Chromatin 3D Structure and Cancer Typing via Deep Learning

石毅 (Shi, Yi)

2017.06.21

Center for Systems Biomedicine
Shanghai Jiao Tong University

USyd-SJTU Joint Research Alliance for Translational Medicine
Outline

- Chromatin 3D Structure
- DNN-based Cancer Typing
- Discussion
Chromatin 3D Structure
Chromatin 3D Structure

- Human chromatin from a single cell if unpacked and chained up: \(~2\) meters long
- Human nucleus: micron meter \((10^{-6} \text{ m})\) scale in diameter

Chromatin structure illustration
Picture from users.rcn.com
Chromatin 3D Structure

From DNA to chromosome

DNA → beads on the string, 10nm → 30 nm fiber

interphase

during meiosis or mitosis

Pictures from Wikipedia.com
Chromatin conformation capture technology (HiC)

- HiC provides genome-wise all-to-all chromatin contact profiling compared to the previous FISH (optical one-to-one), 3C (one-to-one) and 4C (one-to-all), and ChIA-PET (targeted all-to-all).

Insights:
- Chromatin territories exist.
- Genome partitioned into 2 compartments, active and inactive, with high intra-compartment interaction and low inter-compartment interaction.
- The 2 compartment partitioning is correlated to epigenetic signals.
- There are more genes in active compartment and those genes are more active.

Chromatin 3D Structure

In Situ HiC

- Higher resolution, better insights.

Insights:
- Six types of chromatins discovered.
- More certain about detailed looping.
- TAD and replication origin which is correlated to cancerous mutations.

Picture from Rao et al. Cell, 2014
Chromatin 3D Structure

Topological domain (TD) identification

The TD boundary presents a local minimum along the binSignal.

binSignal(i) is the average contact frequency in the diamond area.

Curving Fitting Algorithm:
1. $P_{start} =$ signal start, $P_{end} =$ signal end;
2. $F_N = 0, F_I = 0$;
3. $P_0 = P_{start}$;
4. Do while $P_{start} <= P_{end}$ and $P_I <= P_{end}$;
5. // line($P_s, P_e$) = a line connecting two points $P_s$ and $P_e$;
6. $L_s =$ length of line($P_{start}, P_s$);
7. $E_s =$ sum of distance error ($P_s$, line($P_{start}, P_s$));
8. where $P_i$ are any points between $P_{start}$ and $P_e$;
9. $F = L_s - E_s$;
10. if ($F_e < F_s$) set $P_e$ as turning point;
11. $P_{end} = P_{start};$ $P_0 = P_{start}$;
12. else $F_e = F_s; P_0 = P_{start}$;
13. endif
14. loop

Shin & Shi. et al. NAR, 2015
**Chromatin 3D Structure**

- **Structure Modeling**
  - Generating reasonable structure decoys

- **Structure based studies**
  - Structure clustering to find cell states
  - Radius position and feature association
  - Proximity and feature association

*Kalhor et al. Nature Biotech. 2011*
## Chromatin 3D Structure

### Chromatin features and radius position

<table>
<thead>
<tr>
<th>17 Features</th>
<th>66 Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeneDensi, GeneExpre, EarlyRep, lincRNA, Dnase, Pol2, Ctcf, H3k4me1, H3k4me3, H3k27ac, H3k4me2, H3k79me2, H3k9ac, H3k9me3, H4k20me1, H3k27me3, H2az</td>
<td>GeneCount, GeneDensi, GeneExpre, EarlyRep, lincRNA, Dnase, Pol2, Ctcf, H3k4me1, H3k4me3, H3k27ac, H3k4me2, H3k79me2, H3k9ac, H3k9me3, H4k20me1, H3k27me3, H2az, Bhlhe40c, Brca1, Cdp, Cfos, Chad, Chad2, Corest, Ctcf, E2f4, Ebf1, Elk1, Erra, Gcn5, Ikzf1, Input, Irf3, Jund, Mafk, Max, Maz, Mxi1, Nfe2, Nfkb, Nfya, Nfyc, Nrl, P300b, Pol2, Pol3, Rad21, Rfx5, Sin3, Smc3, Spt20, Srebp1, Srebp2, Stat1, Stat3, Tblr1, Tbp, Tr4, Usf2, Whip, Yy1, Znf143, Znf274, Znf384, Zzz3</td>
</tr>
</tbody>
</table>

**Red:** Histone Modification Markers  
**Green:** TFs  
**Blue:** Others
Chromatin 3D Structure

Chromatin features and radius position

(a) Correlation Coefficient
(b) LASSO Weight
Chromatin 3D Structure

Cancerous translocation and chromatin structure

1 Map Translocation Datasets to Genome Coordinates
   Translocations
   t(12;19)(q13;p13)
   t(16;20)(p22;q31)
   ...

2 Calculate Proximity Scores
   Translocation Partners
   (high proximity)
   Nucleus
   Random Regions
   (low proximity)

3 Assess Significance by Permutation Testing
   - Compare each set of translocations to permuted sets
   - Compare each individual translocation to permuted regions

Chromatin 3D Structure

Somatic Co-mutation Hotspots

Shi. et al. Scientific Reports, 2016
• CTCF enriched in “hotspots”

Shi. et al. Scientific Reports, 2016
• Similar mutation type and flanking sequence conservation in “hotspots”

• Pathway enrichment

DNN-based Cancer Typing
Traditional cancer diagnosis

- Morphological appearance:
  - Pathological section (golden standard)
  - Imaging techniques

- Gene or protein expression

Images from:
- baidu.com
- radiology.med.nyu.edu
- well.ox.ac.uk
- sigmaaldrich.com
Inside drives

- Somatic point mutations
- Insertions and deletions (INDELs)
- Chromatin translocations
- Copy number abnormalities
DNN-based Cancer Typing

Applications of deep neural network (DNN) learning
DeepGene: an advanced cancer type classifier based on deep learning and somatic point mutations

Yuchen Yuan, Yi Shi, Changyang Li, Jinman Kim, Weidong Cai, Zeguang Han and David Dagan Feng

From The 27th International Conference on Genome Informatics
Shanghai, China: 3-5 October 2016
DNN-based Cancer Typing

Why CNA:

• Links between aneuploidy and cancer have long been recognized.
• CNA is the major form of chromosomal instability, affecting a larger fraction of the genome in cancers.
• The technologies of profiling genome-wide CNV is more developed than before, from DNA microarray based to whole-genome DNA sequencing based to exome sequencing based.

Picture from biometrics.cse.msu.edu
Data preprocessing

- The CNA data is first empirically clipped into the interval [0, 10].
- The clipped data is then zero-padded at tail to have the desired length that fits the input of the subsequent neural networks.
- For 2D CNN, the CNA samples are then reshaped into 176*176*1, just like single-layered images.
# DNN-based Cancer Typing

## 1D CNN

Table 1. Architecture of our proposed 1D CNN.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Type</th>
<th>Output size</th>
<th>Conv (size, channel, pad)</th>
<th>Max pooling</th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
<td>in</td>
<td>32768<em>1</em>ch</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>conv1</td>
<td>c+r+p</td>
<td>8192<em>1</em>32</td>
<td>3*1, 32, 1</td>
<td>4*1</td>
</tr>
<tr>
<td>conv2</td>
<td>c+r+p</td>
<td>2048<em>1</em>64</td>
<td>3*1, 64, 1</td>
<td>4*1</td>
</tr>
<tr>
<td>conv3</td>
<td>c+r+p</td>
<td>512<em>1</em>128</td>
<td>3*1, 128, 1</td>
<td>4*1</td>
</tr>
<tr>
<td>conv4</td>
<td>c+r+p</td>
<td>128<em>1</em>256</td>
<td>3*1, 256, 1</td>
<td>4*1</td>
</tr>
<tr>
<td>conv5</td>
<td>c+r+p</td>
<td>32<em>1</em>512</td>
<td>3*1, 512, 1</td>
<td>4*1</td>
</tr>
<tr>
<td>conv6</td>
<td>c+r</td>
<td>1<em>1</em>4096</td>
<td>32*1, 4096, 0</td>
<td>N/A</td>
</tr>
<tr>
<td>fc7</td>
<td>fc+r+d</td>
<td>1<em>1</em>4096</td>
<td>1*1, 4096, 0</td>
<td>N/A</td>
</tr>
<tr>
<td>fc8</td>
<td>fc</td>
<td>1<em>1</em>25</td>
<td>1*1, 25, 0</td>
<td>N/A</td>
</tr>
<tr>
<td>loss</td>
<td>sm+log</td>
<td>1*1</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Annotations - in: input layer; c: convolutional layer; r: ReLU layer; p: pooling layer; fc: fully connected layer; d: dropout layer; sm: softmax layer; log: log loss layer; ch: number of input channels (depending on whether the HiC data is used).
## 2D CNN

<table>
<thead>
<tr>
<th>Layer</th>
<th>Type</th>
<th>Output size</th>
<th>Conv (size, channel, pad)</th>
<th>Max pooling</th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
<td>in</td>
<td>176<em>176</em>ch</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>conv1</td>
<td>c+r+p</td>
<td>88<em>88</em>32</td>
<td>3*3, 32, 1, 1</td>
<td>2*2</td>
</tr>
<tr>
<td>conv2</td>
<td>c+r+p</td>
<td>44<em>44</em>64</td>
<td>3*3, 64, 1, 1</td>
<td>2*2</td>
</tr>
<tr>
<td>conv3</td>
<td>c+r+p</td>
<td>22<em>22</em>128</td>
<td>3*3, 128, 1, 1</td>
<td>2*2</td>
</tr>
<tr>
<td>conv4</td>
<td>c+r+p</td>
<td>11<em>11</em>256</td>
<td>3*3, 256, 1, 1</td>
<td>2*2</td>
</tr>
<tr>
<td>conv5</td>
<td>c+r</td>
<td>1<em>1</em>1024</td>
<td>11*11, 1024, 0</td>
<td>N/A</td>
</tr>
<tr>
<td>fc6</td>
<td>fc+r+d</td>
<td>1<em>1</em>1024</td>
<td>1*1, 1024, 0</td>
<td>N/A</td>
</tr>
<tr>
<td>fc7</td>
<td>fc</td>
<td>1<em>1</em>25</td>
<td>1*1, 25, 0</td>
<td>N/A</td>
</tr>
<tr>
<td>loss</td>
<td>sm+log</td>
<td>1*1</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Annotations** - in: input layer; c: convolutional layer; r: ReLU layer; p: pooling layer; fc: fully connected layer; d: dropout layer; sm: softmax layer; log: log loss layer; ch: number of input channels (depending on whether the HiC data is used).
Implementation details

- Both the 1D CNN and the 2D CNN are implemented in Python under the Caffe framework, which is an open source framework for CNN training and testing.
- The machine used for our experiments is a PC with Intel 6-Core i7-5820K 3.3GHz CPU, 64GB RAM, GeForce GTX TITAN X 12GB GPU, and 64-bit Ubuntu 14.04.3 LTS.
- Software dependencies include CUDA 8.0 and cuDNN 5.1.
DNN-based Cancer Typing

Proposed method in different design options

**Fig. 1.** Performances of our proposed method with different design options. (a) With different HiC data configurations. From left to right: baseline model (2D CNN); baseline with hiESC only; baseline with IMR90 only; baseline with both types of HiC data. The last configuration leads to the optimal performance. (b) With different network and HiC combinations. From left to right: 1D CNN without HiC data; 1D CNN with HiC data; 2D CNN without HiC data; 2D CNN with HiC data. The last configuration leads to the optimal performance.
DNN-based Cancer Typing

Other classifiers

Table 3. Evaluation of SVM with different kernel types.

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Linear</th>
<th>Polynomial</th>
<th>RBF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.317</td>
<td>0.322</td>
<td>0.275</td>
</tr>
</tbody>
</table>

Table 4. Evaluation of KNN with different number of neighbors and p value.

<table>
<thead>
<tr>
<th>p \ n_neighbors</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.257</td>
<td>0.259</td>
<td>0.262</td>
<td>0.265</td>
<td>0.266</td>
</tr>
<tr>
<td>2</td>
<td>0.263</td>
<td>0.273</td>
<td>0.283</td>
<td>0.279</td>
<td>0.277</td>
</tr>
<tr>
<td>3</td>
<td>0.254</td>
<td>0.259</td>
<td>0.264</td>
<td>0.258</td>
<td>0.262</td>
</tr>
</tbody>
</table>

Table 5. Evaluation of NB with different data distribution assumptions

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Bernoulli</th>
<th>Multinomial</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>0.161</td>
<td>0.238</td>
<td>0.139</td>
</tr>
</tbody>
</table>
DNN-based Cancer Typing

Comparing with other classifiers

Fig. 2. Performances of our proposed method against three widely adopted data classifiers. (a) The comparison methods use raw CNA input data (without HiC). From left to right: Our method, SVM (polynomial kernel), KNN (number of neighbors = 5 and p = 2) and NB (multinomial distribution). Our method shows significant advantage against the comparison methods. (b) The comparison methods use both CNA and HiC as input data. From left to right: Our method, SVM (polynomial kernel), KNN (number of neighbors = 5 and p = 2) and NB (multinomial distribution). Our method shows even greater advantage against the comparison methods.
Discussion

Further investigation

- Integrating heterogeneous mutation data together, e.g. SNV, INDEL, CNV, translocation
- What feature (gene) combinations contribute to better prediction accuracy? Why?

How this can help real diagnosis?

- Applying to CTC or ctDNA for early diagnosis, subtyping, locating.
Acknowledgement

Prof. Ze-guang Han
Prof. David Feng
Dr. Yuchen Yuan & I
Prof. Tom Cai

THE UNIVERSITY OF SYDNEY
USyd-SJTU Joint Research Alliance for Translational Medicine

NSFC
National Natural Science Foundation of China

浦江人才计划
Pujiang Scholar
Questions & Comments?
고맙습니다!
ありがとう!
谢谢!
Thank you!
2017.06.21