An Image Representation Based Convolutional Network for DNA Classification

ICLR 2018

Presenter: Joung Min Choi

Background

- **Chromatin**
  - The folding structure of the DNA molecule combined with the helper molecules (e.g. Histone proteins)
  - The spatial configuration defines the functional properties of DNA
  - Can assume several function-defining *epigenetic states*

- **Key determinant of chromatin state**
  - Underlying primary DNA sequence
    - Sequence patterns: Responsible for recruiting histone proteins and their chemical modifications

- **Utmost interest for predicting chromatin related states from primary DNA sequences**
  - Methods based on machine learning and deep neural networks

Problem

- **Treating DNA sequence data as a sequence**
  - Neglects its inherent and biologically relevant spatial configuration and the resulting interaction between distal sequence elements
  - Spatial configuration of DNA suggests the relevance of a higher-dimensional spatial representation of DNA

- **Lack of comprehensive understanding for the structure of the chromatin**
  - Suggestions for higher-dimensional representations of DNA do not exist

Proposed approach

- **HCNN**
  - A convolutional neural network that takes an image-representation of primary DNA sequence as its input, and predicts chromatin-related states

1) Use **space-filling curves** for DNA representation by mapping DNA sequences to higher-dimensional images
2) Predict chromatin states using a CNN designed for detecting distal relations

Related work

1) DNA sequence classification

: The task of determining whether a sequence S belongs to an existing class C

- **Pahm et al. (2005) and Higashihara et al.(2008)**
  - Support vector machines to predict chromatin state from DNA sequence features
- **Nguyen et al. (2016)**
  - CNN-based model (CNN+FC layer) using the sequential form of DNA sequence as input

2) DNA sequence transformation into image using Hilbert curves

- **Anders (2009)**
  - Demonstrating the power of Hilbert curves for visualizing DNA
- **Elgin (2012)**
  - Results indicated that when arranging DNA sequences based on Hilbert curves, contiguous areas belonging to identical chromatin states cover rectangular areas

Methods

1. DNA sequence Representation

1) Represent a sequence as a list of k-mers

- Sequence's k-mers: k-letter words from the alphabet {A,C,G,T} that together make up the sequence

  ex) TGACGAC: the list of 3-mers {TGA, GAC, ACG, CGA, GAC}

- Previous work: 3-mers and 4-mers are useful
- Preliminary experiments: $k = 4$ yields the best performance

2) Transform each k-mer into a one-hot vector

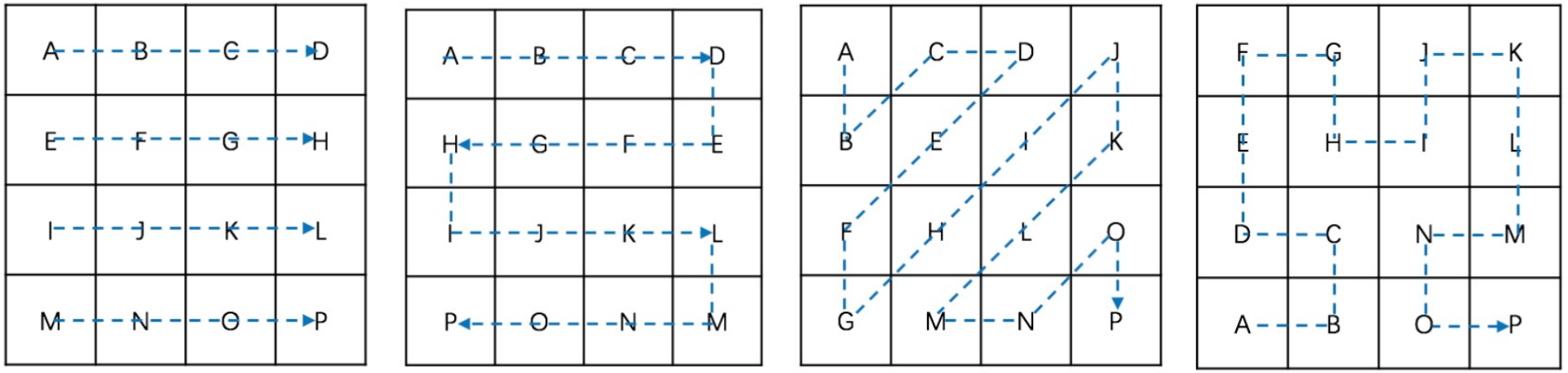
- A vector of length 4^k is needed to represent all k-mers in a DNA sequence
- DNA sequence as a list of 4-mers: a list of one-hot vector of length 256

Methods

3) Transform the list of one-hot vectors int an image

- Assign a one-hot vector of length 256 to each pixel using space-filling curves

- **Space filling curves**
  - Map 1D sequences to a 2D surface preserving continuity of the sequence



(a) Reshape curve (b) snake curve (c) Diag-Snake curve (d) Hilbert curve

Figure 5: Space-filling curves

Methods

- **Hilbert curve**
  - Recursively the curve is divided into four parts, which are mapped to the four quadrants of a square
  - Results: a square image of size $2^n \times 2^n$ (n: the order of the curve)

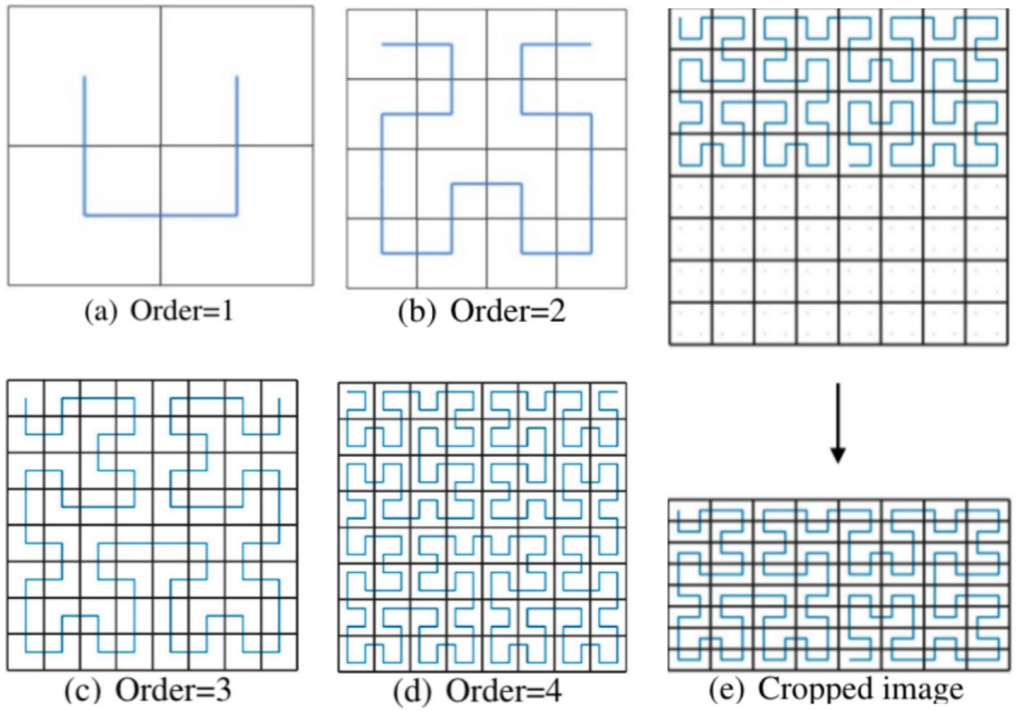
① **Choose n such that $2^n \times 2^n$ is at least the number of k-mers in the sequence to fit all k-mers into the image**

② **Crop the picture by removing the unused part of the image**

- Sequence with length of 500 bp: 497 4-mers

- Need a Hilbert curve of order 5
  - An image of dimensions $2^5 \times 2^5 \times 256$
  - Almost half of the 1024 pixels are filled and other empty
  - Remove the empty half of the image

  ⇒ Results: an image of size $16 \times 32 \times 256$



Methods

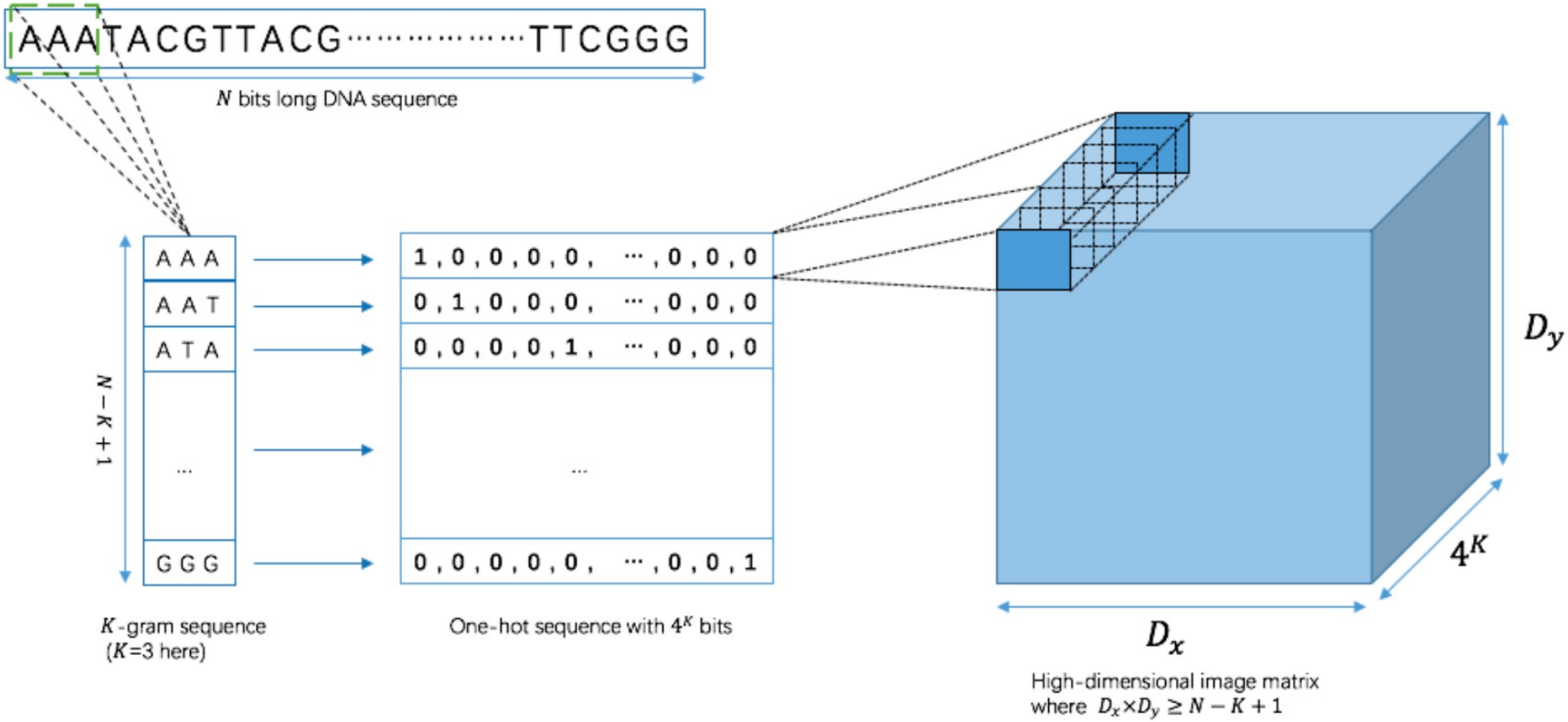


Figure 6: Sequence to Image

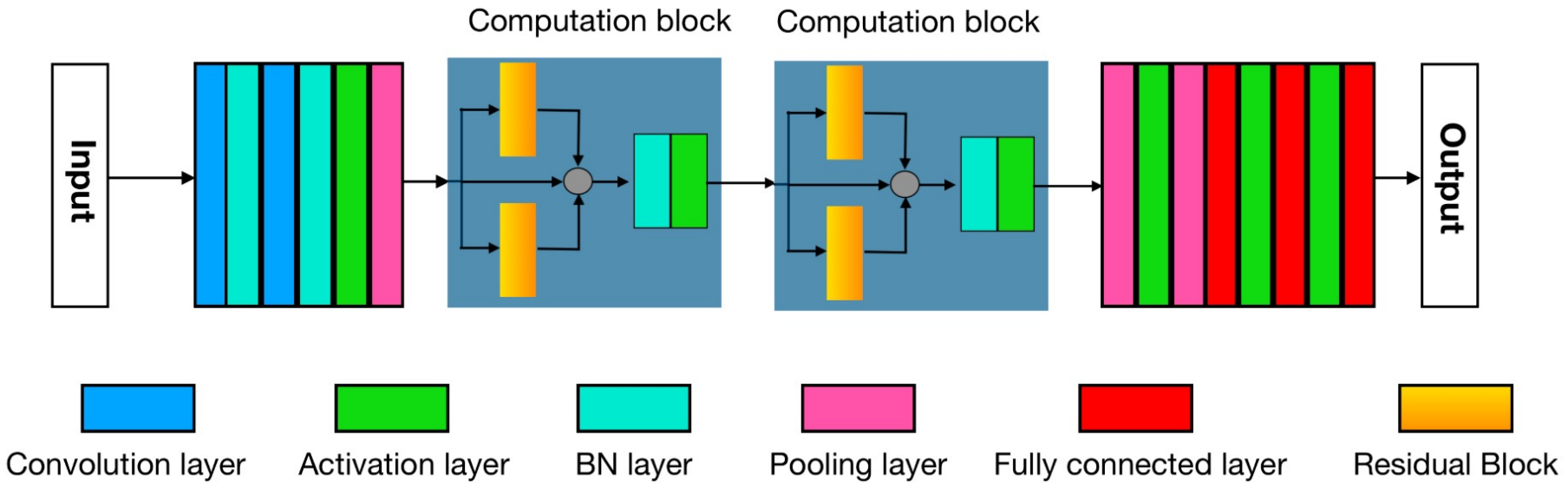
Methods

2. Network architecture

- Each pixel in the generated image: A one-hot vector representing k-mer
  - $k = 4$: Image of 256 channels (Overfitting)
  - Each channel contains very sparse information

- Design a CNN for high dimensional image inspired by *ResNet* and *Inception*

  1) First part: To reduce the sparseness of the input image and capture long range features with large filters

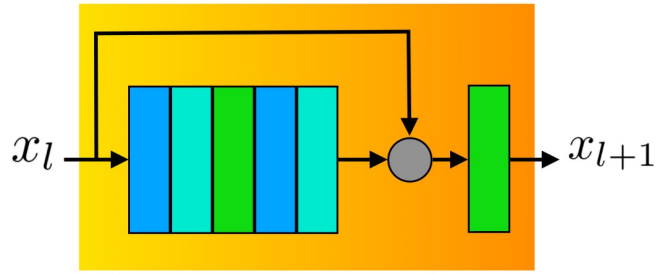


Methods

2) Computation Block

- The outputs of two Residual blocks and one identity mapping are summed
- Residual blocks

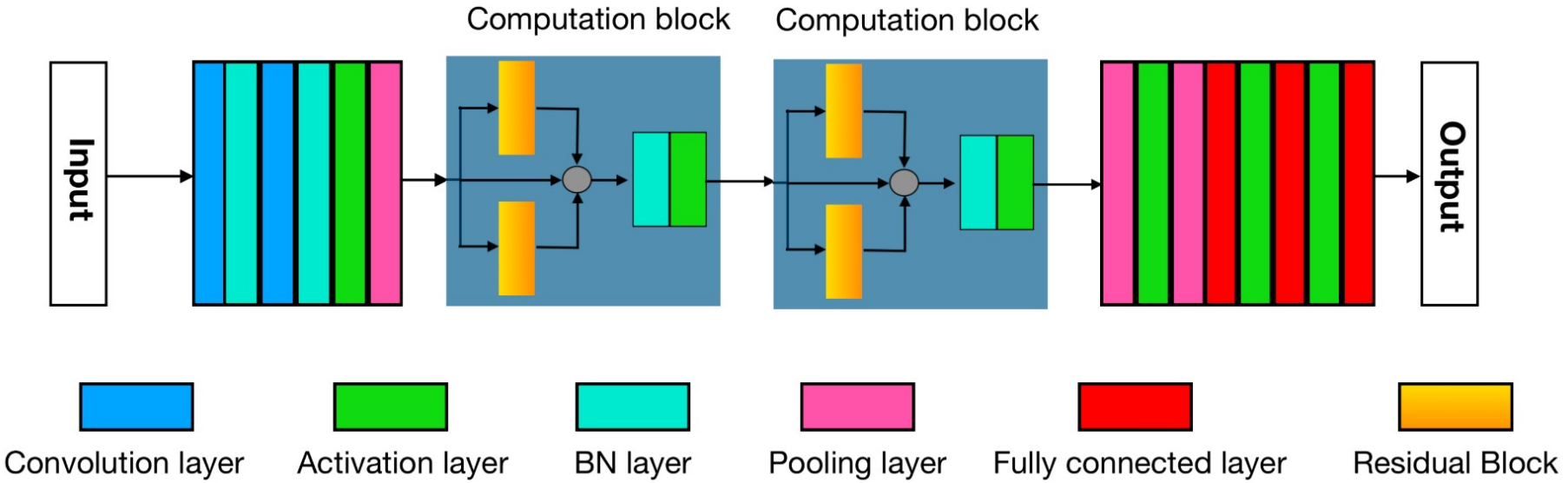
  : Concatenation of the output from five layers with two convolutions and the input



Residual block

3) Last part

- To obtain the output classification label



Experiments

- **Datasets**

  1) Ten publicly available datasets from Pokholok et al. (2005)

  - DNA sequences with a length of 500 base pairs
  - Each sequence is labeled either as “positive” or “negative”, indicating whether the subsequence contains regions that are wrapped around a histone protein
  - Randomly chosen 90% of the dataset: Training, 5%: validation, 5%: Evaluation

  2) Splice-junction genes sequences dataset from Lichman (2013)

  - DNA subsequence of length 61
  - Each subsequence known to be ① an intron-to-exon splice-junction, ② an exon-to-intron splice junction or ③ neither
  - Using 1-mers as the dataset is relatively small

Name	♯Samples	Description
H3	14965	H3 occupancy
H4	14601	H4 occupancy
H3K9ac	27782	H3K9 acetylation relative to H3
H3K14ac	33048	H3K14 acetylation relative to H3
H4ac	34095	H4 acetylation relative to H3
H3K4me1	31677	H3K4 monomethy-lation relative to H3
H3K4me2	30683	H3K4 dimethylation relative to H3
H3K4me3	36799	H3K4 trimethylation relative to H3
H3K36me3	34880	H3K36 trimethylation relative to H3
H3K79me3	28837	H3K79 trimethylation relative to H3
Splice	3190	Splice-junction Gene Sequences

Experiments

- **Competing methods**

1) Support vector machine by Higashihara et al. (2008)

2) Seq-CNN by Nguyen et al. (2016)

3) LSTM using 4-mer profile of the sequence as input
   - Including only the 100 most frequent 4-mers as 256 4-mers showed overfitting in the preliminary test

4) Seq-HCNN
   - Flattened version of HCNN without space-filling curves using 49×1 convolution filter in the 1D-sequence model

Results: Prediction performance comparison for each dataset

Table 3: Prediction accuracy obtained with an SVM-based method, Seq-CNN from Nguyen et al. (2016), LSTM, seq-HCNN and HCNN. The results for SVM are taken from Table 12 in Higashihara et al. (2008). In the splice dataset, Seq-CNN performed best when using 4-mers, while for HCNN and seq-HCNN 1-mers yielded the best performance.

Dataset	SVM	LSTM	Seq-CNN	seq-HCNN	HCNN
H3	86.47%	64.13%	79.25%	86.86 ± 1.563%	**87.34 ±0.263%**
H4	87.82%	63.82%	81.86%	87.31 ± 0.952%	**87.33±0.264%**
H3K9ac	75.08%	63.07%	68.76%	78.47 ± 0.699%	**79.19±0.239%**
H3K14ac	73.28%	68.31%	68.31%	**75.06** ±0.987%	74.79±**0.226%**
H4ac	72.06%	60.63%	64.80%	77.04 ± 1.256%	**77.06±0.233%**
H3K4me1	69.71%	60.43%	62.60%	**73.47** ±0.789%	73.21±**0.221%**
H3K4me2	68.97%	61.45%	62.38%	73.91 ± 0.631%	**74.27±0.224%**
H3K4me3	68.57%	58.03%	62.33%	**74.54** ±0.865%	74.45±**0.225%**
H3K36me1	75.19%	60.78%	72.20%	**77.18** ±0.973%	77.03±**0.232%**
H3K79me1	80.58%	63.84%	75.07%	**81.66** ±1.264%	81.63±**0.246%**
Splice	94.70%	96.23%	91.82%	93.21 ± 1.645%	**94.11±0.284%**

Table 5: Recall, Precision, area under precision-recall curve (AP) and area under ROC curve (AUC) for seq-HCNN and HCNN. The reported values are the means over ten folds.

Dataset	Recall		Precision		AP		AUC	
	seq-HCNN	HCNN	seq-HCNN	HCNN	seq-HCNN	HCNN	seq-HCNN	HCNN
H3	85.67%	**87.33%**	85.67%	**87.33%**	90.33%	**93.33%**	91.00%	**93.67%**
H4	87.00%	**87.33%**	87.00%	**87.00%**	92.67%	**94.67%**	93.67%	**94.67%**
H3K9ac	78.33%	**79.00%**	78.67%	**79.00%**	78.33%	**85.00%**	79.67%	**85.33%**
H3K14ac	**74.00%**	73.67%	74.67%	**75.00%**	73.67%	**79.67%**	76.33%	**81.33%**
H4ac	76.67%	**77.67%**	77.33%	**78.33%**	78.67%	**82.67%**	80.33%	**83.33%**
H3K4me1	72.33%	**73.00%**	72.67%	**73.67%**	70.67%	**76.33%**	71.67%	**78.33%**
H3K4me2	70.67%	**72.33%**	73.00%	**74.00%**	69.33%	**77.33%**	70.00%	**78.67%**
H3K4me3	74.33%	**74.67%**	**75.00%**	74.67%	71.00%	**78.67%**	72.00%	**80.00%**
H3K36me3	76.00%	**76.67%**	77.00%	**77.67%**	76.33%	**82.00%**	79.33%	**83.00%**
H3K79me3	81.00%	**82.33%**	81.00%	**82.67%**	79.67%	**88.00%**	81.00%	**88.67%**
Splice	91.00%	**95.00%**	90.67%	**94.33%**	95.00%	**97.67%**	97.33%	**98.67%**

Results: Performance evaluation for training time and prediction accuracy with different mapping methods

Table 4: Training times, presented as min:sec.

Dataset	LSTM	seq-CNN	seq-HCNN	HCNN
H3	35:43	95:23	6:47	3:40
H4	45:32	95:53	5:12	3:12
H3K9ac	76:06	173:18	17:24	7:40
H3K14ac	81:21	180:56	17:42	13:24
H4ac	93:32	181:33	24:48	17:32
H3K4me1	93:44	192:20	18:30	10:38
H3K4me2	94:22	188:13	18:23	14:38
H3K4me3	96:03	162:32	20:40	11:33
H3K36me3	93:48	161:12	21:52	16:37
H3K79me3	64:28	158:34	14:25	10:13
Splice	6:42	35:12	3:42	1:30

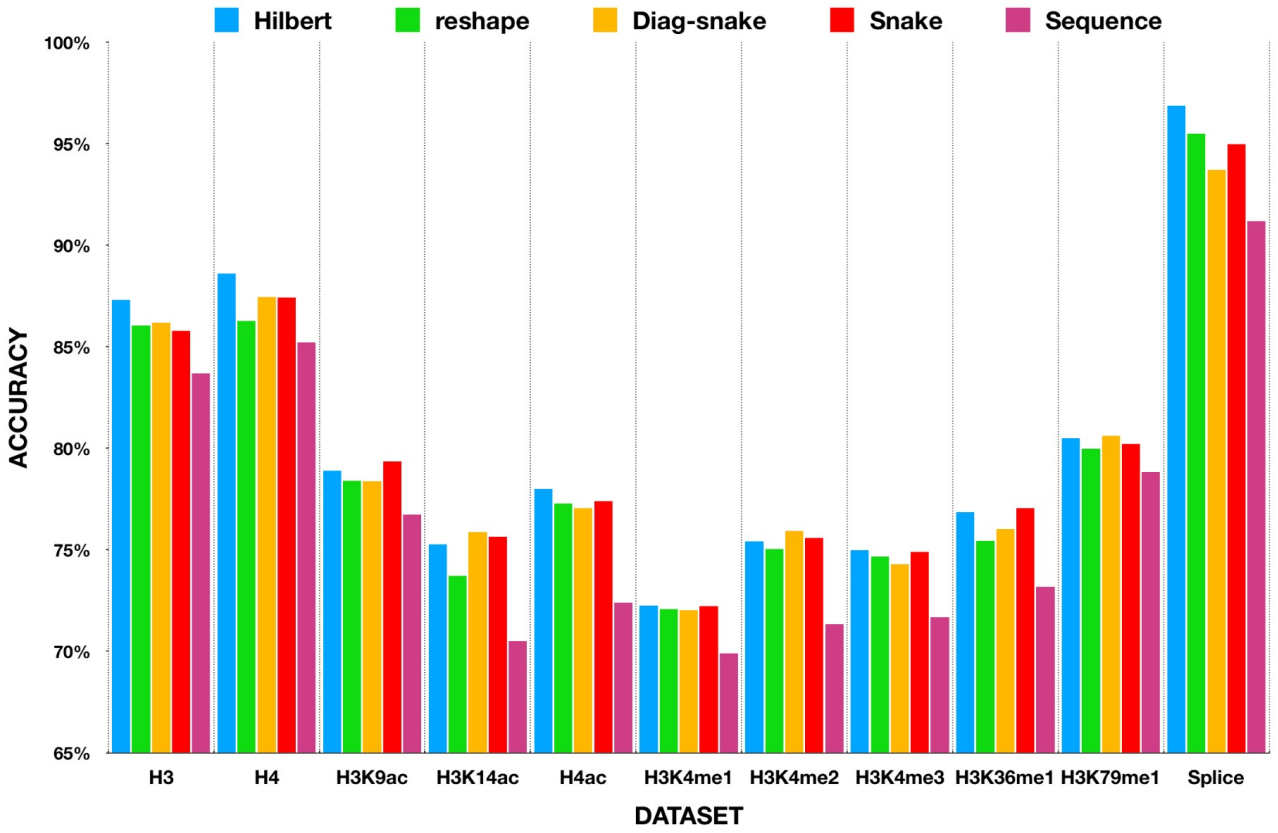


Figure 4: HCNN with different mapping strategies

Discussion

- **Factors for improvement over the existing CNN by Nguyen et al.(2016)**
  - Larger convolutional filters allowing the model to detect long-distance interactions
  - Small number of parameters allowing for faster optimization

    : Due to the size of the layer preceding the fully connected layer, which is larger in the existing model
  - Use of a 2D input which enhances the model's capabilities of incorporating long-term interactions

- **Limitation**
  - Fixed length in Hilbert curves

    : The generated images contain some empty spaces, consuming computation resources

My thoughts

- Hilbert curves does not leverage any biological input (no biological meaning)

  : No difference between randomly putting the sequence and Hilbert curves
- Authors mentioned that treating the sequence as just a sequence neglects its inherent and biological relevant spatial configuration, but usage of Hilbert curves could rather make the model to learn wrong local features
- Too many empty spaces and channels
- Hard to interpret the model by transforming 1D to 2D image

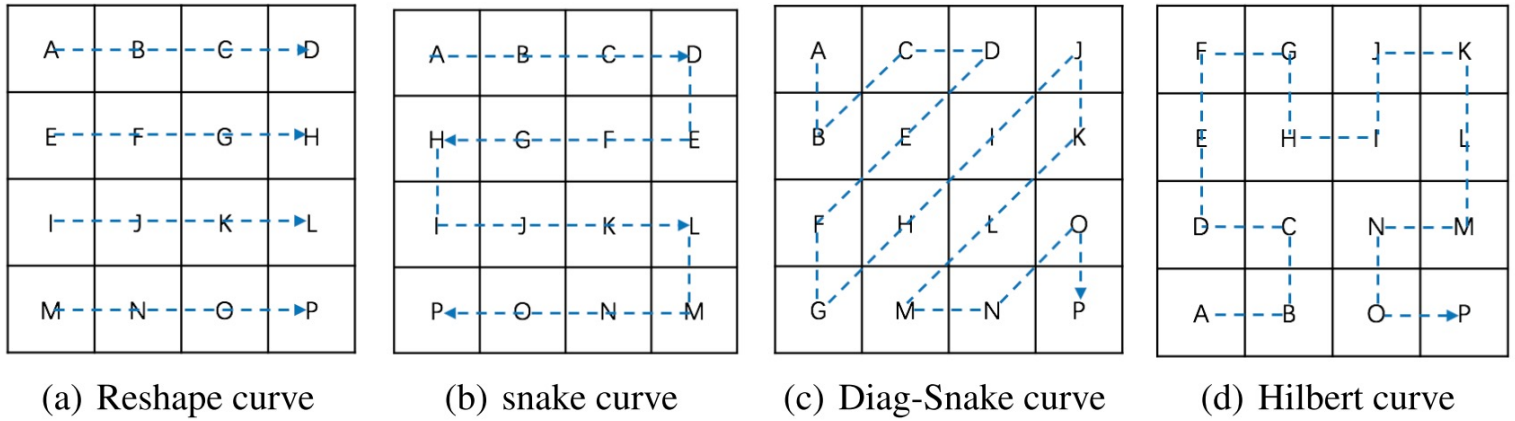


Figure 5: Space-filling curves