, the goal is to choose a subset of the initial candidate set, to solve all the given tasks with a lowest overall loss and total inference time under the budget. The paper proposes two methods to reduce the burden of computation – Early Stopping and High Order Approximation. In High Order Approximation all the single task and dual task models are trained and used to approximate higher order grouping. This results in the reduction of the computation from exponential combination to quadratic. Using this method, the performance of all the networks with 3 or more tasks can be predicted using the performance of all the fully trained two task networks. First, all the networks are trained with two or fewer tasks to convergence. Then, the performance of higher-order networks is being predicted. Later this network selection is run on both the trained networks and the predicted networks. Later the higher order networks are trained from scratch.  In early stopping, at the 0.2 epoch, the validation score was 0.2 which correlates with the final score very well. This in turn saves the computation resource 20x. Also, the optimal grouping is much better than a single or multiple single-task network. For instance, the best strategy discussed is training 2.5 and 2 full sized networks with 2 tasks each and the third-half sized network is trained on the fifth task.  To gain optimal performance for the first four tasks, the fifth task is required to regularize the first two full-sized networks. Another key idea presented in this paper is that the grouping of the tasks is inversely correlated with the task transferability. One of the main key insights from the paper is that the inclusion of an additional task can improve the accuracy. This can be achieved on the tasks that are existing even though the performance of the added task is poor.

The advantage of this paper is that the methods described here analyze how the tasks interact with each other and this method outperforms the single-task-networks and other baselines as well. The multitask affinities depend usually on many factors like the size of the dataset and network capacity and the computational framework like this will be helpful in deciding which tasks should be learnt together in multi-task learning. However, despite the Early Stopping Approximation being competitive, the solutions tended to be worse than optimal. Also, after being trained on more data, networks with adaptive learning rates ceased to be comparable.