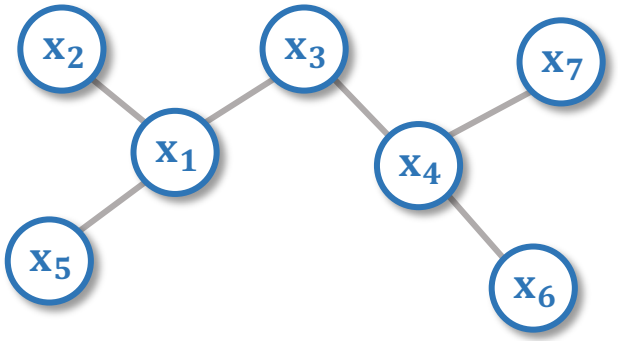
Chromatin interaction aware gene regulatory modeling with graph attention networks

Ping-Han Hsieh and Vipin Kumar

2021 June 16th

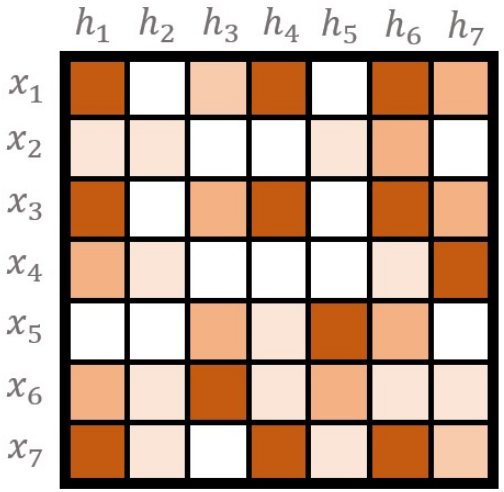
Introduction

What is Graph Data Structure



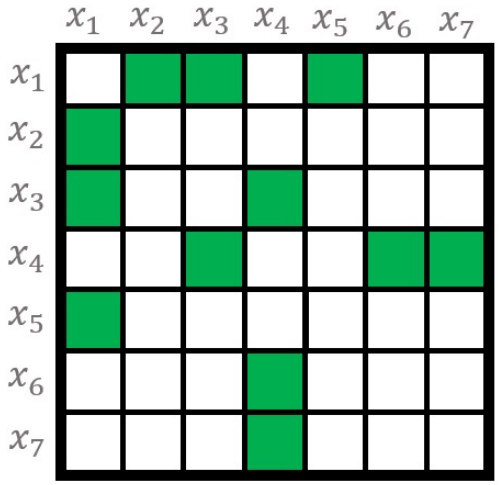
input graph

$G = (V, E)$



vertices feature

$\mathbf{H} \in \mathbb{R}^{N \times F}$



adjacency matrix

$\mathbf{A} \in \mathbb{R}^{N \times N}$

Graph Datatype

- homogenous graph
- heterogenous graph

Graph Property

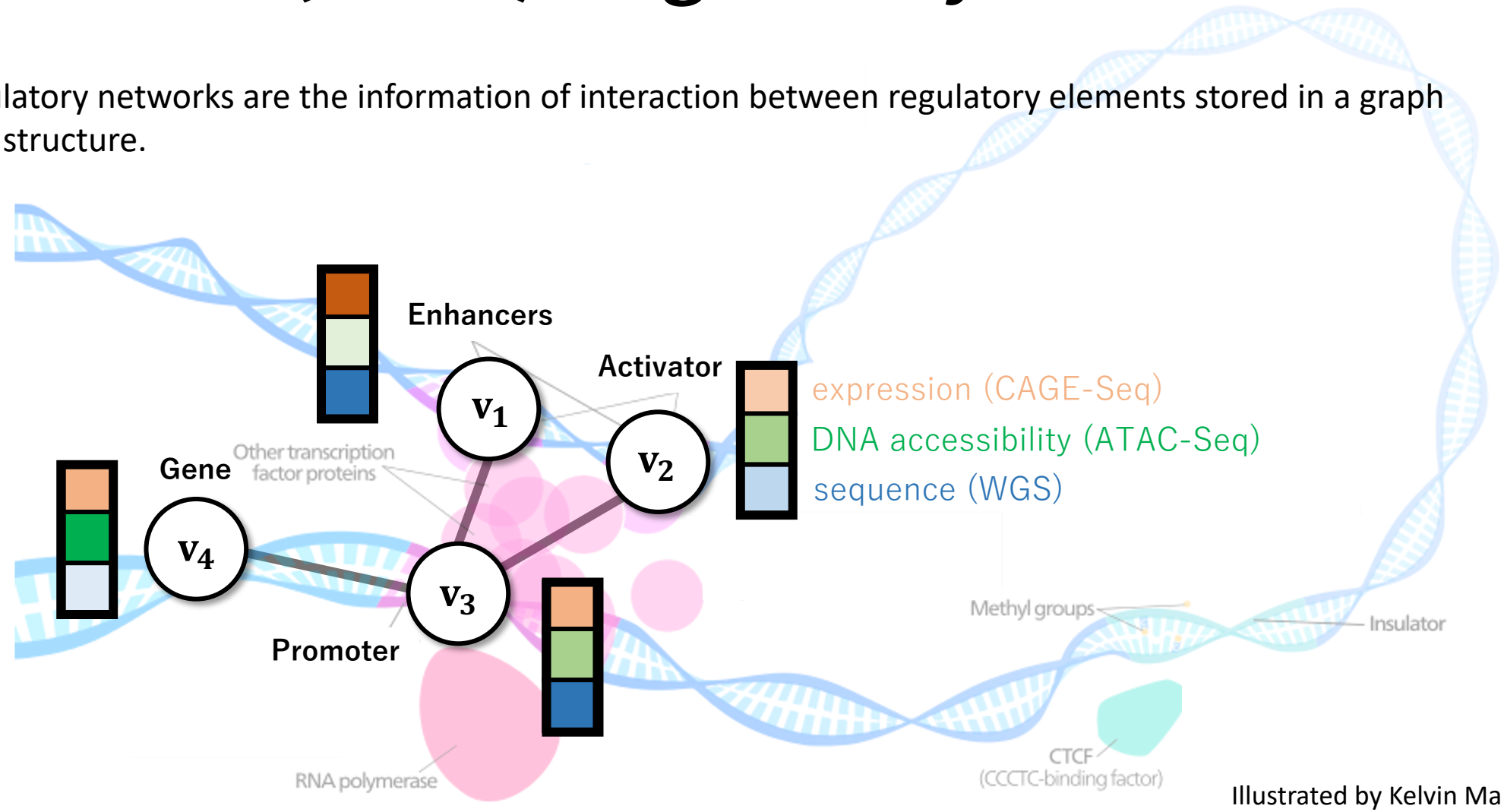
- homophilic graph
- non-homophilic graph

Edges Property

- weighted graph
- non-weighted graph
- directed graph
- undirected graph

What are (Gene) Regulatory Networks

Regulatory networks are the information of interaction between regulatory elements stored in a graph data structure.



Illustrated by Kelvin Ma

How to Build (Gene) Regulatory Networks

Experimental Data Input

- Transcription factor activity
  - ChIP-Seq
- Histone modification
  - ChIP-Seq
- DNA accessibility
  - ATAC-Seq, DNase-Seq
- Expression
  - RNA-Seq, CAGE-Seq
- Map Transcription Start Site (TSS)
  - CAGE-Seq
- Spatial Proximity
  - Hi-C, Hi-ChIP

Computational Approach

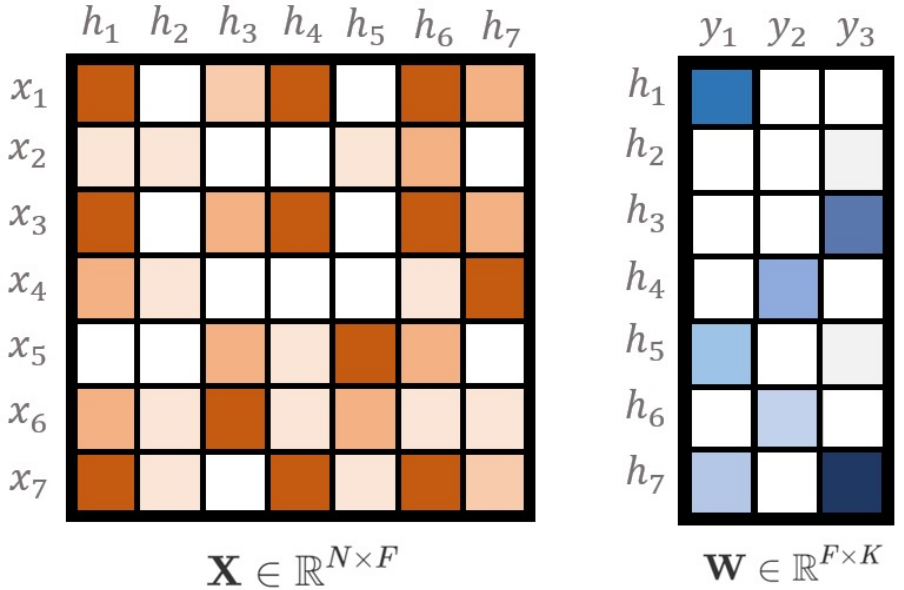
- Pairwise association
  - ARACNE, MRNet, WGCNA
- Machine learning w/ feature selection
  - Matrix Factorization, Tree-based
    - GENIE3 (feature importance)
    - TIGRESS (stability selection)
    - iRafNet (feature importance)
    - COREGNET (apriori algorithm)
  - Deep neural network
    - DeepBind (CNN filter)
    - BPNet (saliency map, CNN filter)
    - Enformer (saliency map, CNN filter)
    - GraphReg (saliency map, DeepSHAP)

How to Build (Gene) Regulatory Graphs

Machine learning w/ feature selection

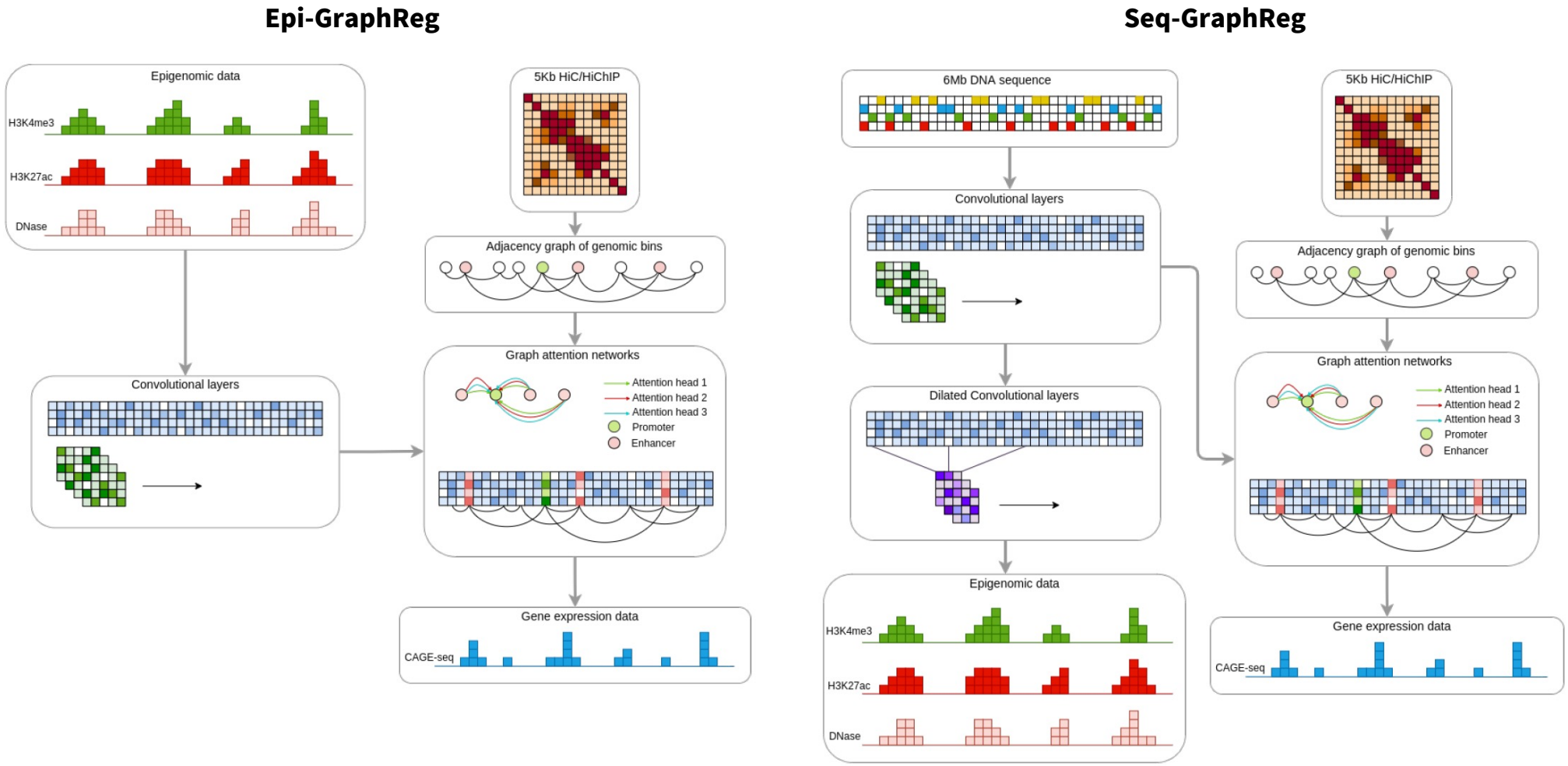
- TIGRESS Model
  1. Given the feature matrix of K gene $\mathbf{Y}$ and a feature matrix of N potential regulators $\mathbf{X}$
  2. Build a linear predictor for the targeted gene:
     $$\mathbf{Y} = \mathbf{X}\mathbf{w} + \varepsilon$$
  3. Use least angle regression (LASSO) to make sure the weight is sparse.

- TIGRESS Feature Extraction
  1. (Stability selection) run step 2 on randomly sampled (on features/columns) dataset several times.
  2. Compute the frequency of the weight is not zero for each corresponding feature.

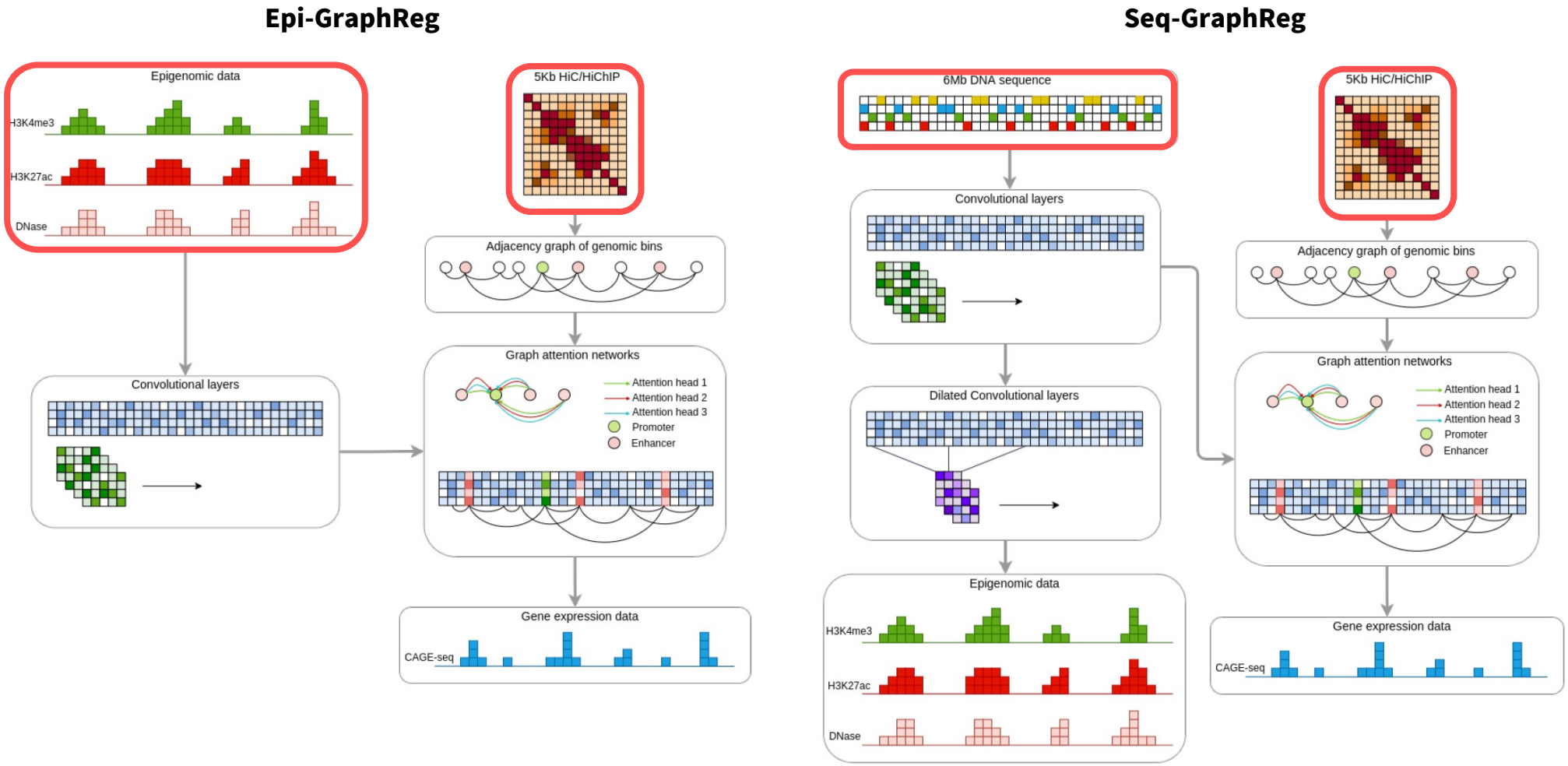


Methods

Overall Network Structure



Data Processing



Data Processing

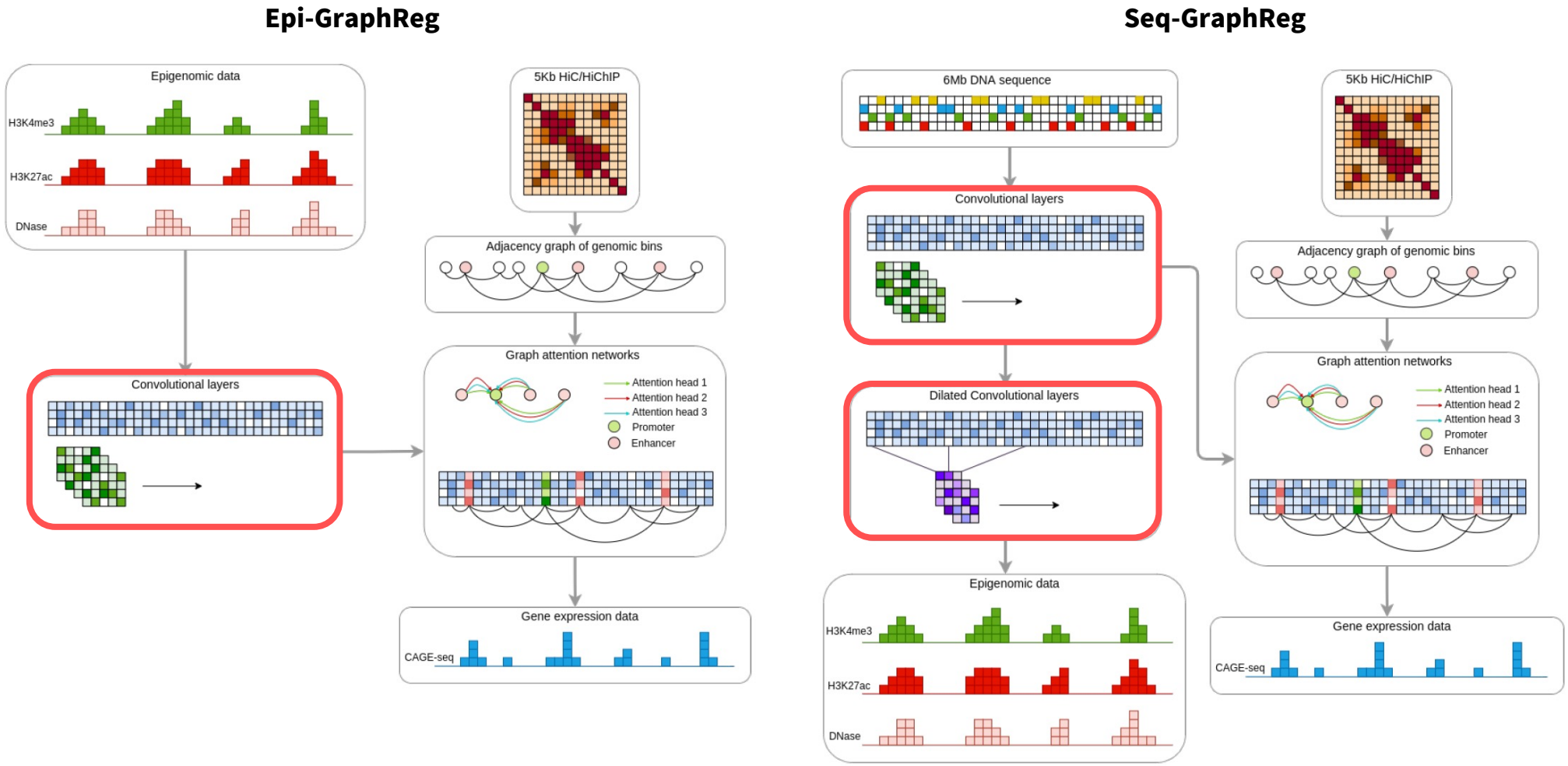
Epi-GraphReg

- Features
  - H3K4me3 ChIP-seq (100bp)
  - H3K27ac ChIP-seq (100bp)
  - DNase-seq (or ATAC-seq) (100bp)
  - HiC/HiChIP (5kb)
  - Across 3Mb each side
- Target
  - CAGE-Seq (5Kb)
  - Across 1Mb each side (400 bins in total)

Seq-GraphReg

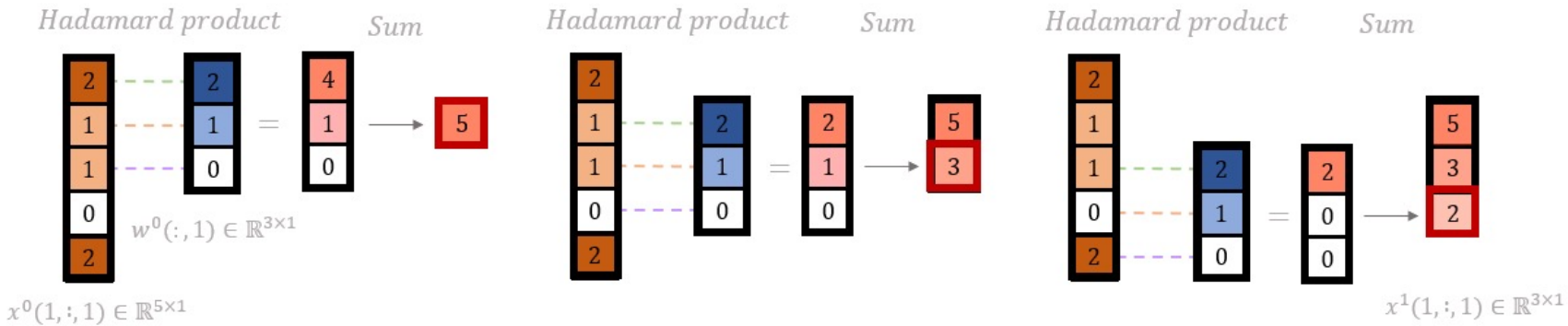
- Features
  - One-hot encoded DNA sequence (100bp)
  - HiC/HiChIP (5kb)
  - Across 3Mb each side
- Target
  - After dilated CNN Layer
    - H3K4me3 ChIP-seq (100bp)
    - H3K27ac ChIP-seq (100bp)
    - DNase-seq (or ATAC-seq) (100bp)
    - Across 3Mb each side (60000 bins in total)
  - After GAT Layer
    - CAGE-Seq (5kb)
    - Across 1Mb each side (400 bins in total)

Convolutional Neural Network



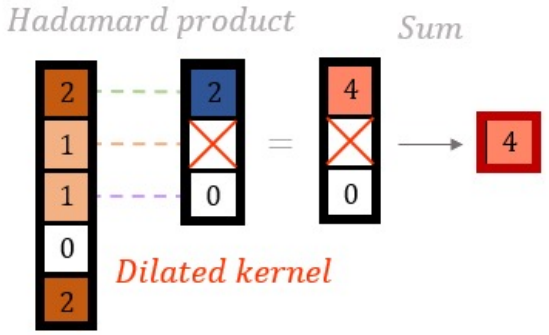
Convolutional Neural Network

1D Convolutional Neural Network

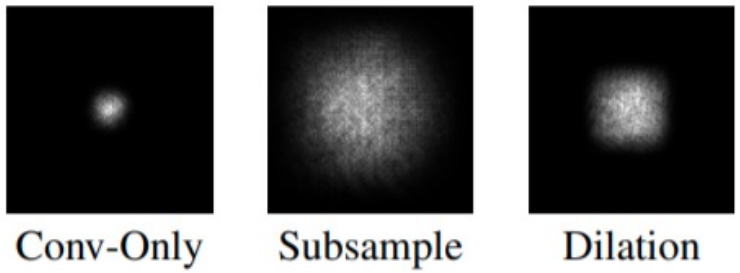


(a) 1: 3 (a) 2: 4 (c) 3: 5

Dilated Convolutional Neural Network



Effect of Dilated Convolutional Neural Network

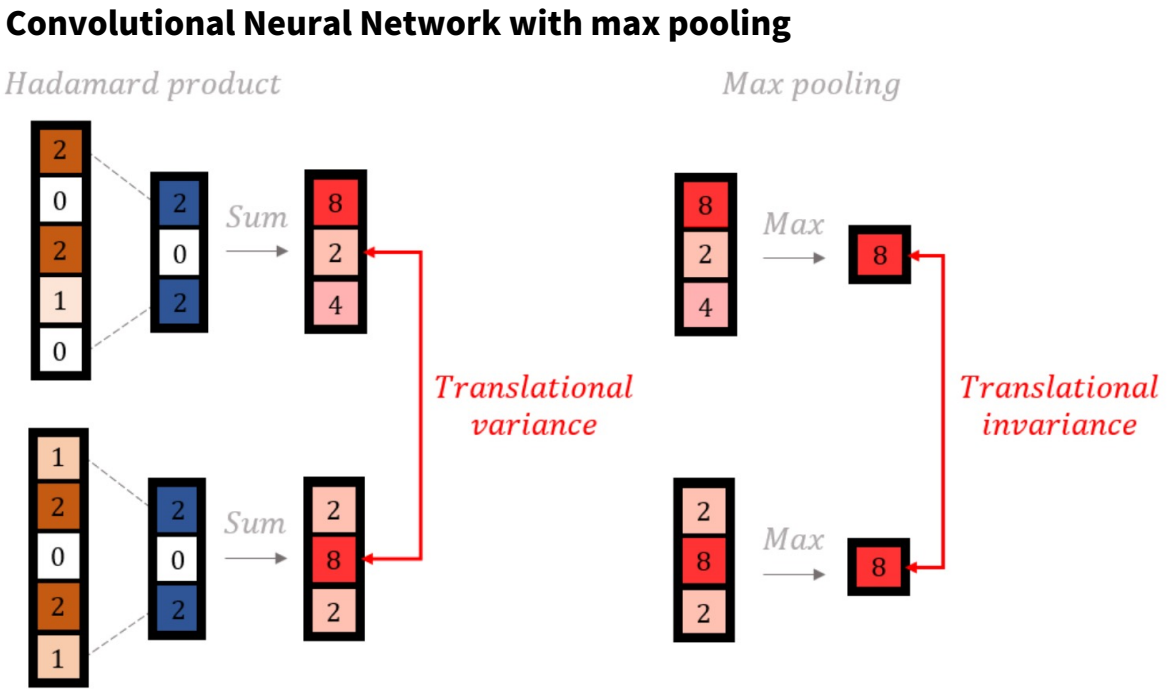


(Luo et al. 2016)

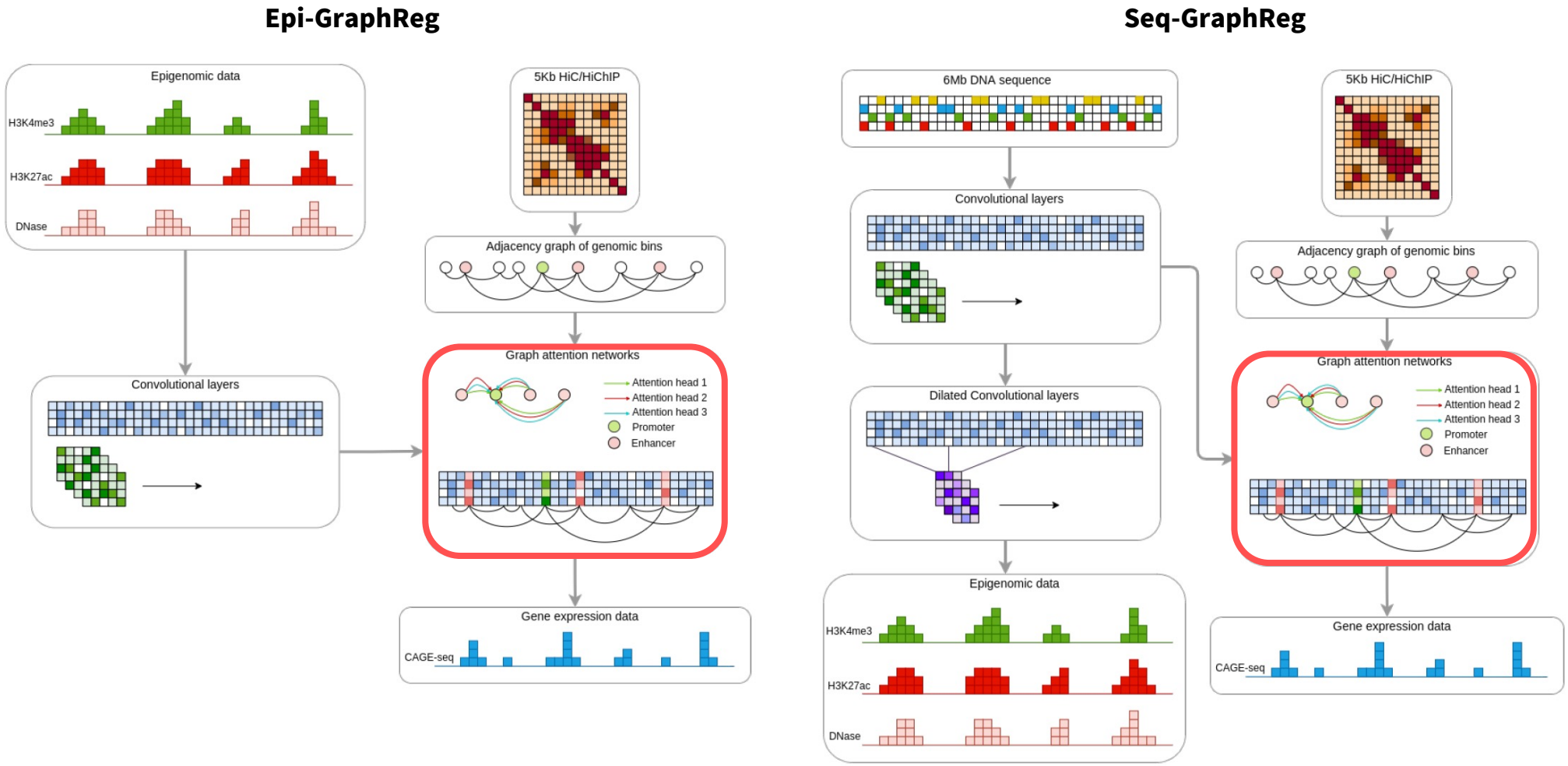
Convolutional Neural Network

- Multiple layers of CNN with max pooling is used to reduce the number of bins from 60000 to 1200.

- The number of filters is 64 in the last convolutional layer.



Graph Attention Network



Attention Mechanism

Key: Horse

Key: Donkey

XW^k: keys XW^q: queries XW^v: representation
$attention$: similarity between keys and queries

Query: Mule

The representation of mule is defined by the similarity between the query (mule) and keys (horse, donkey) in database.

Suppose the mule is 50% like horses and 50% like donkeys
The representation would be the average of the representation of horses and donkeys

Graph Attention Network

- Compute Bahdanau attention between nodes.

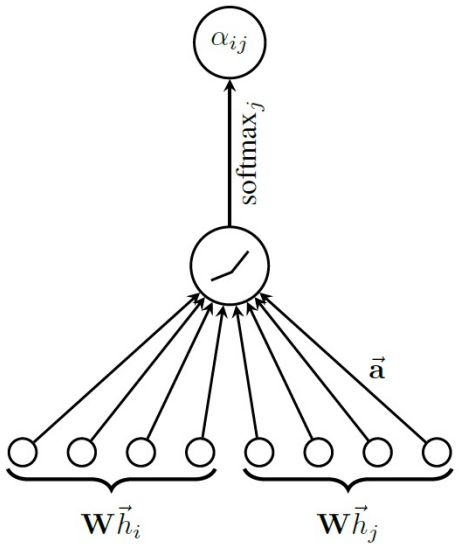
$$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$$

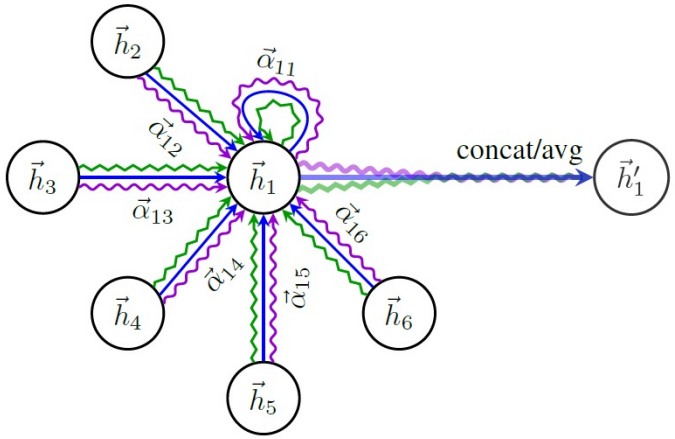
$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)}$$

- Perform node aggregation.

$$\vec{h}'_i = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{W}\vec{h}_j\right)$$

$$\vec{h}'_i = \Big\Vert_{k=1}^{K}\sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}^k\mathbf{W}^k\vec{h}_j\right)$$





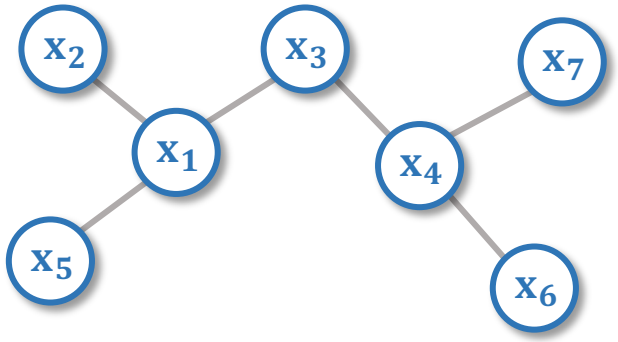
Graph Attention Network (1)

Initialization

1. Initialize the temporary embedding $\mathbf{H^0} = \mathbf{HW}$

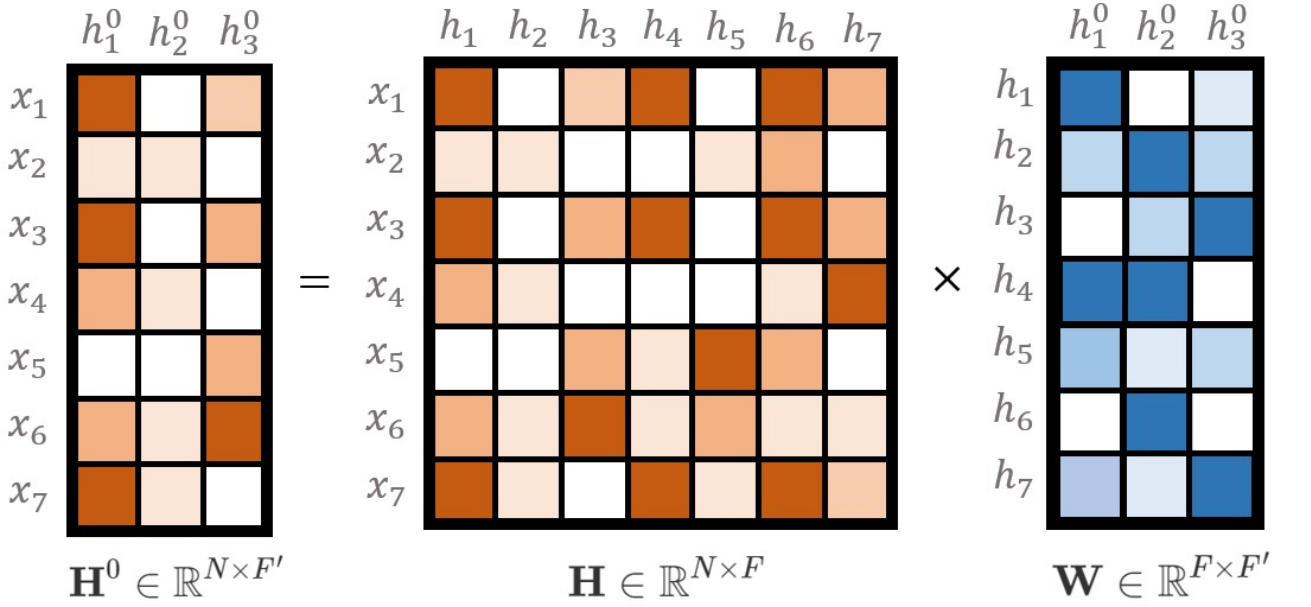
input graph

$G = (V, E)$



$$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$$

use 2 separate weight



$\mathbf{H}^0 \in \mathbb{R}^{N \times F'}$ $\mathbf{H} \in \mathbb{R}^{N \times F}$ $\mathbf{W} \in \mathbb{R}^{F \times F'}$

$$\beta_{i,j}^t = \sigma\left((a_p^t)^T W_p^t h_i^t + (a_e^t)^T W_e^t h_j^t\right)$$

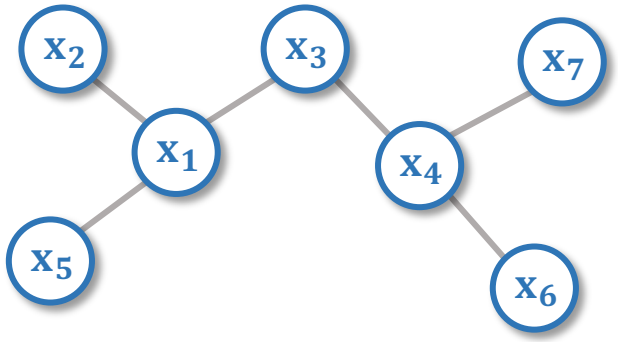
Graph Attention Network (2)

Attention Coefficient

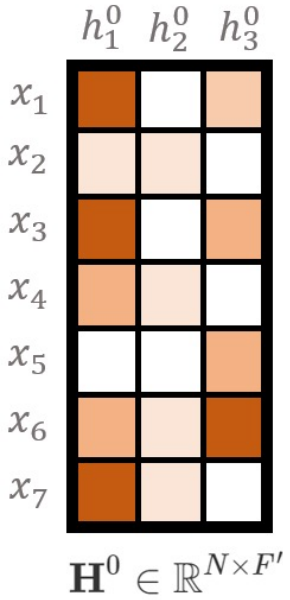
1. Initialize the temporary embedding $\mathbf{H^0} = \mathbf{HW}$
2. Compute attention coefficient with neighboring nodes

input graph

$G = (V, E)$



$$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$$



$\mathbf{H}^0 \in \mathbb{R}^{N \times F'}$

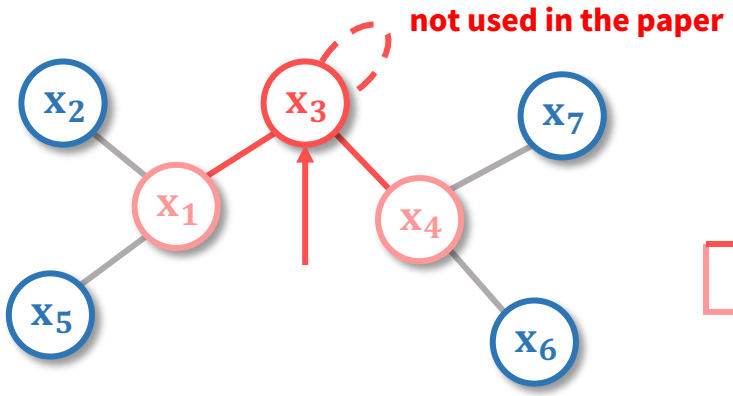
Graph Attention Network (3)

Attention Coefficient

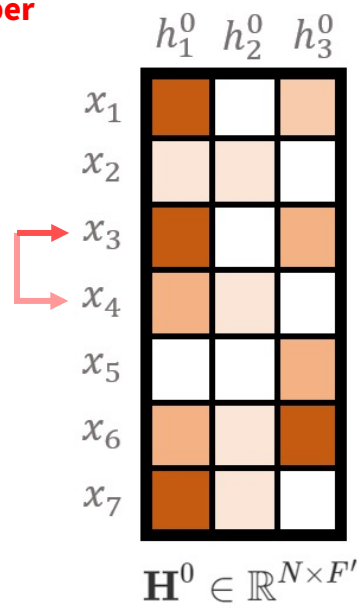
1. Initialize the temporary embedding $\mathbf{H^0} = \mathbf{HW}$
2. Compute attention coefficient with neighboring nodes
    1. Concatenate x_i with x_j

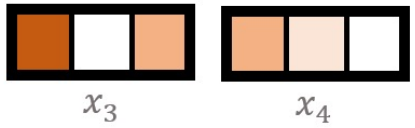
input graph

$G = (V, E)$



$$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$$





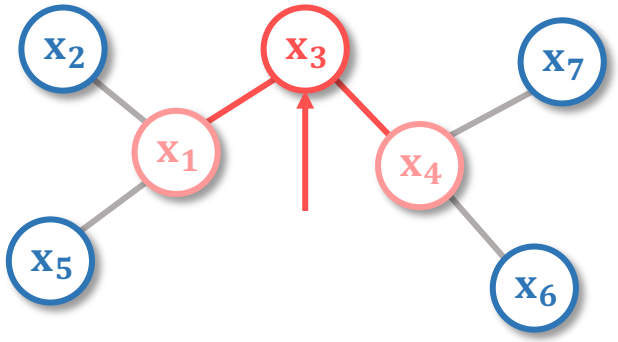
Graph Attention Network (4)

Attention Coefficient

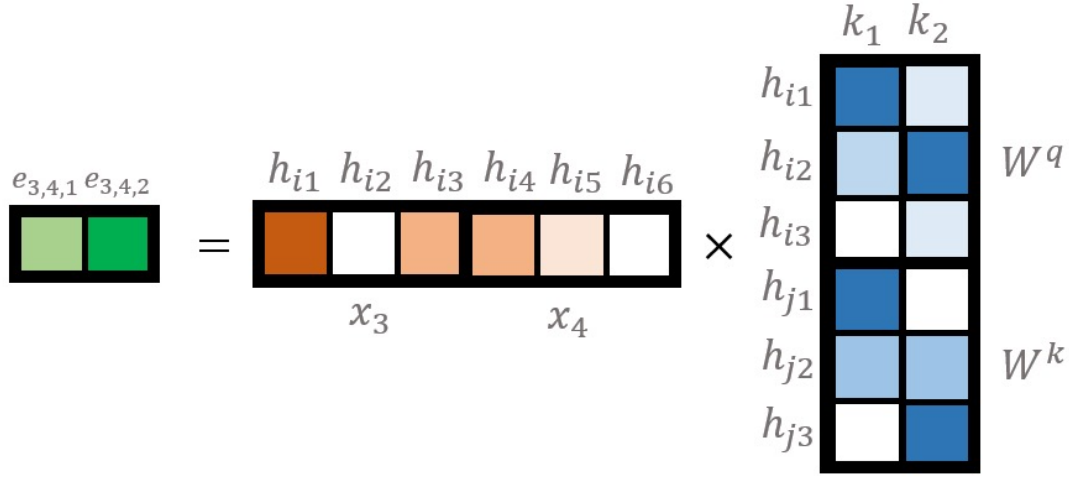
1. Initialize the temporary embedding $\mathbf{H^0} = \mathbf{HW}$
2. Compute attention coefficient with neighboring nodes
    1. Concatenate x_i with x_j
    2. Compute e_{ij}

input graph

$G = (V, E)$



$$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$$



$$\beta_{i,j}^t = \sigma\left((a_p^t)^T W_p^t h_i^t + (a_e^t)^T W_e^t h_j^t\right)$$

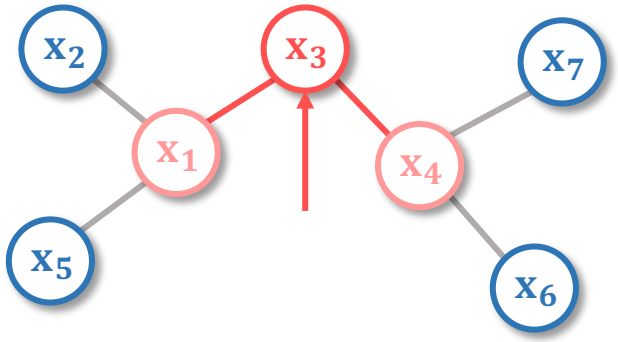
Graph Attention Network (5)

Attention Coefficient

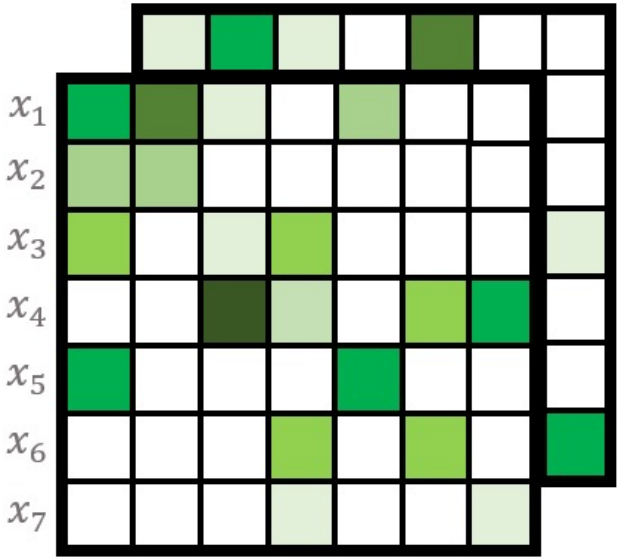
1. Initialize the temporary embedding $\mathbf{H^0} = \mathbf{HW}$
2. Compute attention coefficient with neighboring nodes
    1. Concatenate x_i with x_j
    2. Compute e_{ij}

input graph

$G = (V, E)$



$$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$$



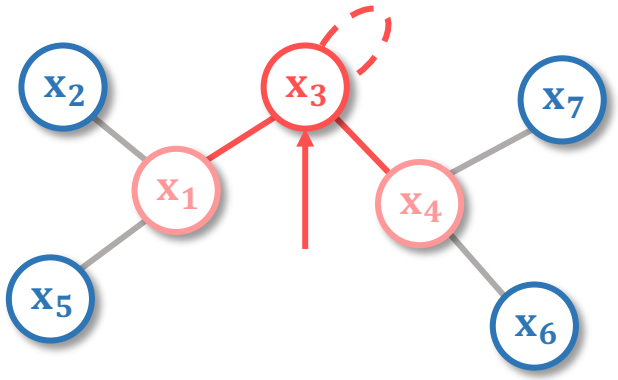
$\mathbf{E} \in \mathbb{R}^{K \times N \times N}$

Graph Attention Network (6)

Attention Coefficient

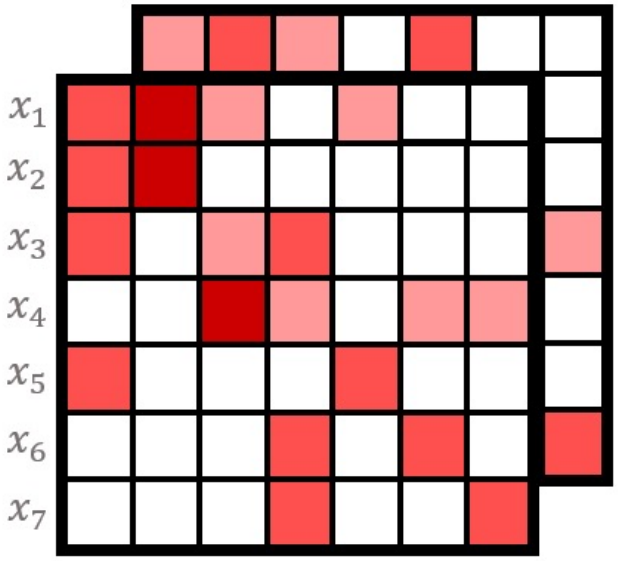
input graph

$G = (V, E)$



$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{a}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\vec{a}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)}$$

1. Initialize the temporary embedding $\mathbf{H^0} = \mathbf{HW}$
2. Compute attention coefficient with neighboring nodes
   1. Concatenate x_i with x_j
   2. Compute e_{ij}
   3. Apply LeakyReLU and a softmax layer



$$\sum_{j=1}^{N} a_{ij} = 1 \qquad 0 \le a_{ij} \le 1$$

LeakyReLU

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ ax & \text{otherwise} \end{cases}$$

Softmax

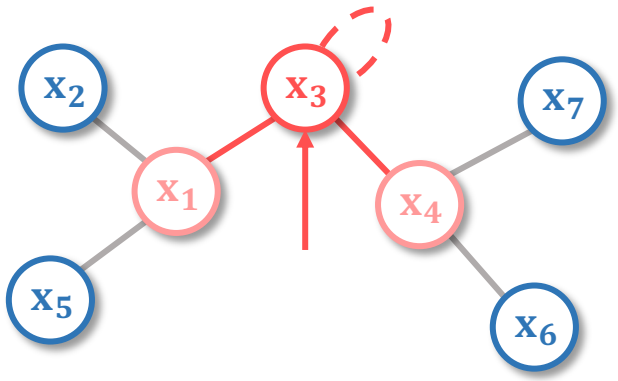
$$f(\mathbf{x}_i) = \frac{\exp(\mathbf{x}_i)}{\sum_{j=1}^{N}\exp(x_{ij})}$$

Graph Attention Network (7)

Aggregation

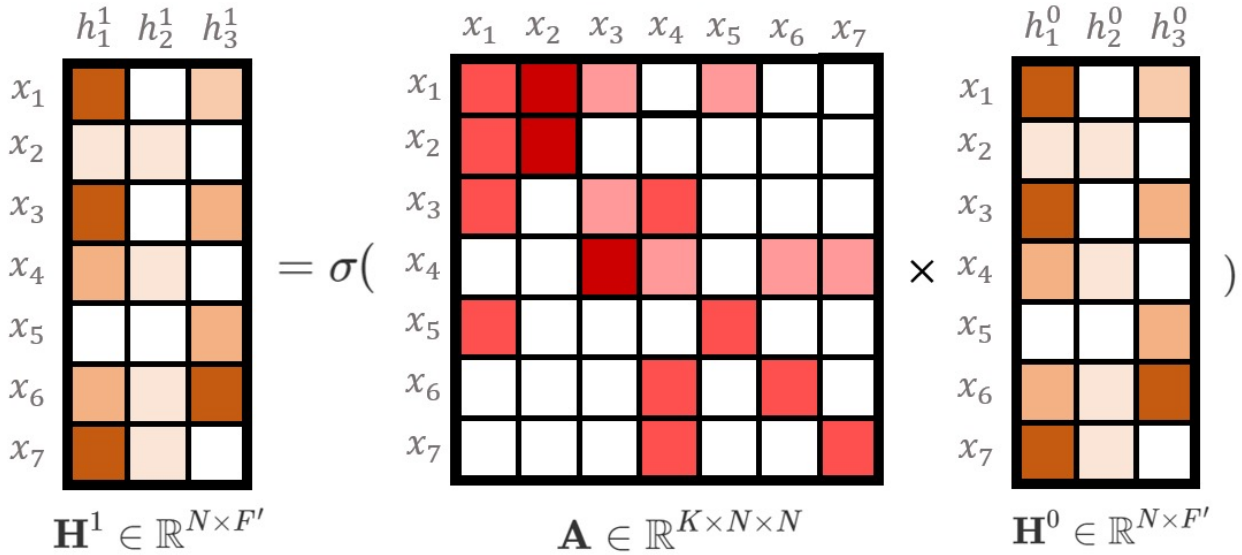
input graph

$G = (V, E)$



$$\vec{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \vec{h}_j\right)$$

1. Initialize the temporary embedding $\mathbf{H^0} = \mathbf{HW}$
2. Compute attention coefficient with neighboring nodes
3. Aggregation



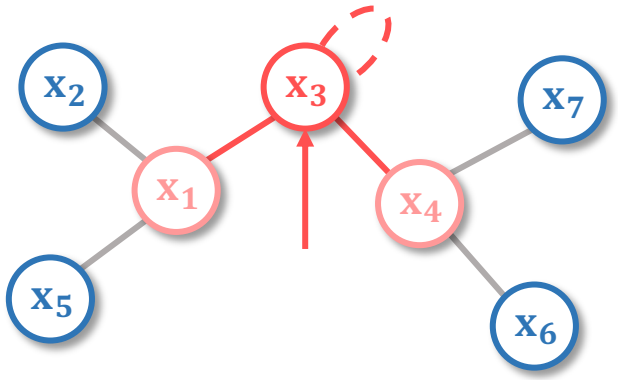
$\mathbf{H}^1 \in \mathbb{R}^{N \times F'}$ $\mathbf{A} \in \mathbb{R}^{K \times N \times N}$ $\mathbf{H}^0 \in \mathbb{R}^{N \times F'}$

Graph Attention Network (8)

Modification for GraphReg

input graph

$G = (V, E)$

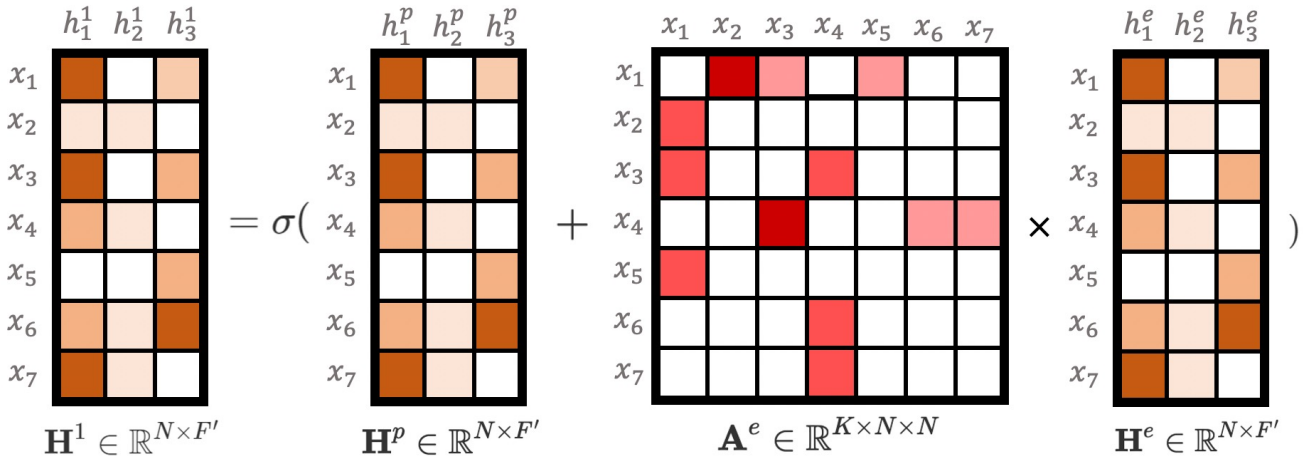


$$\vec{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \vec{h}_j\right)$$

1. Initialize the temporary embedding $\mathbf{H^0} = \mathbf{HW}$
2. Compute attention coefficient with neighboring nodes
3. Aggregation

use 2 weight in step 1

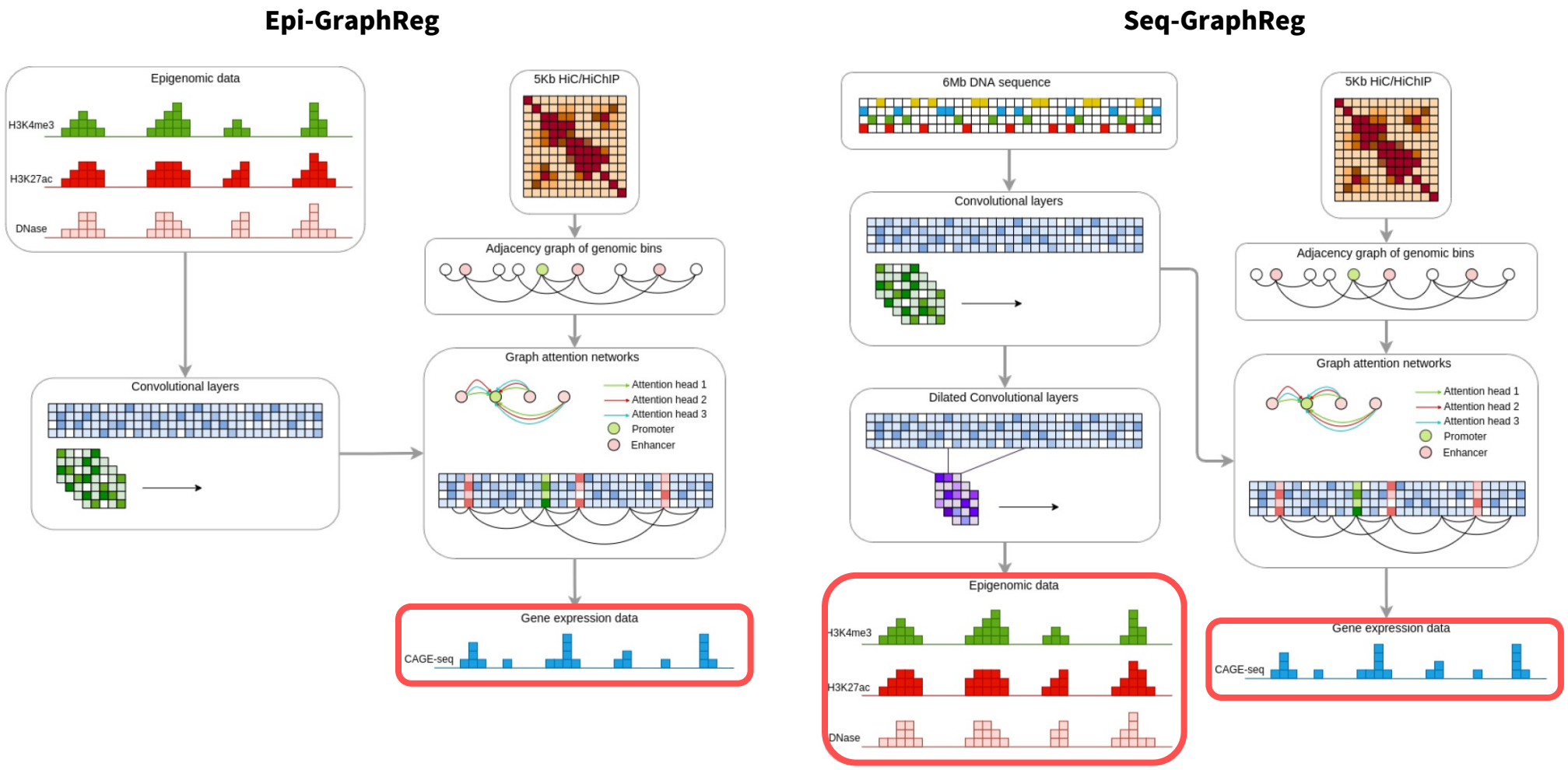
remove diagonal in e



$$\alpha_i^t = \sigma\left(a^t \sqrt{|\mathcal{N}_i|} + b^t\right)$$

$$h_i^{t+1} = f\left(\alpha_i^t \circ \left(W_p^t h_i^t + \sum_{j \in \mathcal{N}_i} \beta_{i,j}^t W_e^t h_j^t\right)\right)$$

Loss Function



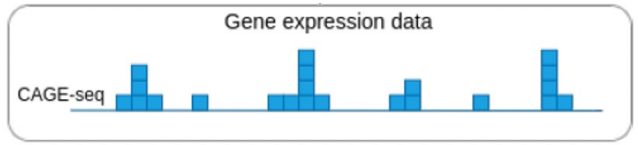
Loss Function

Poisson Regression

$$E(Y|X) = \exp(f_i^\theta(X, G))$$

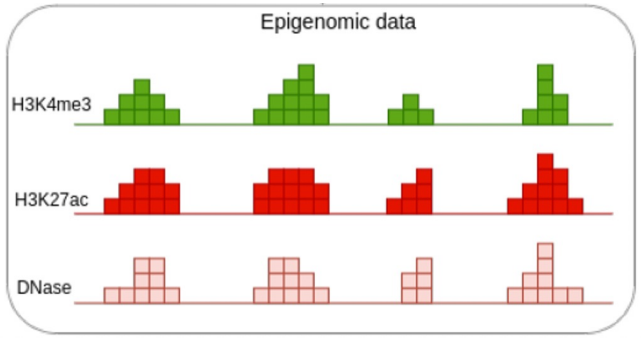
Epi-GraphReg (and Seq-GraphReg) on CAGE-Seq

$$L_E = \frac{3}{N} \sum_{i=N/3}^{2N/3} \left(\log \Gamma(y_i + 1) + f_i^\theta(X, G) - y_i \log f_i^\theta(X, G) \right)$$



Seq-GraphReg on Epigenomic Data

$$L_S^e = \frac{1}{N'} \sum_{i=1}^{N'} \left(\log \Gamma(y_i^e + 1) + g_e^\theta(X) - y_i^e \log g_i^e(X) \right)$$



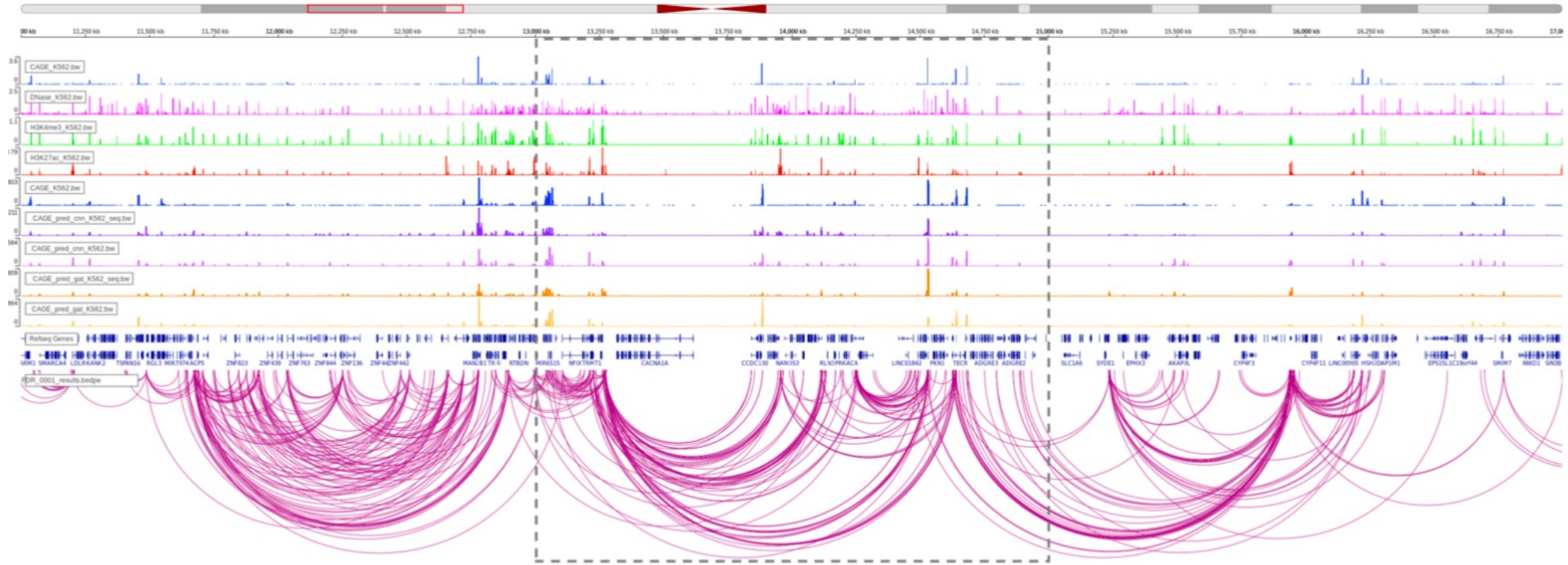
Seq-GraphReg

$$L_S = \lambda L_S^{cage} + (1 - \lambda) \sum_e L_S^e \qquad \lambda = max(0.1, \frac{1}{1 + \exp - (epoch - 10)})$$

CAGE-Seq

$$L_E = \frac{3}{N} \sum_{i=N/3}^{2N/3} \left(\log \Gamma(y_i + 1) + f_i^\theta(X, G) - y_i \log f_i^\theta(X, G) \right)$$

Training Feature **Training Feature + Prediction Target** **Training Feature**



Training Feature + Prediction Target **Training Feature + Prediction Target** **Training Feature + Prediction Target**

$$L_S^e = \frac{1}{N'} \sum_{i=1}^{N'} \left(\log \Gamma(y_i^e + 1) + g_e^\theta(X) - y_i^e \log g_i^e(X) \right)$$

Epigenomics

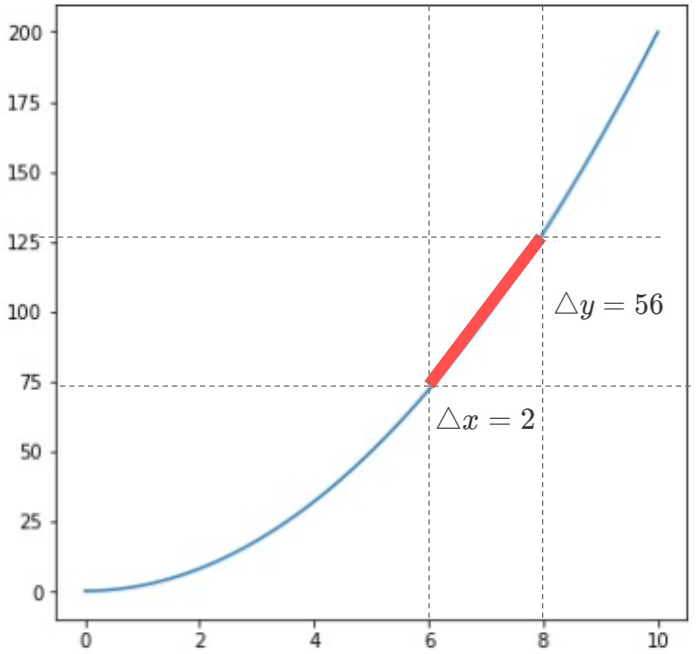
Training and CNN-Reg model

- Architecture
  - GraphReg
    - 3 GAT layers with 4 attention heads
  - CNNReg
    - Replace GAT layers with 8 Dilated CNN Layers
- Dataset
  - Human GM12878 and K562 $(i = 1\ to\ 10)$
    - Validation: $i, i + 10$ chromosome
    - Test: $i + 1, i + 11$ chromosome
  - Mouse mESC $(i = 1\ to\ 10)$
    - Validation: $i, (i + 10\ mod\ 20) + sign(i \geq 10)$ chromosome
    - Test: $i + 1, (i + 11\ mod\ 20) + sign(i \geq 9)$ chromosome

* Number of CNN Layers is not shown

Saliency Map (Smooth Gradient)

- Consider a non-linear function:
  $f(x) = y = 2x^2$
- How much the model output will change when the input is change.
  $\nabla_x f(x) = 4x$
- Generalize to higher dimension
  $\nabla_{\mathbf{x}} f(\mathbf{x}) = f'(\mathbf{x})$
- Consider the input data, if there is no signal, the contribution should be low.
  $\frac{1}{N}\sum_{i=1}^{n} \nabla_{\mathbf{x}+\epsilon} f(\mathbf{x}+\epsilon)$ (Smilkov *et al.* 2017)



DeepSHAP (DeepLIFT)

- Core idea: define a reference (random sequence for DNA, 0 for epigenomic data), and compute the difference between the samples and the reference:

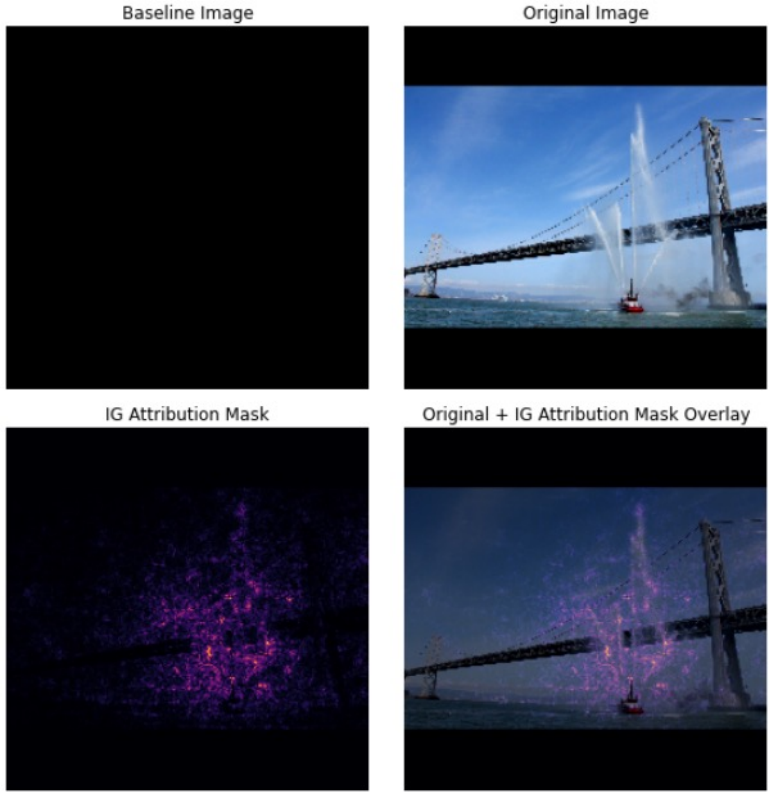
$$\Delta t = t - t^0 \qquad \sum_{i=1}^{n} C_{\Delta x_i \Delta t} = \Delta t$$

- Separate positive and negative contribution

$$C_{\Delta y \Delta t} = C_{\Delta y^+ \Delta t} + C_{\Delta y^- \Delta t}$$

- Derive contribution

$$C_{\Delta x_i^+ \Delta y^+} = 1\{w_i \Delta x_i > 0\} w_i \Delta x_i^+$$
$$C_{\Delta x_i^- \Delta y^+} = 1\{w_i \Delta x_i > 0\} w_i \Delta x_i^-$$
$$C_{\Delta x_i^+ \Delta y^-} = 1\{w_i \Delta x_i < 0\} w_i \Delta x_i^+$$
$$C_{\Delta x_i^- \Delta y^-} = 1\{w_i \Delta x_i < 0\} w_i \Delta x_i^-$$



Integrated Gradients Tensorflow Tutorial

Results

Main Results

Results obtained from held-out validation and test chromosomes for 3 datasets (Humans: K562, GM12878 and Mouse: mESC).

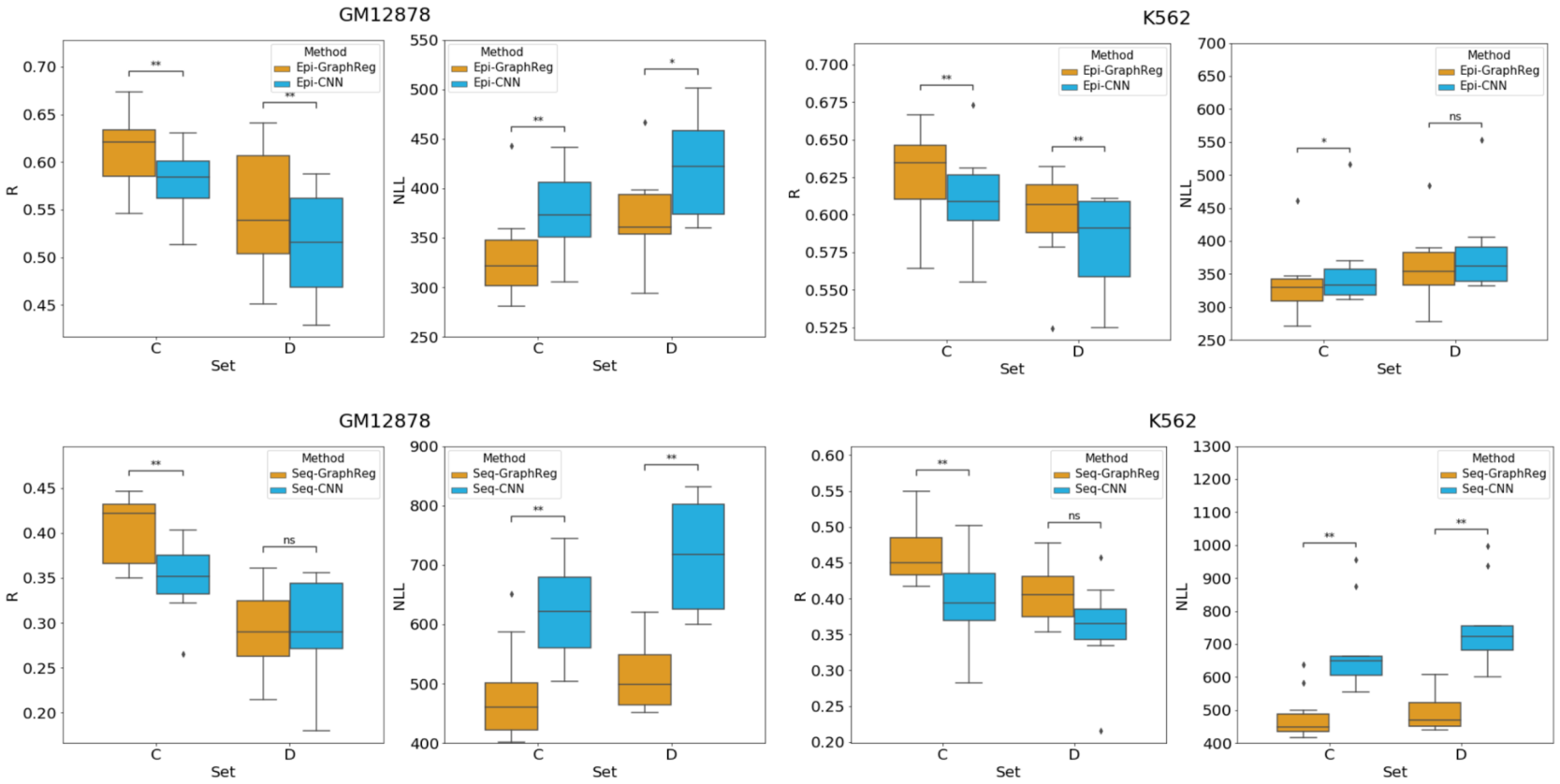
1. Performance of gene expression prediction
2. Recovery of gene expression fold-change
3. Generalization to unseen cell-type
4. Validation through recovery of true distal regulatory elements

Prediction Performance

Performance metric

- Pearson correlation (R)
- Poisson distribution negative log-likelihood (NLL)

* Set C: expressed genes w/ at least one E-P
Set D: expressed genes w/ al least five E-P



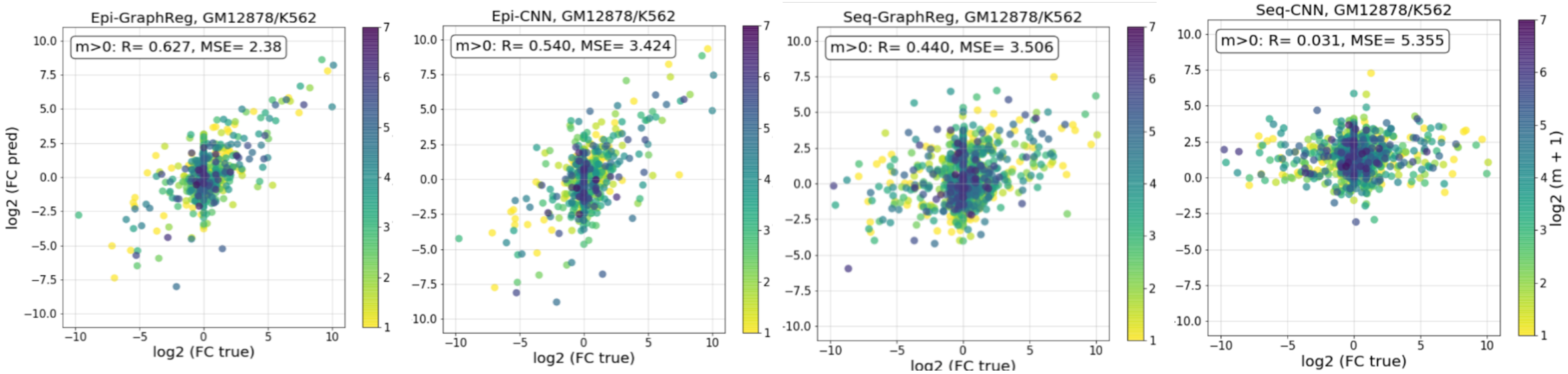
Key Take-away

- GraphReg models outperform all CNN-models.
- Seq-GraphReg performance increase compared to sequence-based CNN-model is more pronounced
    - Especially for complex genes with at least 1 distal regulator

Recovering Expression Fold-change between Cell Types

Criteria

- Pearson correlation (R)
- Mean Squared Error (MSE)

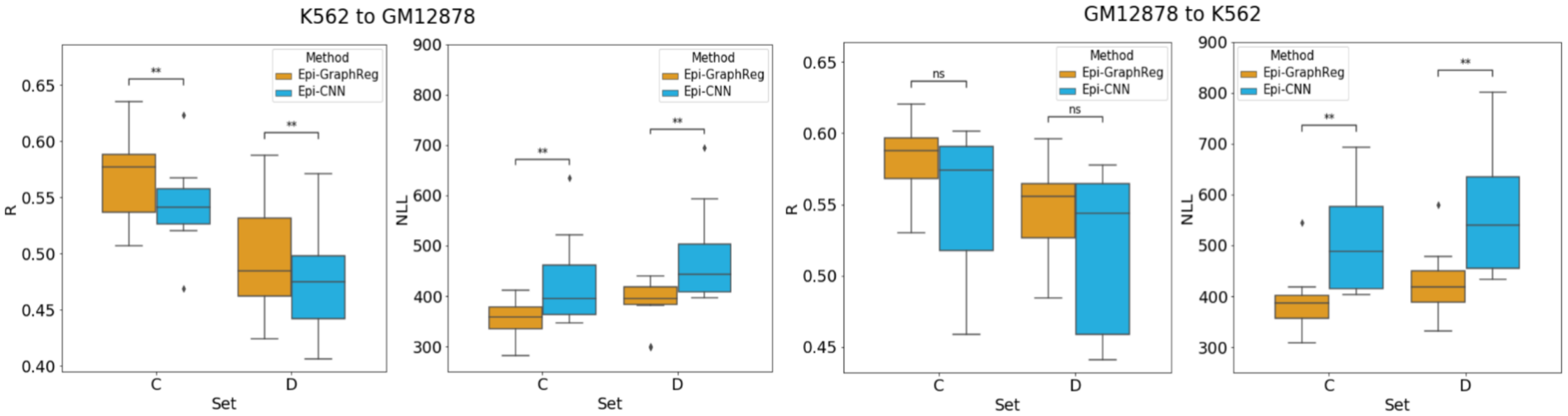


Key Take-away

- Epigenomic based models outperform CNN models in recovering differentially expressed genes between cell types.
- Sequence based models struggle to capture changes in cell-type specific gene expression.

Epi-GraphReg Generalises Better

- Predict CAGE signal in unseen cell-type
- Criteria
  - Pearson correlation (R)
  - Poisson distribution negative log-likelihood (NLL)

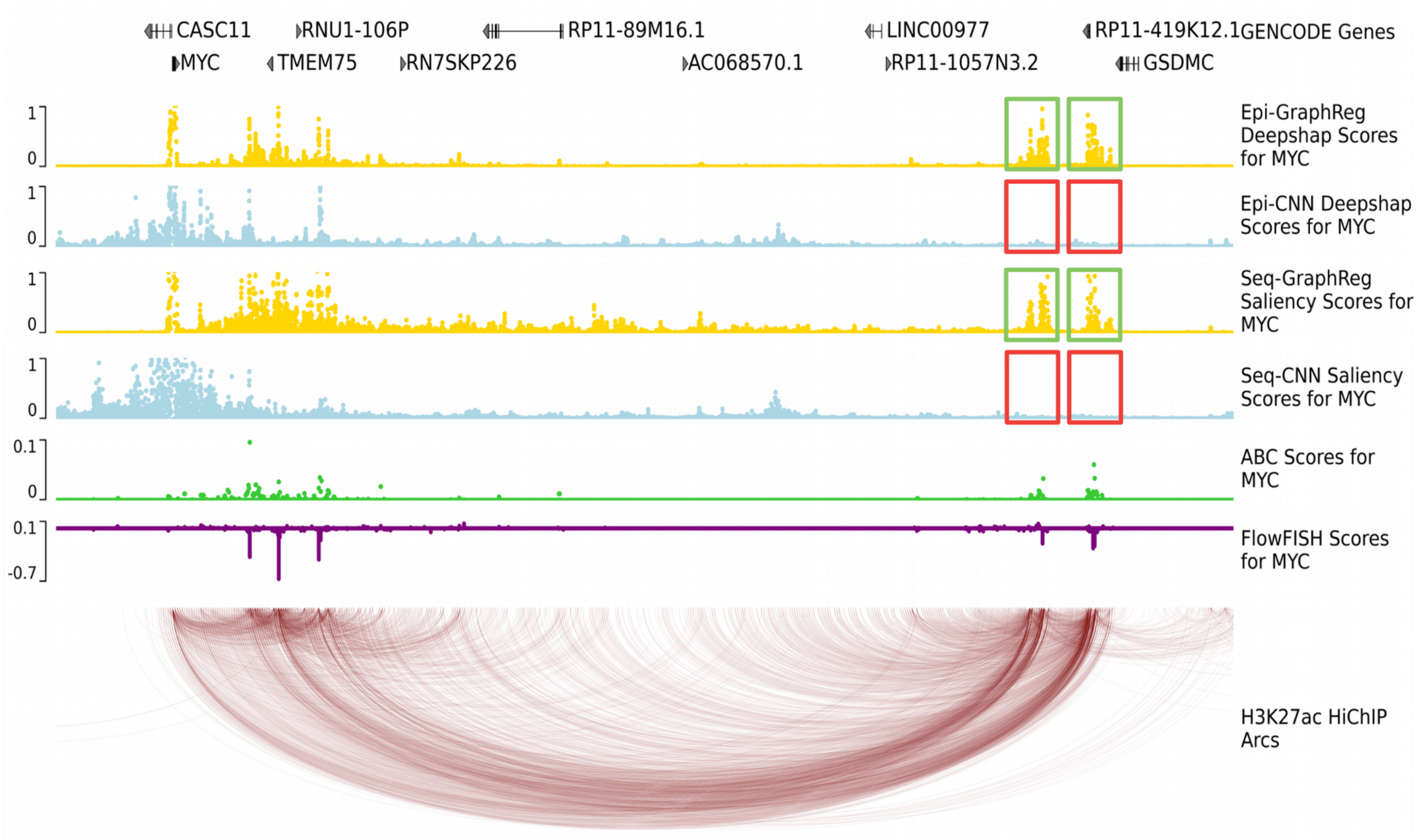


Recovering Distal Regulators using Feature Importance

- DeepSHAP and saliency map evaluate the contribution of individual feature for a given prediction ~ feature importance
  - Here how different positions within the 6Mb window contribute to predict a particular gene containing bin CAGE signal

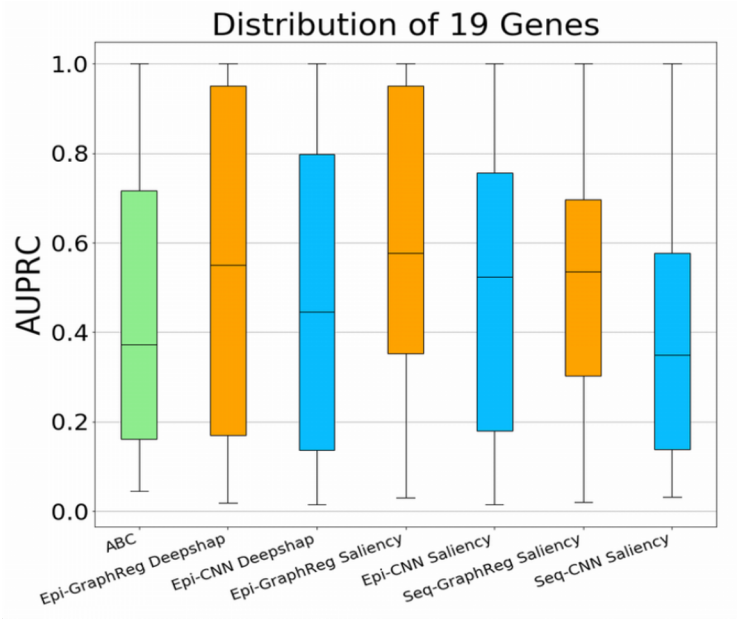
- Biological relevance of this feature importance evaluated by quantifying ability to recover experimentally validated distal regulatory interactions
  - CRISPRi-FlowFISH
    - Disrupt a particular distal regulatory element using“CRISPR”
    - Quantify effect on target gene using FISH
  - TAP-seq
    - Single-cell RNA-sequencing technique with greater sensitivity to probe 1000 genes simultaneously
    - Combined with enhancer“CRISPR”-perturbation to assay regulatory interactions

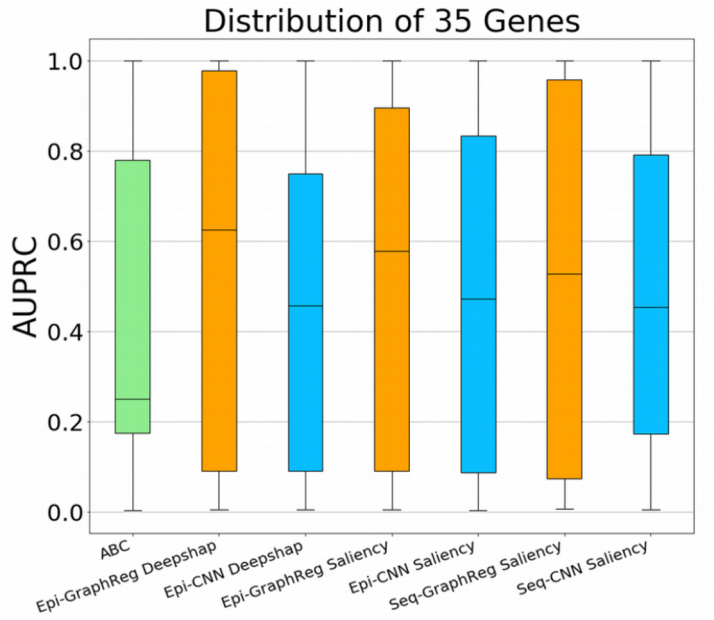
- How well can we recover experimentally observed regulatory interactions from our models’ feature importance?


CASC11
RNU1-106P
RP11-89M16.1
LINC00977
RP11-419K12.1
GENCODE Genes
MYC
TMEM75
RN7SKP226
AC068570.1
RP11-1057N3.2
GSDMC
1
0
Epi-GraphReg Deepshap Scores for MYC
Epi-CNN Deepshap Scores for MYC
Seq-GraphReg Saliency Scores for MYC
Seq-CNN Saliency Scores for MYC
0.1
ABC Scores for MYC
FlowFISH Scores for MYC
-0.7
H3K27ac HiChIP Arcs

AUPRC to Quantify this Correspondence

- From all possible set of distal regulatory elements (>10kb away) how many does the GraphReg model leverage for it's gene expression prediction?
  - For MYC: 8 true regulators out of 200 potential regulators
  - How many such true Gene-Enhancer pairs does the model effectively use to make its predictions?



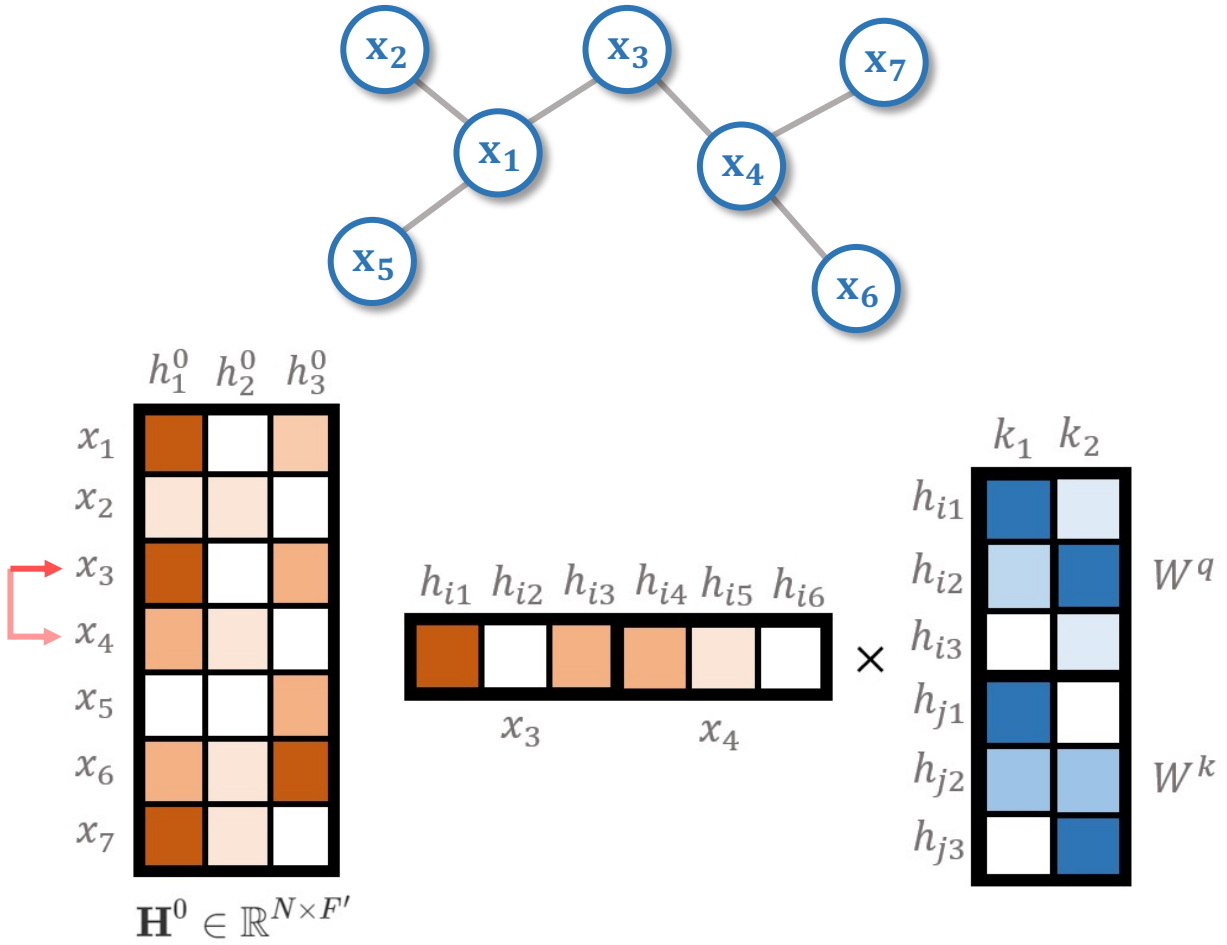


Discussion

Discussion (1)

Comparison with Enformer

- Attention Nodes
  - Enformer compute attention between all bins.
  - GraphReg compute attention between associated bins (using HiChIP data)
- Resolution (of Attention)
  - Enformer: 128bp
  - GraphReg: 5Kb
- Range
  - Enformer: 100kb
  - GraphReg: 3Mb
- Positional Embedding
  - Enformer: Yes
  - GraphReg: No



Discussion (2)

Regulatory Network

- Downstream analysis
  - Graph embedded into edges, then analyze the adjacency matrix.
  - Graph embedded into node feature matrix, then analyze the node feature matrix.
  - Joint prediction (not covered in this paper, could be a potential application).

- What do the edges (interaction) mean in regulatory network?
  - Similarity
  - Message passing
  - Physical interaction
  - Indirect regulation

- Although machine learning model could help us build regulatory network. Can we translate it into our knowledge (e.g. bins to genes, transcription factor etc.)
  - Lack of experimental support (e.g. which genomic region relate to which genes)
  - From data-driven model to knowledge of theories.