About the assignment: This assignment has been designed to help you think critically about the topics we covered in the class, encourage you to look at the literature, and get familiar with applying deep learning frameworks on Hi-C matrices.

For the conceptual questions, we are looking for answers that are maximum 3 lines per point. So if a question is worth 2 points try answering it in maximum 6 lines. When writing answers, try to first list the main idea addressing the question and then expand on it. You may refer to the papers in the Reference section while answering these questions.

For programming assignment, we will run your code to check if it gives the correct output. If the code does not run successfully, we will assign partial scores to the correct logic behind the implementation.

Attempting bonus questions or tasks is encouraged but not required.

Total: [45 points + 3 bonus points]

1 Conceptual Questions

[30 points + 3 bonus points]

Question 1: (6 points)

(b) [2 points] Why do you think using only DNA sequences as input to predict gene expression for different cell lines might be inadequate?
(c) [2 points] What is the potential advantage of predicting gene expression from histone modifications or, more specifically, modelling this relationship?
(d) [2 points] What properties of a deep learning framework, like DeepChrome [1], allow it to be advantageous over previous strategies (using Linear regression, etc.) when classifying gene expression?

Question 2: (4 points)

What changes would you make to the following to convert the classification task of DeepChrome [1] to a regression task (predicting gene expression)?
(a) [1 point] Output
(b) [1 point] Final layer
(c) [1 point] Loss function
(d) [1 point] Evaluation metric to report performance (as used in previous works)

Question 3: (4 points)

(a) [2 points] Does the architecture of DeepChrome [1] allow it to preserve the position information of histone modification signals when making final classification? Why or why not? [Hint: Think about different CNN layers]
(b) [2 points] Can another deep learning architecture help preserve the position information? Why or why not?
**Question 4:** (6 points)

(a) [2 points] What model architecture considerations lead to the splitting of Hi-C maps into smaller input patches for HiCPlus [2]?

(b) [2 points] What assumptions can we make for the Hi-C data to allow treating these patches as images?

(c) [2 points] What information might we be losing by processing the data this way?

**Question 5:** (10 points)

Compare and contrast HiCPlus [2] or HiCGAN [3] for the following:

(a) [2 points] Model architecture

(b) [2 points] Loss Function

(c) [2 points] Evaluation metric

(d) [2 points] Performance

(e) [2 points] Biological insights from results

**Question 6 (bonus):** (3 points)

Can you propose an experiment to establish that models like HiCPlus [2] or HiCGAN [3] can be applied to the task of improving HiC-data resolution in a real-world setting? By real-world setting, we mean that the low-resolution map is not artificially simulated by downsampling high-resolution data.

## 2 Programming Assignment

[15 points]

**Background:** Studying the three-dimensional (3D) organization of the human genome is vital for understanding cellular functions. The spatial organization of the genome can directly or indirectly affect the regulation of genes, and in turn, can decide the fate of the cell. Various high-throughput experimental techniques, such as Hi-C, are used to study higher-order chromatin structure at different scales. The Hi-C assay uses high-throughput sequencing to measure 3D genome structure, where each read pair corresponds to an observed 3D contact between two genomic loci. Data from a Hi-C assay is typically coalesced into a matrix in which rows and columns correspond to fixed-width windows (“bins”) tiled along the genomic axis, and values in the matrix are counts of read pairs that fall into the corresponding bins.

Due to the high sequencing costs, Hi-C experiments can result in low read coverage and high data sparsity. To analyze such datasets, researchers use large (100kb) fixed-width bin sizes to reduce noise in the data. While this approach may give insights into the global interactions within and among chromosomes, it is hard to locate finer interactions among regulatory elements of the DNA. In order to improve the Hi-C resolution for better downstream analysis, multiple studies have proposed the use of deep neural networks to predict high-resolution Hi-C maps from low-resolution ones.

**Dataset:** The provided dataset was used by Zhang et al. [2] for the development of HiCPlus. The dataset consists of Hi-C data obtained for the cell line GM12878 (lymphoblastoid cells). The training outputs are patches or sub-matrices (of size $40 \times 40$) of an experimentally-obtained high-resolution Hi-C matrix for chromosomes 19-22. The training inputs were generated by randomly down-sampling this high-resolution Hi-C matrix and splitting the matrix into $40 \times 40$ patches. The test inputs were similarly generated by randomly down-sampling an experimentally-obtained high-resolution Hi-C matrix for chromosome 17 and splitting this matrix into $40 \times 40$ patches.

**Task [15 points]:**

Given the skeleton code (`hw2_stencil.py`):

1. [6 points] Implement a Convolutional Neural Network (CNN) to transform patches of low-resolution Hi-C data (training inputs) into patches of high-resolution Hi-C data (training outputs). Your CNN should use valid padding, so it should reduce the size of the input matrix patches. This means that you will
Homework Assignment #2

need to design your model architecture (numbers and types of layers) and hyperparameters (filter sizes, layer sizes, etc.) to produce accurate transformations without reducing the size of the matrix patches too much. Additionally, you will need to remove the borders of the training outputs to ensure that the model outputs and training outputs have the same size. For example, the architecture of HiCPlus reduces the size of the input matrix patches from $40 \times 40$ to $28 \times 28$, so a border of size 6 would need to be removed from the training outputs to ensure that the training outputs and model outputs both have size $28 \times 28$. To train the model, you should use a cross-validation scheme with a mean squared error loss function.

2. [3 points] Describe how you used cross-validation to tune the model’s hyperparameters.

3. [3 points] Report the final mean squared error and Pearson correlation coefficient of the finalized model averaged over the entire training set. Based on these quantitative metrics, how well does the model seem to be performing on the training set? If you plot the actual high-resolution Hi-C patch output and the predicted Hi-C patch, do they look comparable and verify the quantitative performance of the model?

4. [3 points] For several matrix patches in the test set, visualize and compare the training input and predicted label. Based on this qualitative assessment, how well does the model seem to be performing on the test set? Include the visualizations for 3 example matrix patches that appear to be transformed accurately, plotting the original low-resolution input Hi-C patch on the left and the predicted high-resolution Hi-C patch on the right.

References


Course policies:

Collaboration Policy: Discussion of material with your classmates is both permitted and encouraged. However, showing, copying or other sharing of answers to written questions for this assignment is forbidden.

Missed assignments (including late assignments): You can get a 3-day extension for at most 3 deadlines without penalty. Excluding the scenario mentioned above, 20% of the total points will be deducted for late submissions and missed submissions won’t be assigned any score.