Multiscale and integrative single-cell Hi-C analysis with Higashi

Reproduction of the experiment

Group member:
何彥南、張修誠、陳偉瑄

Ruochi Zhang, Tianming Zhou, and Jian Ma. Nature biotechnology 40.2 (2022): 254-261.

Table of contents

01
Introduction

02
Methods

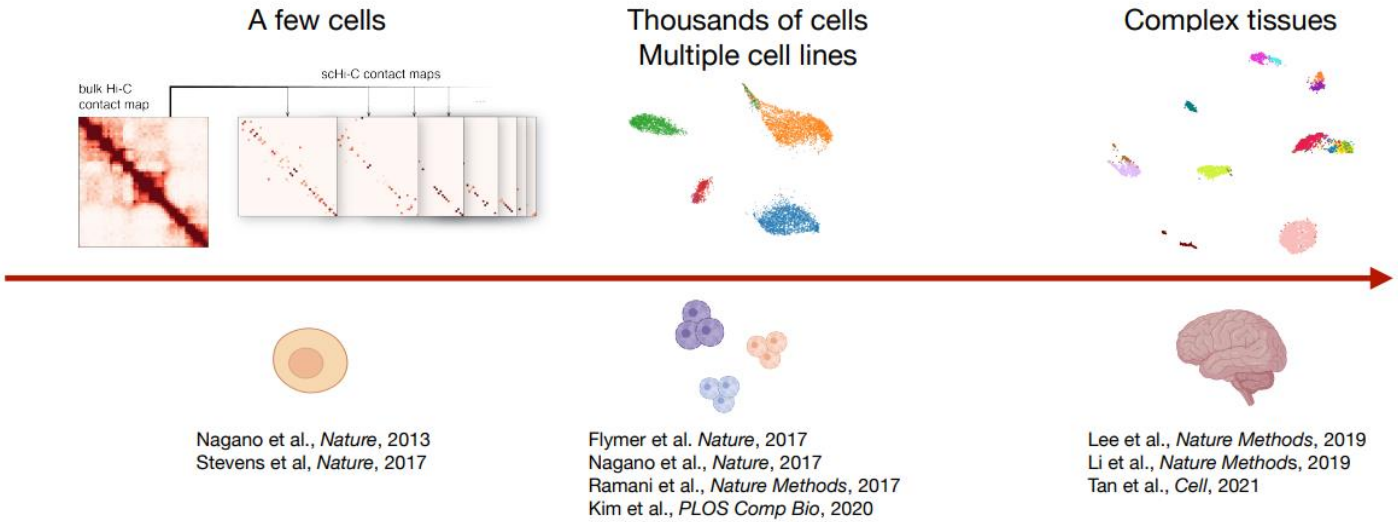
03
Results

04
Our Reproduction

01

Introduction

Single cell Hi-C reveals heterogeneity of genome organization



Challenges

1. High dimensionality
2. Sparse and noisy data
3. Co-assayed scHi-C
   DNA methylation

Goals

1. Generate embeddings
2. Impute sparse contact maps
3. Extract multi-scale 3D genome features

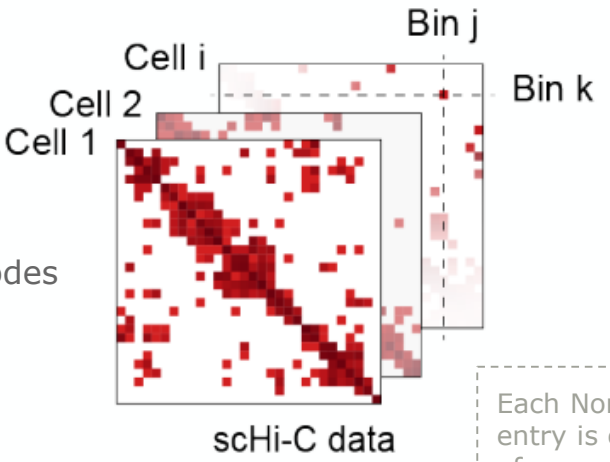
Modeling scHi-C data as a hypergraph

A hypergraph

$$G = (V, E)$$

- V/E : the set of nodes / hyperedges
- $e \in E$: a hyperedge connects two or more nodes

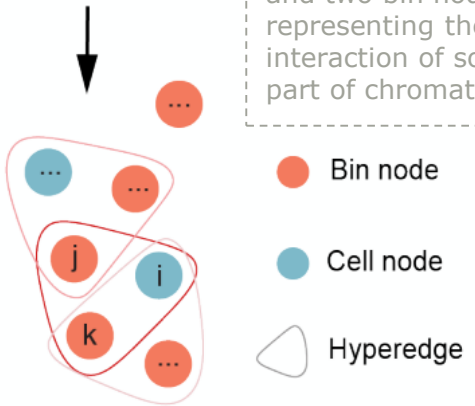
- Non-zero entry in contact matrix →
- Co-assayed signals → node attributes



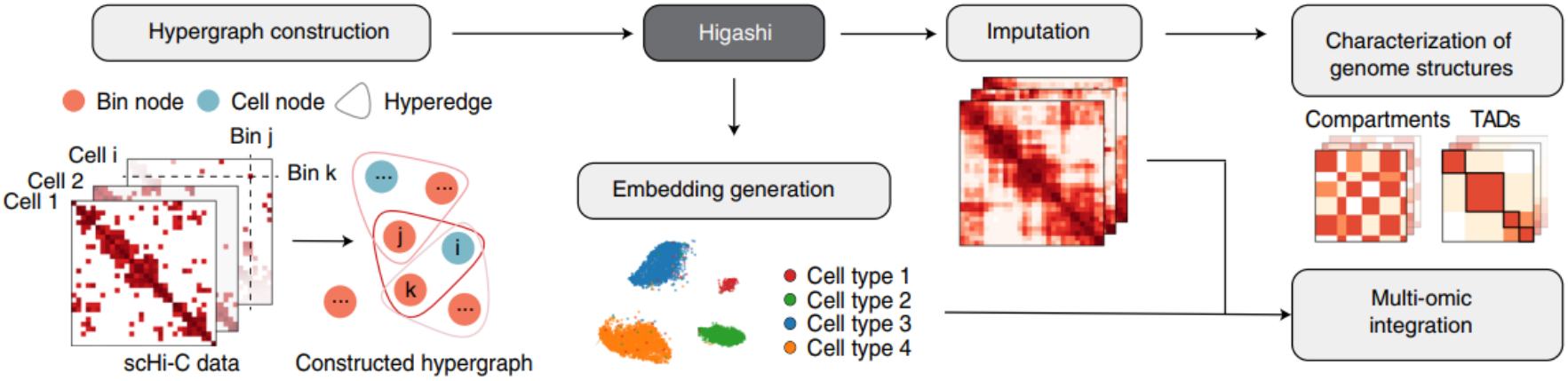
Each Non-zero entry is composed of one cell node and two bin nodes, representing the interaction of some part of chromatin.

Hypergraph representation learning

- Embeddings for the nodes → *Embeddings for cells*
- Predict the missing hyperedges → *Imputation of contact maps*
- Predict node attributes → *Joint modeling of co-assayed signals*



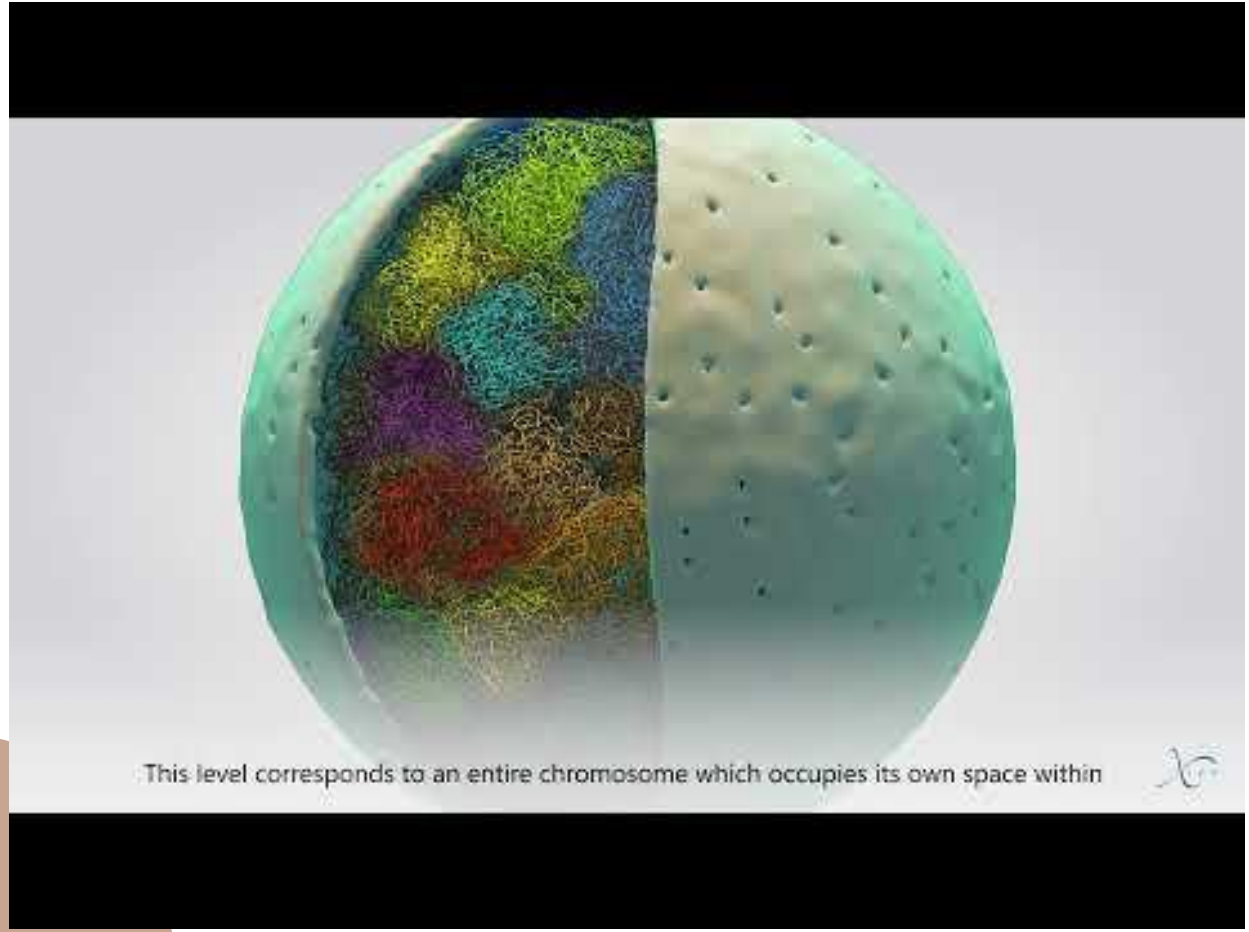
Overview of the Higashi framework



Advantages

- Integrate embedding and data imputation for scHi-C in the same formalism
- Flexibility to analyze co-assayed scHi-C datasets
- Incorporate the latent correlations between cells to enhance overall imputation

3D genome organization



02

Methods

scHi-C data and other genomic data processing

Single-cell Hi-C datasets

- Ramani et al.14: *(GEO) GSE84920* 👉 we use
- Nagano et al.15: *(GEO) GSE94489*
- 4DN sci-Hi-C20: *(4DN Data Portal) 4DNES4D5MWEZ, 4DNESUE2NSGS, 4DNESIKGI39T, 4DNES1BK1RMQ and 4DNESTVIP977*
- WTC-11 iPSC line: *(4DN Data Portal) 4DNESF829JOW and 4DNESJQ4RXY5*

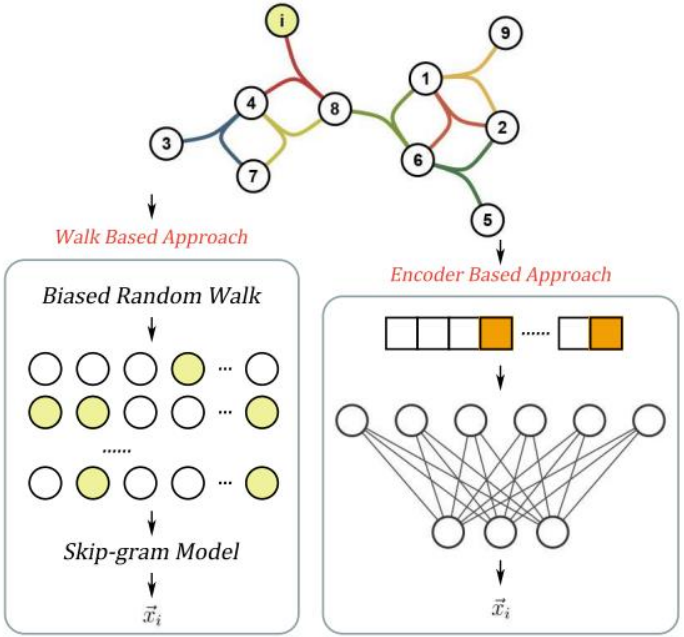
Data processing

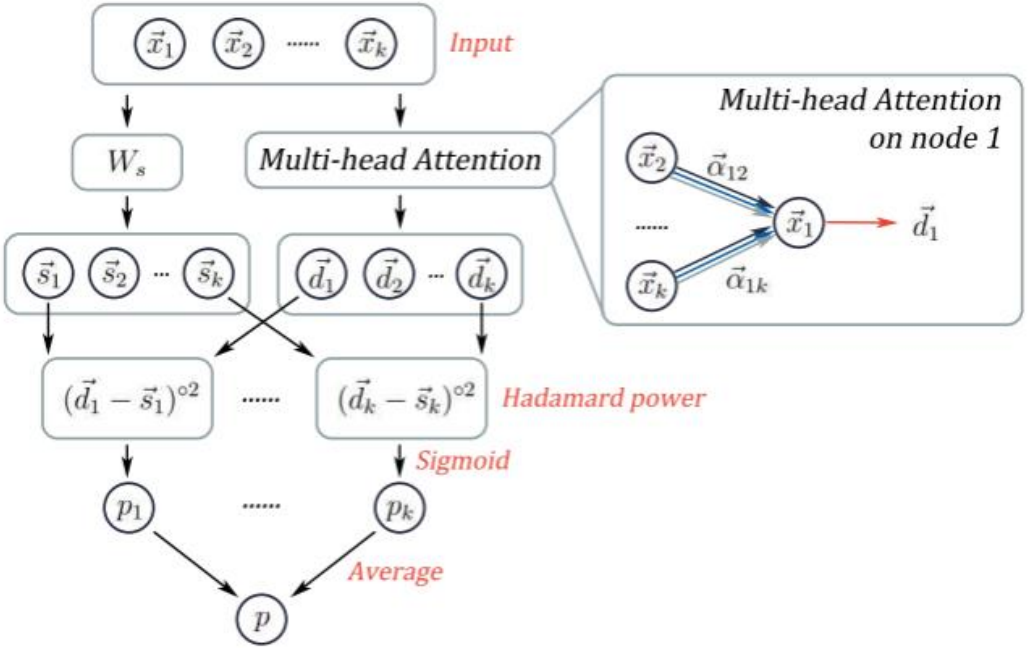
For all scHi-C datasets, we kept only the cells with more than 2,000 read pairs that have genomic span greater than 500 Kb. At a given resolution, we define the number of contacts per cell as the number of interaction pairs (read count) assigned to the non-diagonal entries of the intra-chromosomal contact maps.

Hypergraph NN architecture in Higashi

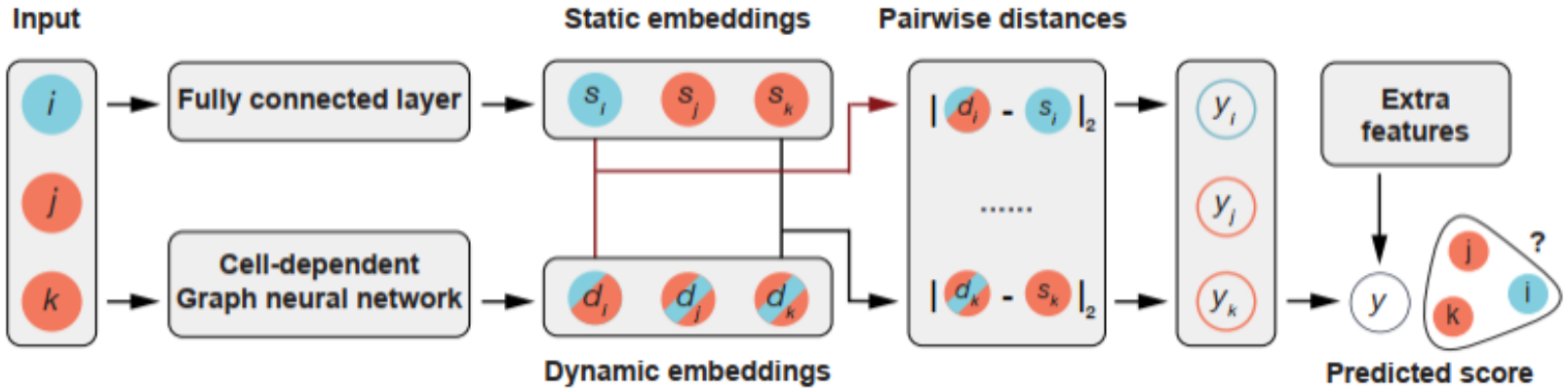
Hyper-SAGNN

The model aims to predict the value of an entry (that is, contact frequency) in an scHi-C contact map using the rest of the contact map as input. The model also has the option to use the contact maps from cells that share similar 3D genome structures (that is, close to each other in the embedding space) as auxiliary information for the prediction as well.





Hypergraph NN architecture in Higashi

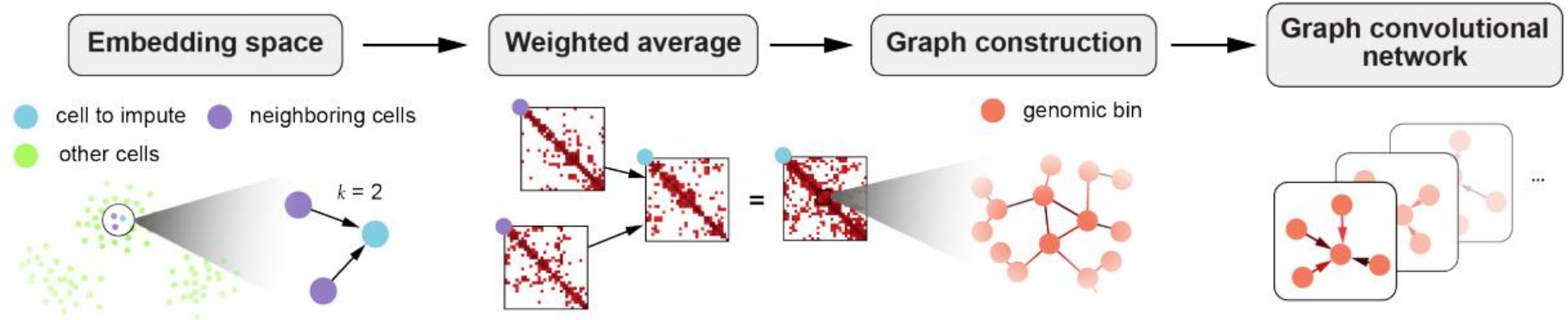


The structure of the hypergraph neural network

The input triplet consisting of one cell node and two bin nodes passes through two branches of the network to generate static embeddings and dynamic embeddings for each node, respectively. Then the pairwise distances between static and dynamic embedding pairs are calculated. These pairwise distances are combined with extra features such as genomic distance between the two bins to produce the final predicted score for the input triplet, which represents the probability of an entry in the single-cell contact map.

Controlling information transferred across cells

- Hyper-parameter k
- For $cell_i$, its k-nearest neighbors in the embedding space would contribute
- Graph convolutional network learns the embeddings for bin nodes
- k = 4 for all results in this presentation



Loss function and training details of Higashi

The hypergraph NN in Higashi produces a score y^ for any triplet (ci, bj, bk). The NN is trained to minimize the difference between the predicted score y^ and the target score y (that is, the observations in the dataset), indicating the probability of the pairwise interaction between bin nodes bj and bk in cell ci. In Higashi, we offer several choices of loss function for scHi-C datasets with different coverage:

Binary classification loss

For scHi-C datasets with relatively low sequencing depths, or the analysis resolution is high (hence, fewer reads in each genomic bin), the model is trained with a binary classification loss (cross-entropy) where the triplets corresponding to all non-zero entries in the single-cell contact maps are treated as positive samples, and the rest are considered as the negative samples (that is, y(ci, bj, bk)∈{0, 1}).

$$\mathrm{Loss}_{\mathrm{class}} = -\sum_{i,j,k} y(c_i, b_j, b_k)\log \hat{y}(c_i, b_j, b_k) + \left[1 - y(c_i, b_j, b_k)\right] \log \left[1 - \hat{y}(c_i, b_j, b_k)\right]$$

Ranking loss

For datasets with relatively high sequencing depths or when the analysis resolution is low (hence, more reads in each genomic bin), we further differentiate among the non-zero values by training the model with a ranking loss, which maintains consistent ranking of predicted scores versus the continuous target scores (that is, y(ci, bj, bk) ∈ R)

$$l_{ij} = \mathbb{I}\left[y(t_i) > y(t_j)\right]$$

$$p_{ij} = \mathrm{Sigmoid}\left[\hat{y}(t_i) - \hat{y}(t_j)\right]$$

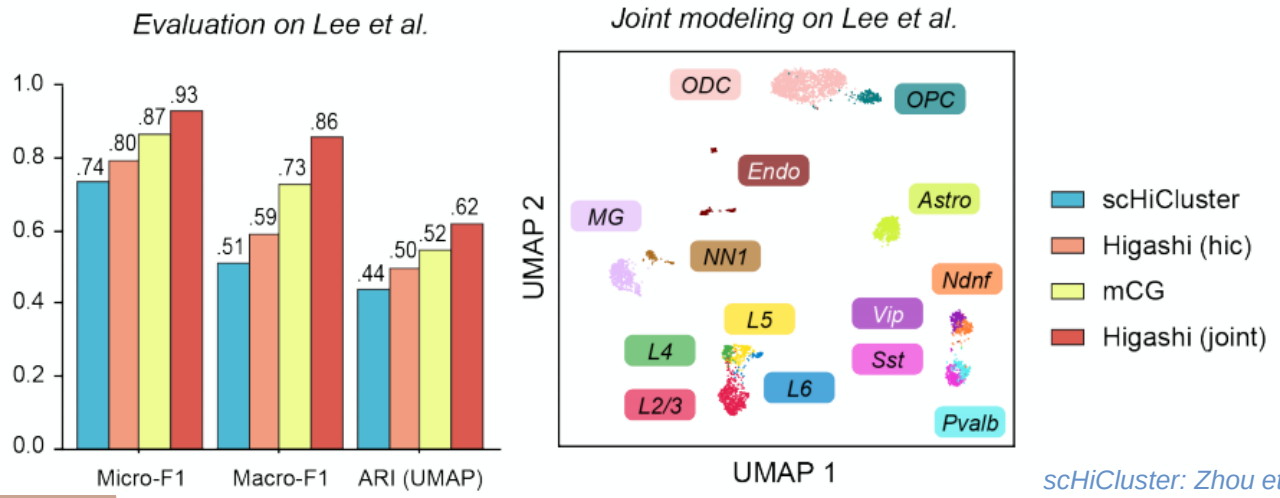
$$\mathrm{Loss}_{\mathrm{rank}} = -\sum_{|y(t_i)-y(t_j)|\geq\alpha} l_{ij}\log p_{ij} + (1 - l_{ij})\log\left(1 - p_{ij}\right)$$

03

Results

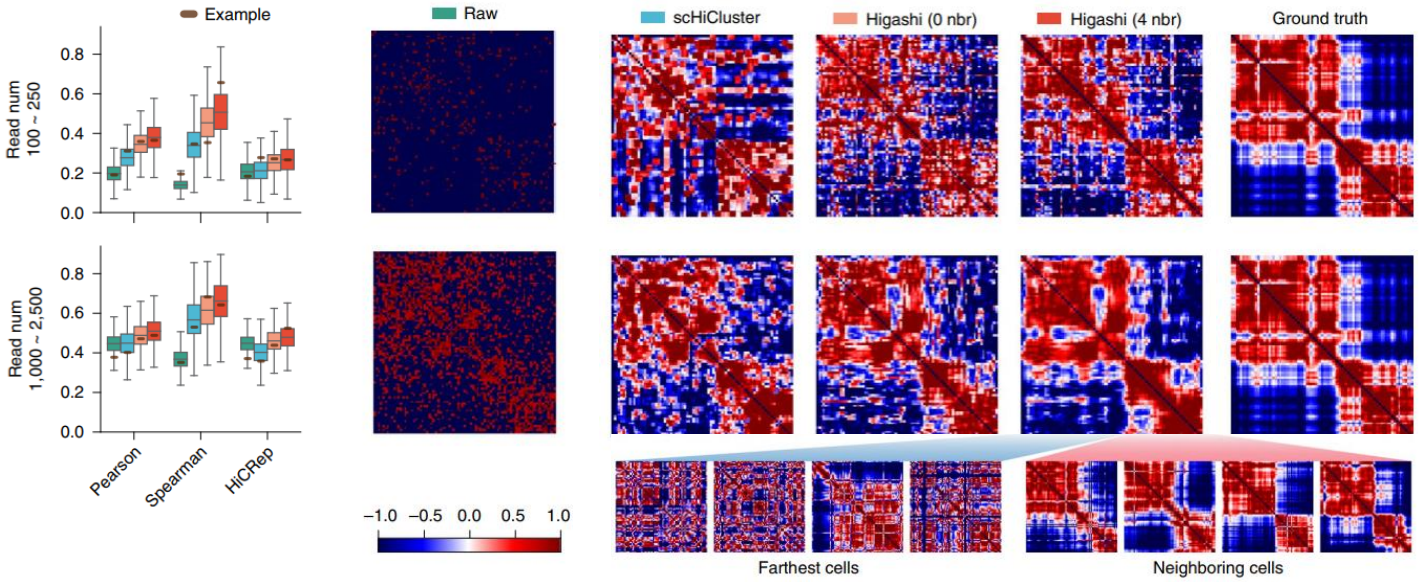
Embeddings from Higashi reflect cell identity information

- Evaluation on public scHi-C datasets
  - Logistic regression + Micro-F1 / Macro-F1
  - K-means clustering + ARI (adjusted rand index)
- Higashi outperforms existing methods for identifying cell types / cellular states
- Joint modeling of co-assayed signals further improves the performance



scHiCluster: Zhou et al., PNAS, 2019

Higashi robustly imputes scHi-C contact maps

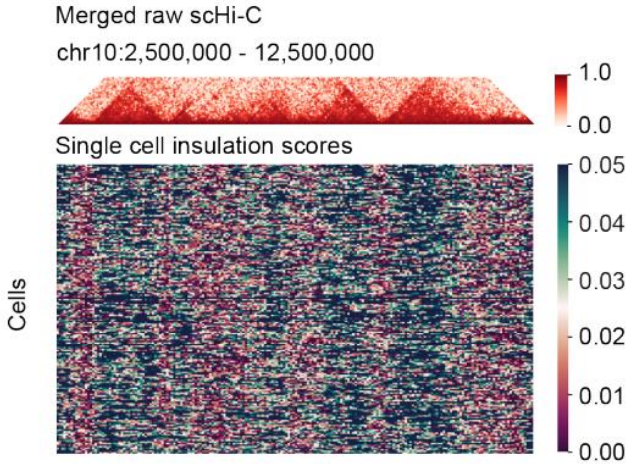


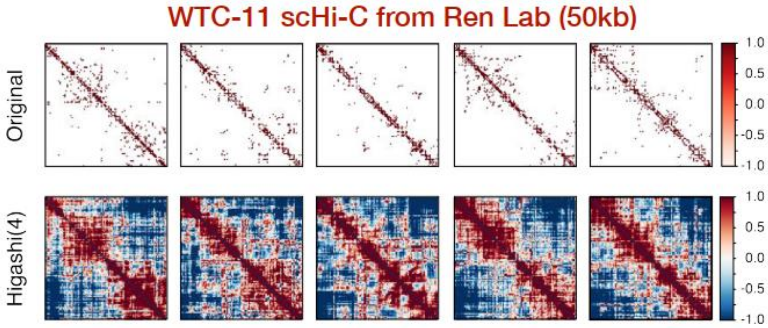
- Simulating scHi-C from multiplexed 3D genome imaging data *Bintu et al. Science, 2018*
- Higashi (0 nbr) already outperforms baseline methods
- Higashi (4 nbr) further improves the imputation accuracy
- Similar conclusion with simulation from Su et al. Cell, 2020

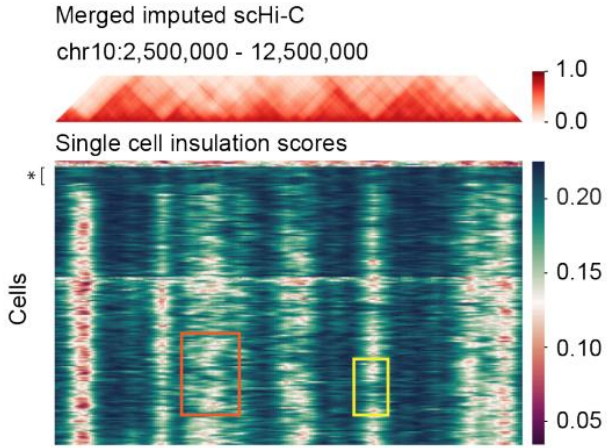
Single cell TAD-like domain boundary identification

Variability of TAD boundaries:

- On/off
- Sliding along the genome

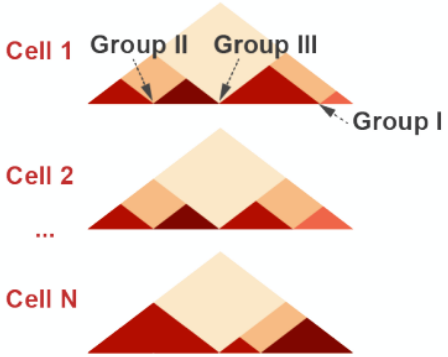


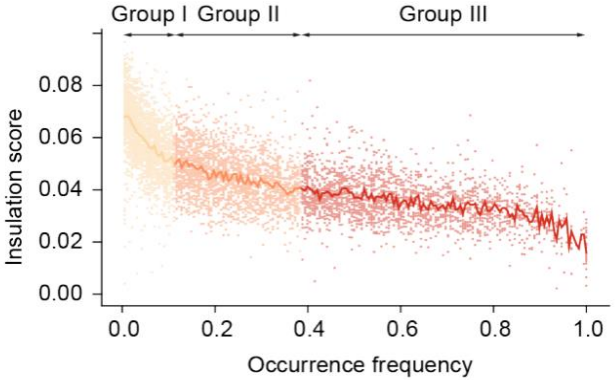


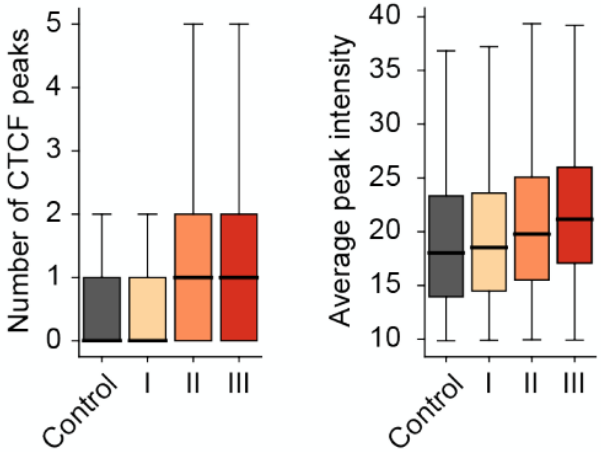


Properties of the single cell TAD-like boundaries

- Each element corresponds to a single-cell domain boundary
- TAD boundaries with higher occurrence frequencies
  - Lower single cell insulation scores
  - More CTCF peaks
  - Higher average CTCF peak intensity

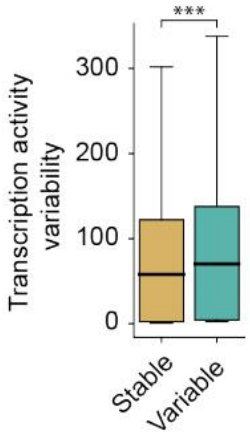


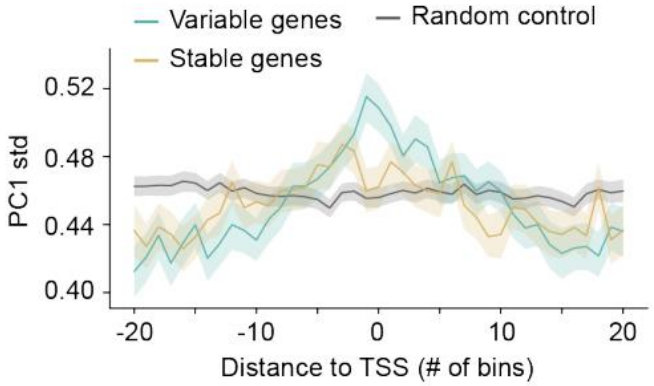


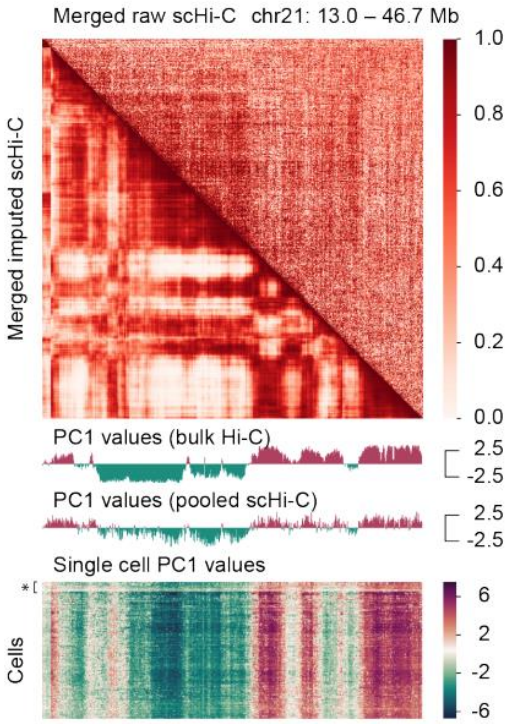


Single cell A/B compartment variability

- Bulk projection matrix for single cell Hi-C PCA (scA/B)
- Variability of A/B compartment correlates with the variability of transcription activity







WTC-11 scHi-C from Ren Lab (50kb)

04

Our Reproduction

Environment settings

Platform selection

We tested several development environments:Win10、 WSL、VM(ubuntu18.04):

- **Win10**: cython & cooler problem → need mingw or visualstudio ❌
- **WSL**: Cuda is difficult to set up. ❌(15 hour)
- **VM(ubuntu18.04)**: CPU is too slow and Our GPU is run out of memory ❌
- **Colab**: The latest pytorch environment has been installed and Powerful GPU 👍(4 hour)

```
Wed Jun 15 02:55:06 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   35C    P8    11W /  70W |      3MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

GPU in colab (random):

R-80, K80, T4, P100

What we use:

P100

Input data format

Files we need

- **data.txt:** input data
- **config_ramani.JSON:** Used to set all the parameters used by all the different tasks (more..)
- **label_info.pickle:** label
- **hg19.chrom.sizes.txt:** genome reference file from USCS Genome Browser, will be used to generate bin nodes.
- **cytoBand_hg19.txt:** cytoband reference file from USCS Genome Browser, will be used to remove centromere regions.

Input data format

GSE84920(NCBI)

1.GSM2254215_ML1.validPairs.txt(4.47G)

2.GSM2254217_ML3.validPairs.txt(860MB)

Part of the ML1 file 👉

mouse_ch3	22801139	22801196	mouse_ch9	53295578	53295687	D00584:136:HMTLJBCXX:1:1101:10000:101176	37	42	+	-	ATCCGCGG	GTCATAGT	HIC_mouse_ch3_53191_0		HIC_mouse_ch9_128673	4		
human_chr2	88637036	88637116	human_chr2	89109729	89109866	D00584:136:HMTLJBCXX:1:1101:10000:12410	42	42	-	-	GAGGAGCA	CGATGACA	HIC_human_chr2_217931	5	HIC_human_chr2_219230	5		
human_chr4	126924974	126925025	human_chr4	127334997	127335124	D00584:136:HMTLJBCXX:1:1101:10000:15521	42	42	+	+	GCTACGGT	AGTCGTAT	HIC_human_chr4_289987	0	HIC_human_chr4_290886	0		

Data.txt

Process GSE84920 file to fit "higashi_v1" format

Data.txt(863MB)

Part of the Data.txt file 👉

data.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

cell_name	cell_id	chrom1	pos1	chrom2	pos2	count
ML1_GAGGAGCA_CGATGACA	0	chr2	88637036	chr2	89109729	1
ML1_GCTACGGT_AGTCGTAT	1	chr4	126924974	chr4	127334997	1
ML1_AGGTGCGA_ATACATGT	2	chr15	42130039	chr15	42677209	1
ML1_GCCTCGAA_GAGTACGT	3	chr1	209393848	chr1	232468122	1

https://github.com/ma-compbio/Higashi/wiki/Input-Files

Input data format

Configure the parameters

All customizable parameters are stored in a JSON config file. The path to this JSON config file will be needed when running the program.

For examples of the configuration JSON file ☞

Important parameters

- **data_dir:** Path to the folder that store data.txt
- **genome_reference_path:** Path to the genome reference file
- **cytoband_path:** Path to the cytoband reference file

```
{
    "config_name": "ramani et. al",
    "data_dir": "Ramani",
    "temp_dir": "Ramani/temp",
    "genome_reference_path": "Ramani/hg19.chrom.sizes",
    "cytoband_path": "Ramani/cytoBand_hg19.txt",
    "chrom_list": ["chr1","chr2","chr3","chr4","chr5",
                   "chr6","chr7","chr8","chr9","chr10",
                   "chr11","chr12","chr13","chr14","chr15",
                   "chr16","chr17","chr18","chr19","chr20",
                   "chr21","chr22","chrX"],
    "resolution": 1000000,
    "resolution_cell": 1000000,
    "minimum_distance": 2000000,
    "maximum_distance": -1,
    "local_transfer_range": 1,
    "dimensions": 64,
    "impute_list":["chr1","chr2","chr3","chr4","chr5",
                   "chr6","chr7","chr8","chr9","chr10",
                   "chr11","chr12","chr13","chr14","chr15",
                   "chr16","chr17","chr18","chr19","chr20",
                   "chr21","chr22","chrX"],
    "minimum_impute_distance": 0,
    "maximum_impute_distance": -1,
    "neighbor_num": 5,
    "plot_start": 0,
    "plot_end": -1,
    "plot_label": ["cell type"],
    "call_tads": false,
    "embedding_name": "exp_zinb3",
    "cpu_num": -1,
    "gpu_num": 1,
    "optional_smooth": false,
    "optional_quantile": false,
    "rank_thres": 1,
    "loss_mode": "zinb",
    "random_walk": false,
        "UMAP_params": {"n_neighbors": 20
    }
}
```

[Configure the parameters · ma-compbio/Higashi Wiki (github.com)](https://github.com/ma-compbio/Higashi/wiki/Configure-the-parameters)

Input data format

Why we need genome reference ?

As the cost of DNA sequencing falls, and new full genome sequencing technologies emerge, more genome sequences continue to be generated. Reference genomes are typically used as a guide on which new genomes are built, enabling them to be assembled much more quickly and cheaply than the initial Human Genome Project.

hg19.chrom.sizes.txt

chr	size
chr1	249250621
chr2	243199373
chr3	198022430
chr4	191154276
chr5	180915260
chr6	171115067
chr7	159138663
chrX	155270560
chr8	146364022
chr9	141213431
chr10	135534747
chr11	135006516
chr12	133851895
chr13	115169878
chr14	107349540
chr15	102531392
chr16	90354753
chr17	81195210
chr18	78077248
chr20	63025520
chrY	59373566
chr19	59128983
chr22	51304566
chr21	48129895
chr6_ssto_hap7	4928567
chr6_mcf_hap5	4833398
chr6_cox_hap2	4795371
chr6_mann_hap4	4683263

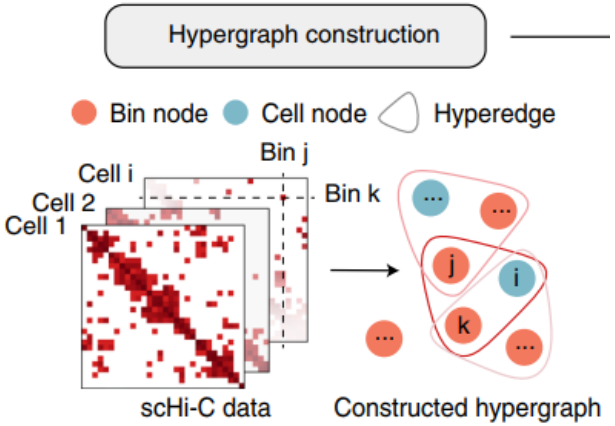
cytoBand_hg19.txt:

	chr	start	end	stain	band
1	chr1	0	2300000	p36.33	gneg
2	chr1	2300000	5400000	p36.32	gpos25
3	chr1	5400000	7200000	p36.31	gneg
4	chr1	7200000	9200000	p36.23	gpos25
5	chr1	9200000	12700000	p36.22	gneg
6	chr1	12700000	16200000	p36.21	gpos50
7	chr1	16200000	20400000	p36.13	gneg
8	chr1	20400000	23900000	p36.12	gpos25
9	chr1	23900000	28000000	p36.11	gneg
10	chr1	28000000	30200000	p35.3	gpos25
11	chr1	30200000	32400000	p35.2	gneg
12	chr1	32400000	34600000	p35.1	gpos25
13	chr1	34600000	40100000	p34.3	gneg
14	chr1	40100000	44100000	p34.2	gpos25
15	chr1	44100000	46800000	p34.1	gneg
16	chr1	46800000	50700000	p33	gpos75
17	chr1	50700000	56100000	p32.3	gneg
18	chr1	56100000	59000000	p32.2	gpos50
19	chr1	59000000	61300000	p32.1	gneg
20	chr1	61300000	68900000	p31.3	gpos50
21	chr1	68900000	69700000	p31.2	gneg
22	chr1	69700000	84900000	p31.1	gpos100
23	chr1	84900000	88400000	p22.3	gneg
24	chr1	88400000	92000000	p22.2	gpos75
25	chr1	92000000	94700000	p22.1	gneg
26	chr1	94700000	99700000	p21.3	gpos75

refer: Reference genome - Wikipedia

Data processing

1. generate a dictionary that'll map genomic bin loci to the node id.
2. extract data from the data.txt and turn that into the format of hyperedges (triplets)
3. create contact maps based on sparse scHi-C for visualization, baseline model, and generate node attributes

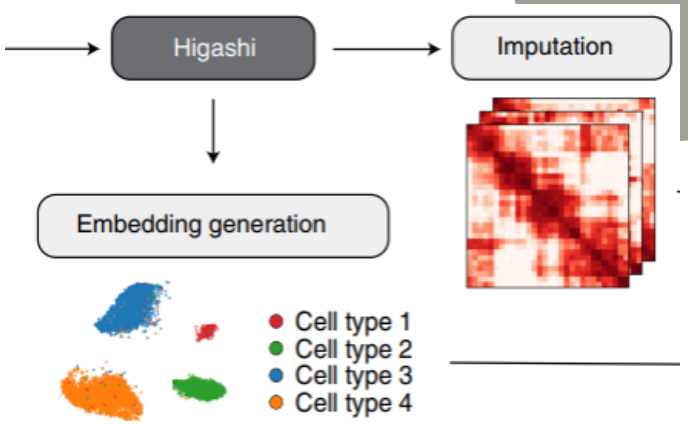


Train the Higashi model

1. Train Higashi without cell-dependent GNN to force self-attention layers to capture the heterogeneity of chromatin structures.(60 epoch)(we use 5)
2. Train Higashi with cell-dependent GNN, but with k=0.(45 epoch)
3. Train Higashi with cell-dependent GNN, but with k=`neighbor_num` in the config JSON.(30 epoch)

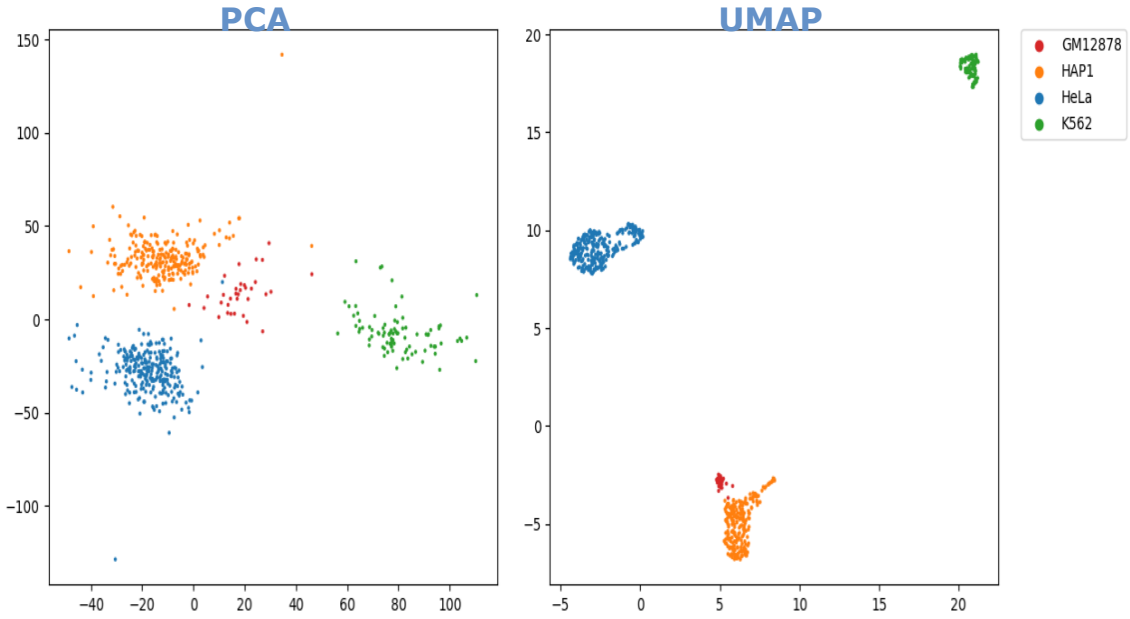
(we use 5)

```
First stage training
update_rate: 0.000000   0.000000
[ Epoch 0 of 60 ]
  - (Training)   bce:  0.6349, mse:  0.6636,  acc: 33.545 %, pearson: 0.255, spearman: 0.235, elapse: 120.431 s
  - (Validation-hyper) bce:  0.5863,  acc: 44.694 %,pearson: 0.369, spearman: 0.381,elapse: 0.144 s
pair_ratio 0.1
no improve 0
update_rate: 0.132628   1.071344
[ Epoch 1 of 60 ]
  - (Training)   bce:  0.5749, mse:  0.6246,  acc: 49.048 %, pearson: 0.388, spearman: 0.381, elapse: 116.830 s
  - (Validation-hyper) bce:  0.5596,  acc: 55.866 %,pearson: 0.436, spearman: 0.425,elapse: 0.124 s
pair_ratio 0.2
no improve 0
update_rate: 0.036743   0.228028
```

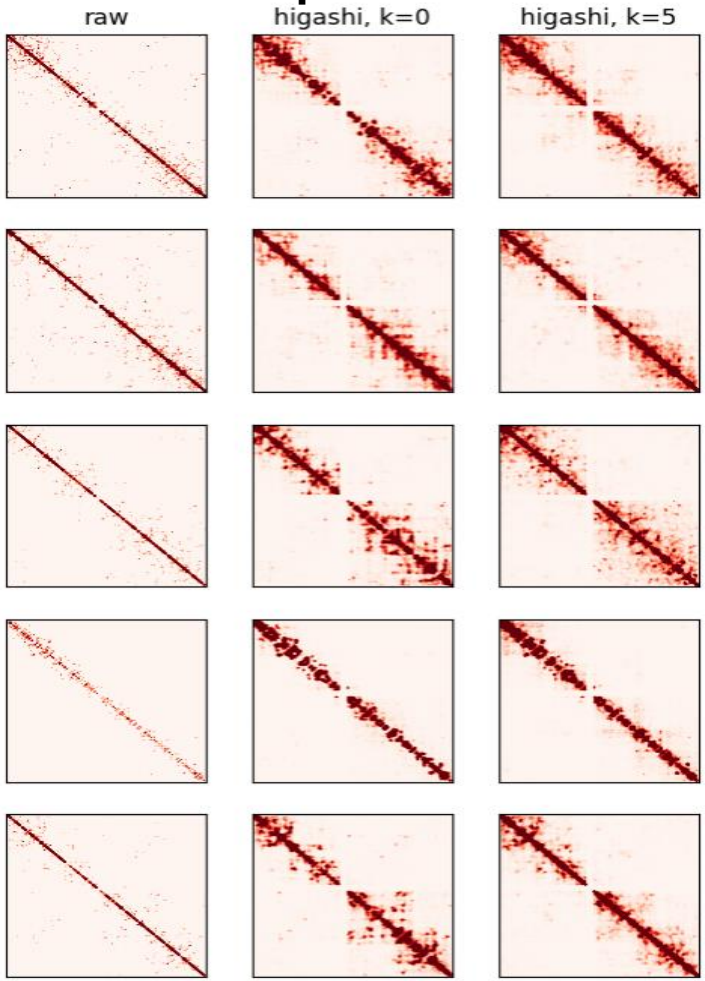


Our result

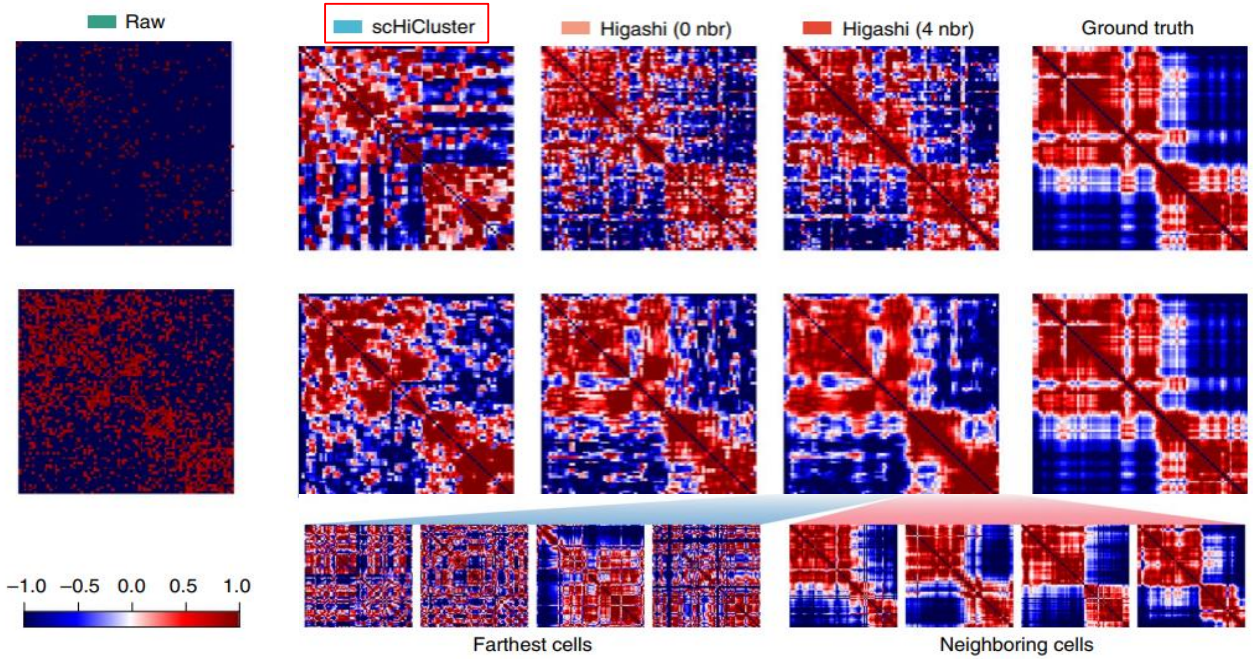
embedding



Imputation



Our target: Higashi robustly imputes scHi-C contact maps



擷取至: [Multiscale and integrative single-cell Hi-C analysis - YouTube]

Thanks for listening