Predicting gene expression using millions of random promoter sequences
DREAM Challenge 2022
SBME UBC
IBM Research
Google Research TPU Research Cloud
SageBionetworks

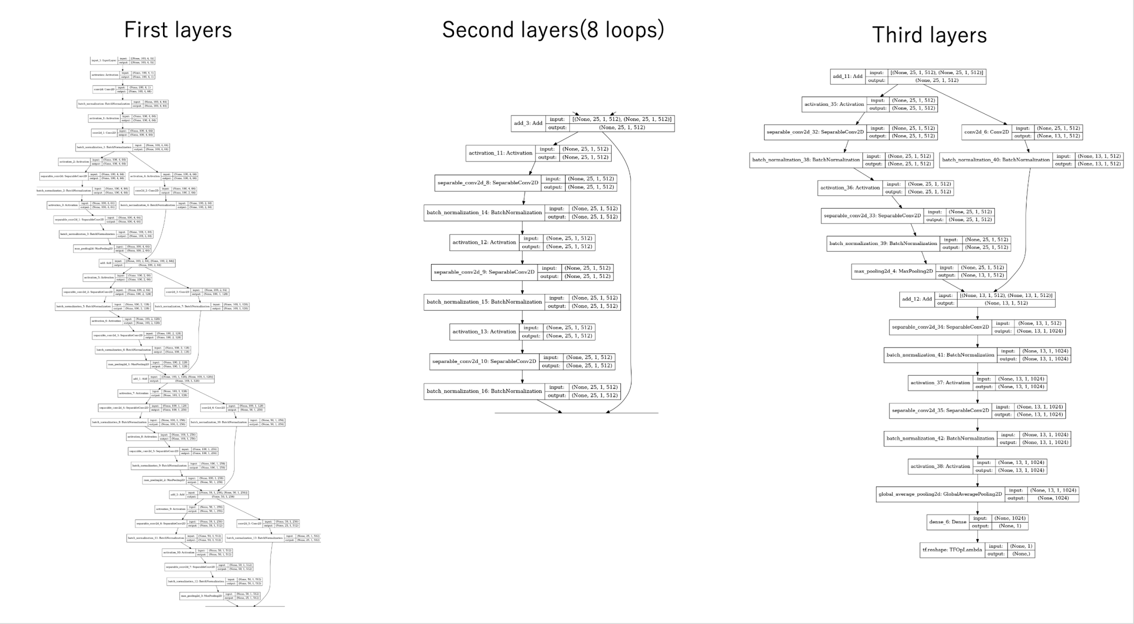
DREAM Challenge 2022
Predicting gene expression using millions of random promoter sequences by NGT4

1. Description of data usage

- Split data 8:2 for training and validation
- At approximately 80 bases add 'AACTGCATTTTTTTCACATC' before and 'GGTTACGGCTGTTTCTTAAT' after. In order to make all 120 bases, sequences longer than 120 were deleted after the 121st position, and those shorter than 120 were added with N at the end to make 120 bases.
- Of the 120 bases, the 10th to 110th bases were randomly slid in each sequence every 1 epoch to enhance the data.
- For the expression level, using a trained model and ranking the expression level, which is an integer value, within the same integer value (assumed to be N), evenly in the range of N – 0.5 < x < N + 0.5 The corrected expression level x was set so as to be distributed. Then, all the expression levels were ranked, and based on that, the values were re-set so as to be evenly distributed in the range of 0 or more and less than 1.
- A -> [1,0,0,0],C -> [0,1,0,0],G -> [0,0,1,0],T -> [0,0,0,1],N -> [0,0,0,0] to represent one array as a two-dimensional list. (Example: ACGT → [[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1]])

2. Description of the model

Created with reference to the Xception model. Please refer to the figure below for the model structure. If the resolution is not enough to read it, please refer to model.png in the submitted file.


First layers
Second layers(8 loops)
add_3: Add
activation_11: Activation
separable_conv2d_8: SeparableConv2D
batch_normalization_14: BatchNormalization
activation_12: Activation
separable_conv2d_9: SeparableConv2D
batch_normalization_15: BatchNormalization
activation_13: Activation
separable_conv2d_10: SeparableConv2D
batch_normalization_16: BatchNormalization
Third layers
add_11: Add
activation_35: Activation
separable_conv2d_32: SeparableConv2D
conv2d_6: Conv2D
batch_normalization_38: BatchNormalization
batch_normalization_40: BatchNormalization
activation_36: Activation
separable_conv2d_33: SeparableConv2D
batch_normalization_39: BatchNormalization
max_pooling2d_4: MaxPooling2D
add_12: Add
separable_conv2d_34: SeparableConv2D
batch_normalization_41: BatchNormalization
activation_37: Activation
separable_conv2d_35: SeparableConv2D
batch_normalization_42: BatchNormalization
activation_38: Activation
global_average_pooling2d: GlobalAveragePooling2D
dense_6: Dense
tf.reshape: TFOpLambda

3. Training procedure

Loss function uses MSE

The optimizer uses Adam and the learning rate is 0.0001

Record the correlation coefficient and, if this does not improve, move the correct label 10% closer to the predicted label. This is repeated twice, and the model with the best correlation coefficient is used as the final weight.

The training results are as followstrain_loss: 0.02583 ,val_loss: 0.02698 ,train_R: 0.9135, val_R: 0.8961

4. Other important features

5. Contributions and Acknowledgement

5.1 Contributions

Name	Affiliation	Email
Yu Hiratsuka	4th year student, Niigata University School of Medicine	m19a074k@mail.cc.niigata-u.ac.jp
Mao Takatsu	3th year student, Niigata University School of Medicine	mao.takatsu@gmail.com

5.2 Acknowledgement

Professor Okuda of the Medical AI Center, Niigata University School of Medicine, provided useful advice on how to proceed with research and the framework.
I would like to take this opportunity to express my deep gratitude.

6. References

Carl G. de Boer .Deciphering eukaryotic gene-regulatory logic with 100 million random promoters, nature aeticles