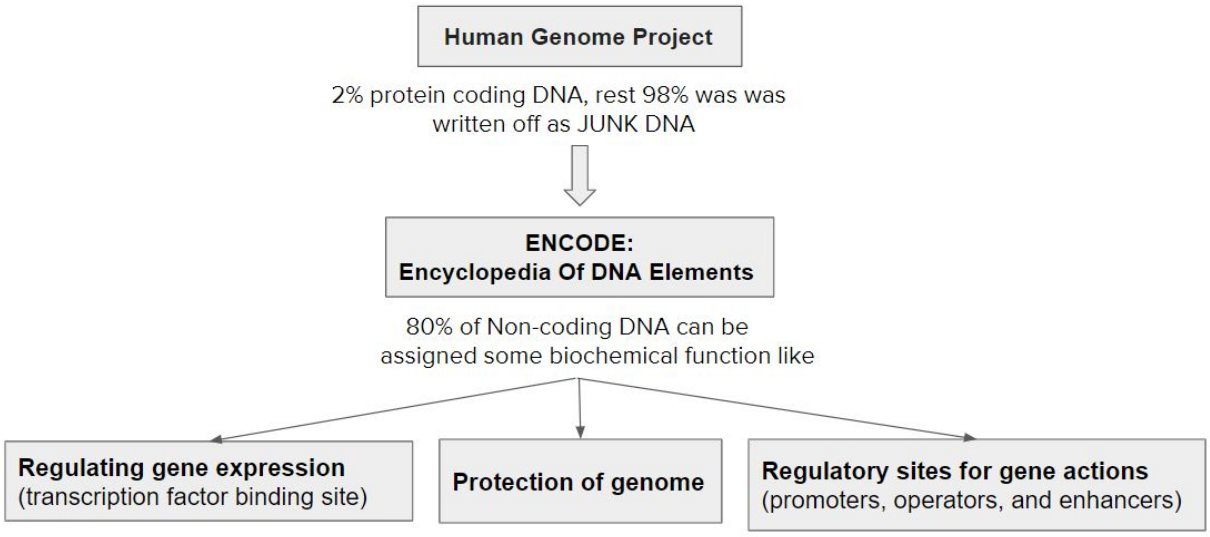
ACB Project Report

Identification of transcription factor-binding motifs using convolution neural network (CNN)

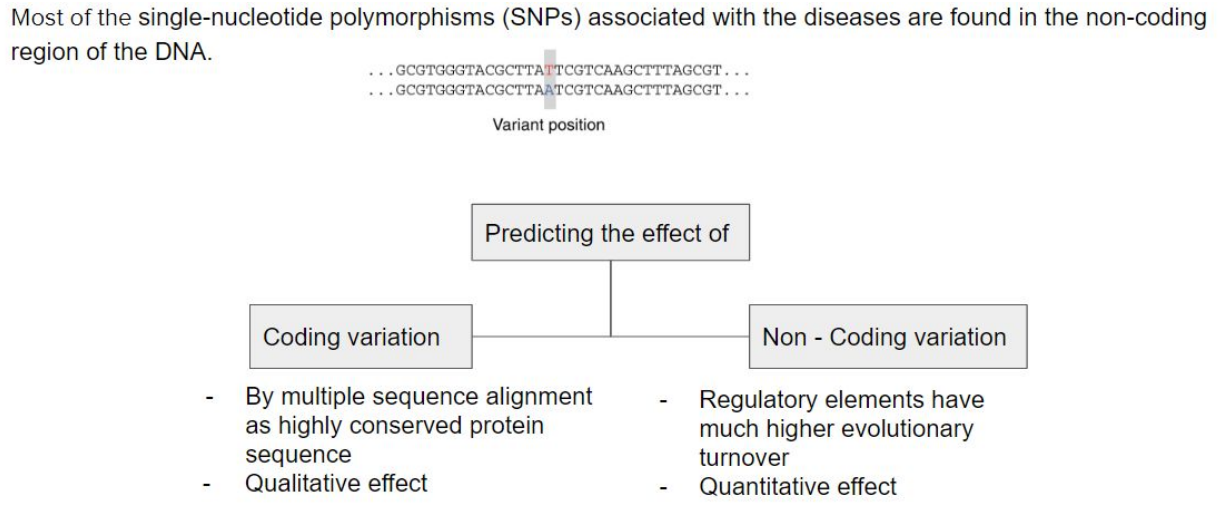
Sai Krishna Vamshi - 2016033
Tushar Dhyani - MT19221
Omkar Chandra - PhD17206

Motivation and Problem statement



Noncoding DNA separate genes from each other with long gaps, so alteration in one gene or part of a chromosome does not extend to the whole chromosome. In high genomic complexity like in case of human genome, not only different genes, but also inside one gene there are gaps of introns to protect the entire coding segment to minimize the changes caused by changing part of the gene.

Why Non-Coding DNA is important ?



Modeling the properties and functions of non-coding DNA sequences is particularly difficult, the vast majority of which is still poorly understood in terms of function. We need models to predict the function directly from the genome sequence and predict with single-nucleotide sensitivity the effects of noncoding variants.

Problem Solving: Identifying the TF-binding motifs using a predictive model.

Data Acquisition & Understanding the Data

Data Source:

- DeepSea:http://deepsea.princeton.edu/media/code/deepsea_train_bundle.v0.9.tar.gz
- Supplementary-Data:https://academic.oup.com/nar/article/44/11/e107/2468300#supplementary-data

Understanding the Data:

- Training labels were computed from uniformly processed ENCODE and Roadmap Epigenomics data releases.
- GRCh37 reference genome into 200-bp bins. For each bin they computed the label for all 919 chromatin features; a chromatin feature was labeled 1 if more than half of the 200-bp bin is in the peak region and 0 otherwise.
- Each training sample consists of a 1,000-bp sequence from the human GRCh37 reference genome centered on each 200-bp bin and is paired with a label vector for 919 chromatin features.

Sequence converted into binary matrix

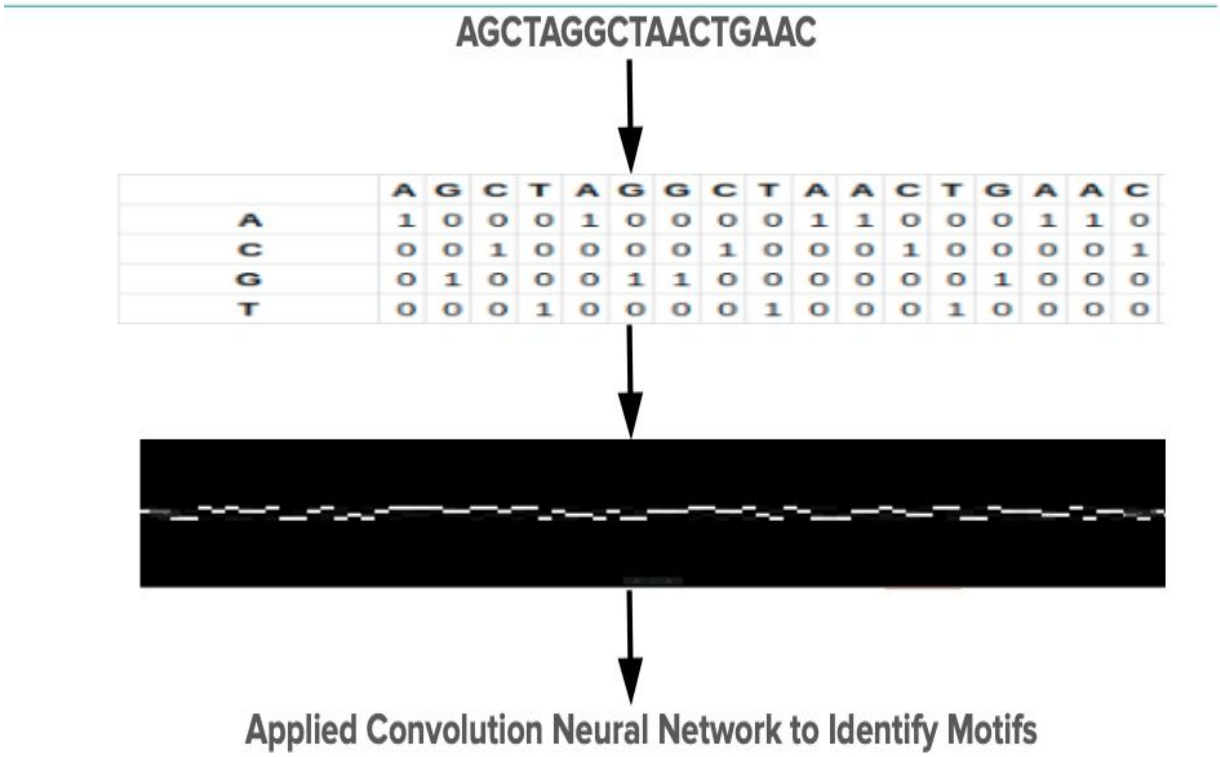
Example: .....AGCTAGGCTAACTGAA.....

	A	G	C	T	A	G	G	C	T	A	A	C	T	G	A	A	C
A	1	0	0	0	1	0	0	0	0	1	1	0	0	0	1	1	0
C	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1
G	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0
T	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0

Transcription factor binding data converted as binary array

TF5	TF6	TF828	TF857
1	0	1	1

Project Workflow



Code for Steps

Loading the Data:

```
import h5py
import numpy as np
fileLoc = 'train.mat'
file = h5py.File(fileLoc)
arrays = {}
for k,v in file.items():
  arrays[k]=np.array(v)
Xcsvtemp = arrays['trainxdata']
Ycsvtemp = arrays['traindata']
```

Taking 5000 Samples:

```
Xcsv = []
for i in range(5000):
  Temp1=[]
  for j in range(4):
    Temp2=[]
    for k in range(1000):
      Temp2.append(Xcsvtemp[k][j][i])
    Temp1.append(Temp2)
  Xcsv.append(Temp1)

Ycsv = []
for i in range(5000):
  Temp1=[]
  for j in range(919):
    Temp1.append(Ycsvtemp[j][i])
  Ycsv.append(Temp1)

import csv
with open('X_5k.csv','w',newline='') as file:
  w = csv.writer(file,delimiter=',')
  w.writerows(Xcsv)

YcsvNumpy = np.array(Ycsv)
np.savetxt('Y_5k.csv',YcsvNumpy,delimiter=',')
```

Converting 5000 Samples to Matrix:

```
import pandas as pd
Xtemp = pd.read_csv('X_5k.csv',header=None)

XNew = []
for i in range(5000):
    temp1=[]
    for j in range(4):
        temp2=[]
        for k in range(3000):
            if(k%3==1):
                temp2.append(int(Xtemp.iloc[i][j][k]))
        temp1.append(temp2)
    XNew.append(temp1)

YNew = pd.read_csv('Y_5k.csv',header=None)
```

Converting Matrix to Images:

```
from matplotlib.pyplot import imshow
import numpy as np
from PIL import Image

for i in range(5000):
  img = Image.fromarray(np.uint8(XNew[i] * 255) , 'L')
  img.save('Im'+str(i)+'.jpg')
```

Convolution Neural Network Model:

```
cnn_model_gen =  tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (2, 2), activation='relu', input_shape=(4,1000, 1)),
    tf.keras.layers.MaxPooling2D((2,2)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(14, activation='softmax')
])
```

Convolution Model Summary:

```
cnn_model_gen.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 3, 999, 32)        160
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 1, 499, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 15968)             0
_________________________________________________________________
dropout (Dropout)            (None, 15968)             0
_________________________________________________________________
dense (Dense)                (None, 512)               8176128
_________________________________________________________________
dense_1 (Dense)              (None, 14)                7182
=================================================================
Total params: 8,183,470
Trainable params: 8,183,470
Non-trainable params: 0
_________________________________________________________________
```

Results of TFs 857,828,5,6

TF 857

1000 Samples	Actual Positive	Actual Negative
Predicted Positive	10	10
Predicted Negative	153	227

3000 Samples	Actual Positive	Actual Negative
Predicted Positive	156	226
Predicted Negative	146	672

5000 Samples	Actual Positive	Actual Negative
Predicted Positive	6	13
Predicted Negative	415	1566

TF 828

1000 Samples	Actual Positive	Actual Negative
Predicted Positive	5	8
Predicted Negative	131	256

3000 Samples	Actual Positive	Actual Negative
Predicted Positive	41	140
Predicted Negative	128	891

5000 Samples	Actual Positive	Actual Negative

Predicted Positive	20	60
Predicted Negative	246	1674

TF 6

1000 Samples	Actual Positive	Actual Negative
Predicted Positive	0	0
Predicted Negative	24	376

3000 Samples	Actual Positive	Actual Negative
Predicted Positive	1	2
Predicted Negative	86	1111

5000 Samples	Actual Positive	Actual Negative
Predicted Positive	1	8
Predicted Negative	130	1861

TF 5

1000 Samples	Actual Positive	Actual Negative
Predicted Positive	0	0
Predicted Negative	40	360

3000 Samples	Actual Positive	Actual Negative
Predicted Positive	27	109
Predicted Negative	116	948

5000 Samples	Actual Positive	Actual Negative
Predicted Positive	6	37
Predicted Negative	231	1726

ROC Curve:

(TPR vs FPR)

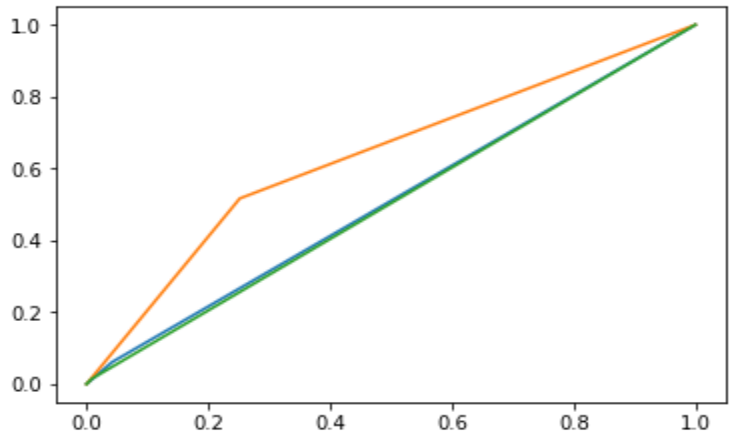
TPR = TP/(TP+FN) , FPR = FP/(FP+TN)

➜ 1000 Samples

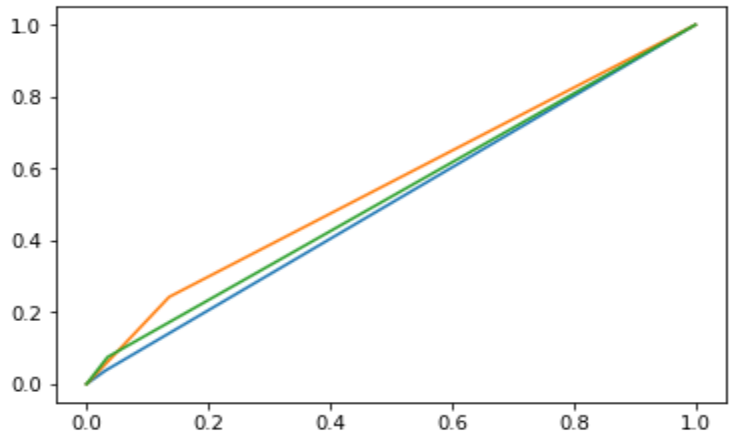
➜ 3000 Samples

➜ 5000 Samples

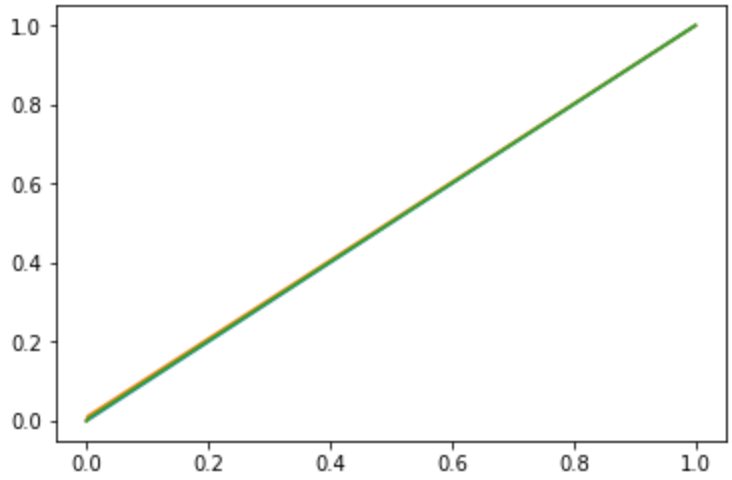
TF 857



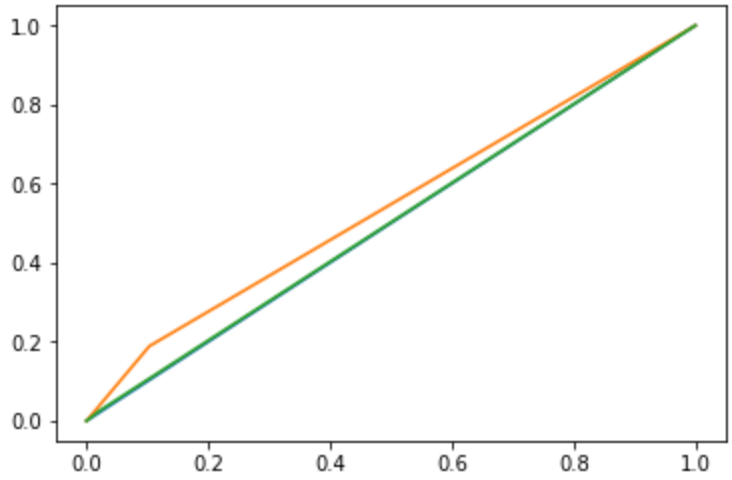
TF 828



TF 6



TF 5



Discussion

Further implementation of this algorithm along with a function which can capture the interaction of transcription factors, DNase, histone marks along with the chromatin could reveal the nature of transcriptional regulation of genes including across non-coding regions of the genome.

Applications

- Predict new motifs
- High sensitivity and specificity of the algorithm could be used to detect the effect of SNPs/mutations on motifs.

- Detection of Quantitative Trait Loci (QTL).

References

- Quang, D. and Xie, X. ``DanQ: a hybrid convolutional and recurrent neural network for predicting the function of DNA sequences'', NAR, 2015
- Zhou J., Troyanskaya O.G. (2015) Predicting effects of noncoding variants with deep learning-based sequence model. Nat. Methods, 12, 931–934