STA496: Readings in Statistics Report

Rogers Yang

Supervisor: Prof. Elena Tuzhilina

March 31, 2025

Contents

1 Introduction

The three-dimensional organization of the genome plays a critical role in regulating gene expression and maintaining cellular function. However, the intricate architecture of chromatin makes direct spatial observation infeasible. Advances in high-throughput techniques—most notably Hi-C—have enabled researchers to reconstruct chromatin structure indirectly by capturing contact frequencies between genomic loci. These contact maps serve as a statistical and mathematical proxy for the underlying spatial organization of DNA within the nucleus.

Motivated by the need to understand how chromatin organization differs among cell types, this study leverages single-cell Hi-C data to uncover patterns and structural variations across four distinct cell lines: *Hela*, *GM12878*, *HAP1*, and *K562*. The central goal is to employ explainable statistical methods to identify and characterize the chromatin interaction patterns that differentiate these cells. By doing so, we aim to contribute insights into the relationship between genome architecture and cellular identity.

Our approach integrates a range of methodologies drawn from both classical statistical learning and modern computational biology. Foundational concepts from texts such as *Introduction to Statistical Learning* (with particular emphasis on resampling methods, unsupervised learning via principal component analysis, and clustering techniques) underpin our analysis. In addition, we critically review several key papers that have advanced the field of single-cell Hi-C analysis.

The remainder of this report documents the literature review, mathematical derivations related to these methods (detailed in the Appendix), and our experimental analyses conducted on the Ramani dataset [4]. By synthesizing advanced statistical learning techniques with high-dimensional genomic

data, we seek to elucidate the structural underpinnings that distinguish cell types, thereby enhancing our understanding of chromatin architecture in a cellular context.

2 Literature Review

Ramani introduced the sciHi-C method, which employs combinatorial indexing for high-throughput single-cell Hi-C analysis [5]. Their paper motivated our work by detailing a robust preprocessing pipeline, filtering long-range intrachromosomal contacts (i.e., > 20 kb) and excluding self-interactions, to generate high-quality contact matrices. This method informed our own data-loading procedures and matrix alignment strategies.

Liu proposed an unsupervised embedding approach for scHi-C data by combining specialized distance measures with multidimensional scaling (MDS) [2]. They evaluated several metrics—including CDP-JSD, HiCRep (using stratum-adjusted correlation), GenomeDISCO, and HiC-Spector—to capture biologically meaningful variations in chromatin structure. Our interest in this paper stemmed from its emphasis on choosing appropriate distance measures, which is crucial for reliable downstream analysis.

Kim applied Latent Dirichlet Allocation (LDA) to single-cell Hi-C data, treating each cell as a "document" and each chromatin interaction as a "word." [1] This cross-disciplinary approach from natural language processing effectively extracts latent topics, revealing cell type-specific interaction patterns. The innovative use of LDA to overcome data sparsity and uncover underlying chromatin structures provided key insights that influenced our exploration of unsupervised methods for genomic data.

Recent advances in single-cell Hi-C (scHi-C) technologies have enabled unprecedented exploration of cell-to-cell variability in 3D chromatin organization, offering insights into its role in gene regulation and cellular identity [6]. However, the extreme sparsity (0.25–1% of contacts captured) and technical heterogeneity of scHi-C data pose significant challenges for computational analysis. Current methods, such as Higashi [6], scHiCluster [7], and scDEC-Hi-C [3], address these limitations through diverse strategies: Higashi employs hypergraph representation learning to impute sparse contact maps and integrate multimodal epigenomic signals, while scHiCluster applies convolution- and random walk-based imputation to enhance clustering accuracy. Deep generative models like scDEC-Hi-C further unify embedding and clustering tasks in an end-to-end framework. Despite these advances, critical gaps persist. First, many methods rely on computationally intensive imputation (e.g., hypergraphs or random walks) that may not scale efficiently for high-resolution data [6, 7]. Second, most frameworks treat embedding and clustering as separate steps, potentially compromising performance [3]. Third, coverage heterogeneity and multiscale 3D feature variability (e.g., compartments, TAD-like domains) remain inadequately addressed, limiting robust cell-type identification.

To overcome these limitations, we propose a streamlined workflow combining low-rank approximation, nonlinear dimensionality reduction, and clustering. Low-rank approximation offers a computationally efficient alternative to existing imputation methods by denoising sparse matrices while preserving structural patterns. Subsequent integration of UMAP—a nonlinear embedding technique—may better capture chromatin interaction dynamics compared to linear methods like PCA. By unifying these steps into a single pipeline, our approach aims to mitigate technical biases, enhance scalability, and improve clustering accuracy for scHi-C data. This strategy addresses unmet needs in the field, particularly for studies prioritizing interpretability and efficiency in analyzing complex 3D genome architectures.

3 Methods

The study workflow is summarized as shown in Figure 1. The details are provided in the sections 3.1, 3.2, 3.3, 3.4.

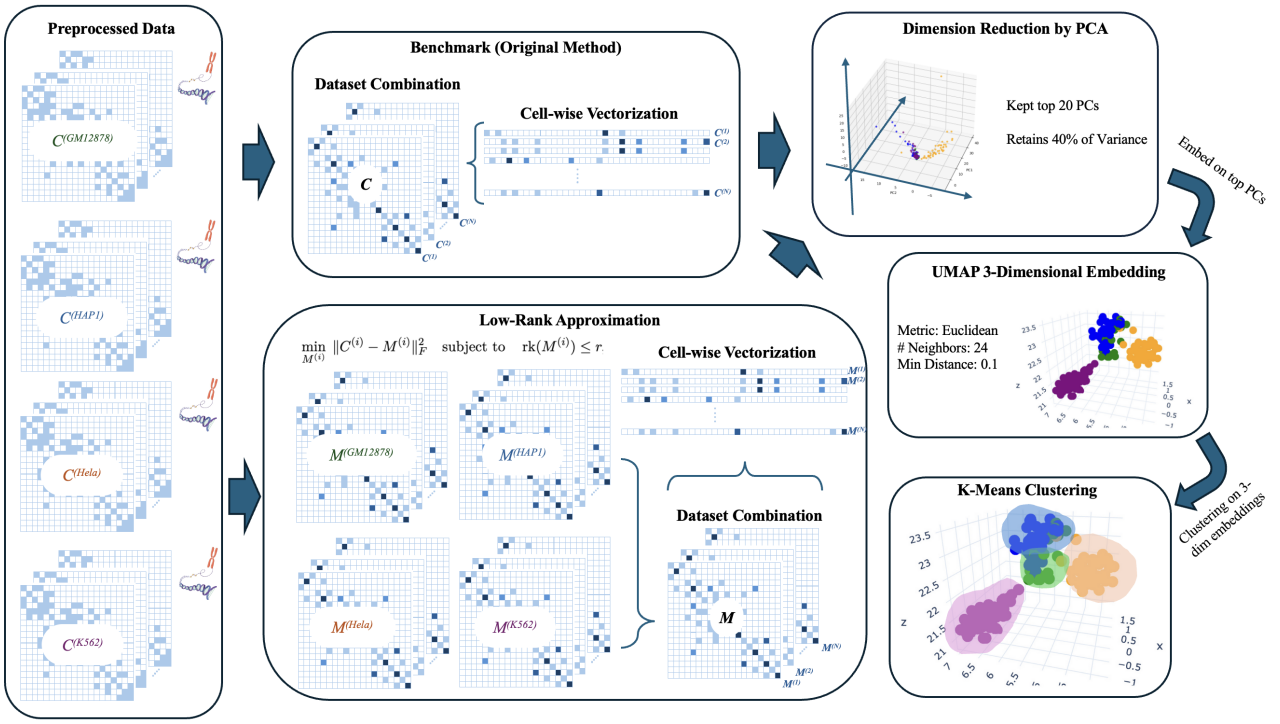


Figure 1: Summary of the study workflow

3.1 Data Preprocessing and Matrix Construction

The data used in this study were obtained from [4]. The dataset comprises four cell types with markedly different sample sizes, as illustrated in Figure 2. There are total of $N = 2611$ sample of cells included in this study, containing 1622 *Hela* cells (62.12% of sample), 917 *HAP1* cells (35.12%), 48 *K562* cells (1.84%), and 24 *GM12878* cells (0.92%).

For each selected cell, contact information is recorded based on the genomic positions associated with each interaction (e.g., chromosome 1 at location 1, chromosome 2 at location 2, etc.). Only intrachromosomal contacts are included in the dataset, where all the interactions between different chromosomes are ignored. These contacts are then aligned to generate a binary matrix representation of fixed dimensions. Specifically, cell is denoted by C and is represented by a 2965×2965 matrix, where each entry is defined as

$$C_{ij} = \begin{cases} 1, & \text{if a contact between loci } i \text{ and } j \text{ is observed,} \\ 0, & \text{otherwise.} \end{cases}$$

To mitigate biases arising from self-contacts, the main diagonal entries (which represent self-contacts) are set to zero.

Each contact matrix is structured as a block-diagonal matrix composed of 23 chromosome-specific submatrices:

$$C = \text{diag}(C_1, C_2, \ldots, C_{22}, C_X),$$

where each block C_j corresponds to the contact matrix for chromosome j. Further, we denote the ith sample cell as $C^{(i)} = \text{diag}(C_1^{(i)}, C_2^{(i)}, \ldots, C_{22}^{(i)}, C_X^{(i)})$.

The 2965 in size is determined by the total sum of the number of genomic loci of each chromosome, which is listed in Figure 3. For instance, chromosome 1 is represented as C_1 with the size of 250 by 250, and chromosome 22 is C_{22} with the size of 36 by 36. Notably, chromosomes appearing earlier in the cell's sequence are generally larger, with the exception of chromosome X. Following the approach described in Ramani's work, cells with fewer than 1000 unique contacts were excluded to ensure data quality.

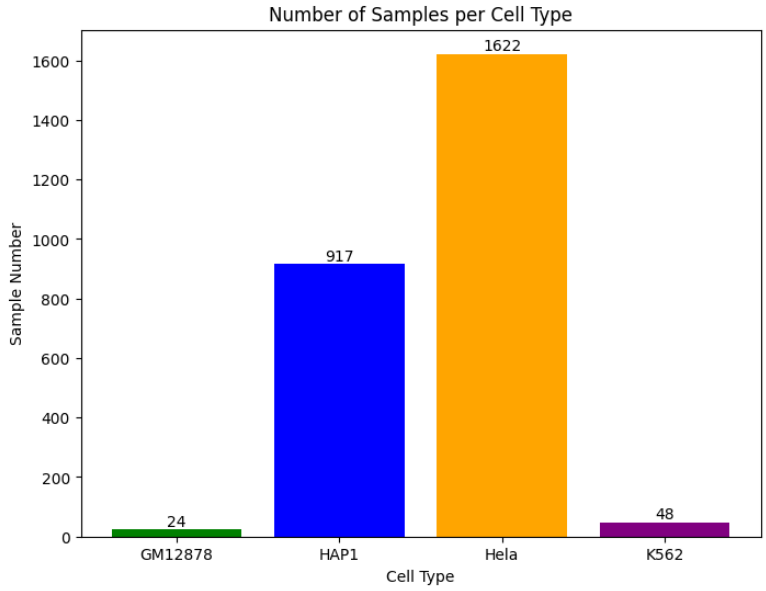


Figure 2: Counts comparison for different cell types in dataset

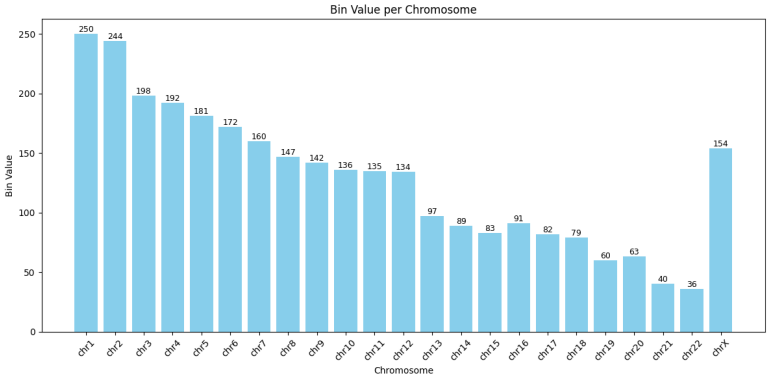


Figure 3: Chromosome Sizes Comparison

3.2 Low-Rank Approximation of Contact Matrices

Due to the high dimensionality and inherent noise in the raw binary matrices, direct analysis is computationally inefficient and may lead to spurious results. Therefore, we performed a low-rank approximation to extract the most informative features of each matrix.

For each cell matrix $C^{(i)}$, we compute its low-rank approximation $M^{(i)}$ by solving the optimization problem:

$$\min_{M^{(i)}} \|C^{(i)} - M^{(i)}\|_F^2 \quad \text{subject to} \quad \text{rk}(M^{(i)}) \leq r,$$

where $\|\cdot\|_F$ denotes the Frobenius norm and r is a predetermined rank that balances noise reduction and information preservation.

3.3 Dimensionality Reduction and Visualization

To facilitate the identification of structural disparities among cells, each denoised matrix M is vectorized by stacking its columns to form a high-dimensional feature vector. Principal Component Analysis (PCA) is then applied to these vectors. PCA projects the data into a lower-dimensional space by capturing the directions of maximal variance, which not only aids in visualization but also highlights the genomic regions contributing most significantly to the observed differences. Detailed steps are included in Appendix 5.

Subsequently, the PCA loadings are used as inputs for Uniform Manifold Approximation and Projection (UMAP). UMAP is employed to embed the cells into a three-dimensional space, providing a clear visual representation of potential clusters. These clusters are then analyzed to assess whether they correspond to known or biologically distinct cell types, thereby validating the effectiveness of the preprocessing and low-rank approximation steps.

Overall, this pipeline, from raw data processing through low-rank approximation and dimensionality reduction, enables robust and interpretable analysis of single-cell Hi-C data.

3.4 Clustering and Accuracy Assessment

For each cell sample, we first processed the data using a series of dimensionality reduction techniques, starting with a low-rank approximation, followed by PCA, and finally, UMAP, to obtain a three-dimensional embedding $\mathbf{X} \in \mathbb{R}^{N\times 3}$ (where N is the number of cells). K-means clustering (with $k = 4$) was then applied to the rows of $\mathbf{X}$ to group the cells. We further compare the clustering results with the four cell types.

Since the K-means algorithm assigns arbitrary labels to clusters, an optimal mapping between the predicted labels and the true labels is required. Let $L = \{1, 2, 3, 4\}$ be the set of labels assigned by K-means, and $T = \{\textit{GM12878}, \textit{Hela}, \textit{HAP1}, \textit{K562}\}$ be the set of true labels. Define $\mathcal{P} = \{\pi : L \to T \mid \pi \text{ is a one-to-one mapping}\}$ as the collection of all bijections from L to T.

For each sample i, the K-means algorithm produces a predicted label $\hat{y}_i \in L$. The mapping π then assigns a true label $\pi(\hat{y}_i) \in T$ to the prediction. Our goal is to find the mapping π^* that maximizes the total number of correctly assigned labels. In other words, if $y_i \in T$ denotes the true label for the i-th sample and $\mathbb{I}\{\cdot\}$ is the indicator function, then

$$\pi^* = \arg\max_{\pi\in\mathcal{P}} \sum_{i=1}^{N} \mathbb{I}\{\pi(\hat{y}_i) = y_i\}.$$

The clustering accuracy is then defined as the fraction of samples that are correctly classified by the optimal mapping:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\{\pi^*(\hat{y}_i) = y_i\}.$$

This approach effectively aligns the cluster assignments with the true cell types by exhaustively evaluating all possible label mappings, which is equivalent to solving a linear assignment problem on the contingency table of predicted versus true labels. The reported accuracy reflects the best possible alignment between the unsupervised cluster assignments and the known cell type labels.

4 Preliminary Results

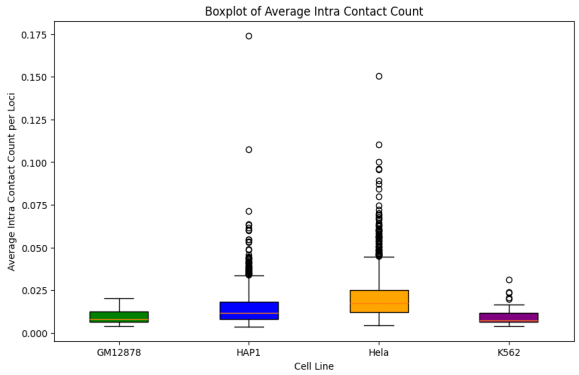
Given a specific cell C, we can find its average intrachromosomal counts per loci by dividing all intrachromosomal contacts by the sum of the total genomic loci of 23 chromosomes.

Figure 4a shows a boxplot of the average contact counts per loci for each cell type, while Figure 4b displays the corresponding density distributions. The density curve normalizes the distributions onto the same scale, facilitating direct comparison, and revealing differences in their shapes. The result from the boxplot closely aligns with the density distribution, with *Hela* cells having the higher mean of average contact per loci, then *HAP1*, *K562*, and *GM12878* having the lowest average contact counts per loci.

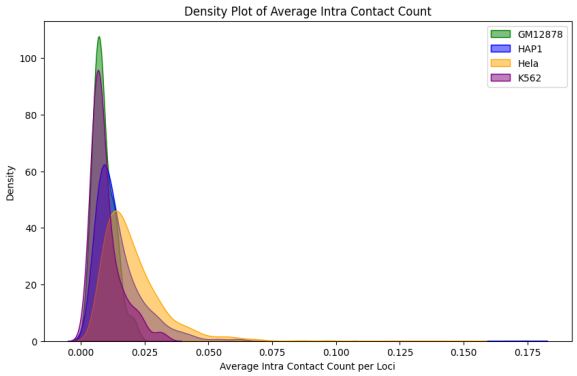
An unpaired t-test was performed on each pair of distinct cell types' average contact count distributions, with the null hypothesis that both distributions share the same mean. The comparison results are presented in Table 1. As shown, all pairs except *GM12878* vs. *K562* yield p-values significantly below 0.05, indicating strong evidence that their distributions differ in their means, which agrees with the visualizations in Figure 4a and Figure 4b.

Comparison	**t-statistic**	**p-value**	**Significant (p<0.05)**
GM12878 vs *HAP1*	-2.61193	0.00914707	True
GM12878 vs *Hela*	-4.44322	9.45423e-06	True
GM12878 vs *K562*	-0.533242	0.595555	False
HAP1 vs *Hela*	-10.8191	1.05962e-26	True
HAP1 vs *K562*	3.2454	0.00121326	True
Hela vs *K562*	5.88375	4.83609e-09	True

Table 1: T-test comparisons among cell lines.



(a) Boxplot of distributions of average contact counts per loci, where the vertical axis represents the average of contact counts per loci of individual cells.



(b) Density distribution of average contacts, where the horizontal axis represents the average of contact counts per loci of individual cells, and the vertical axis is the corresponding density.

Figure 4: Average counts per loci distribution of four cell types

On the other hand, for each cell type, we can find the average contact for each genomic loci, concatenate the results for the upper half of C and create a more detailed visualization on the cell level instead of the sample level. We are taking the upper triangular region for the cell since the matrix is symmetric. As shown in Figure 5, for each subgraph, the horizontal axis represents the concatenated location of genomic loci of the upper triangular region of the cell, where the vertical axis represents the average contact counts for the loci. The genomic region is marked for each chromosome. From the figure we can observe some interesting pattern, for example *K562* and *GM12878* has similar average genomic distribution, whereas *HAP1* and *Hela* are more similar.

For each cell type, the averaged contact matrix can also be plotted to guide our analysis. For instance, Chromosome 4 of average *Hela* is plotted in Figure 6a, Chromosome X of averaged *HAP1* is

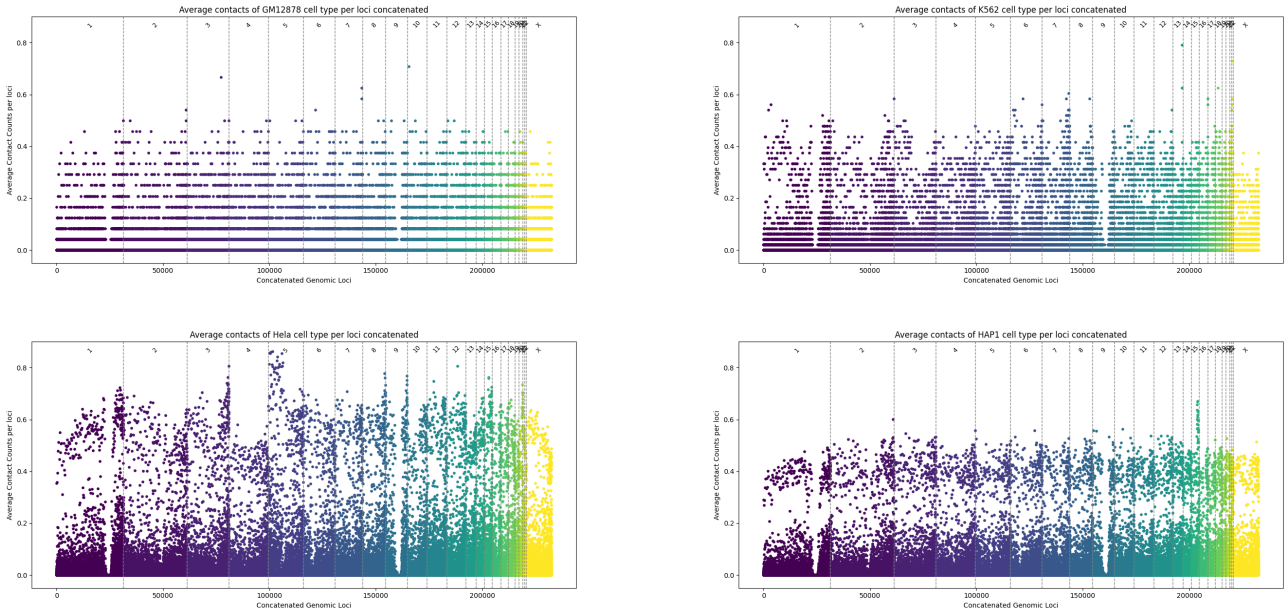
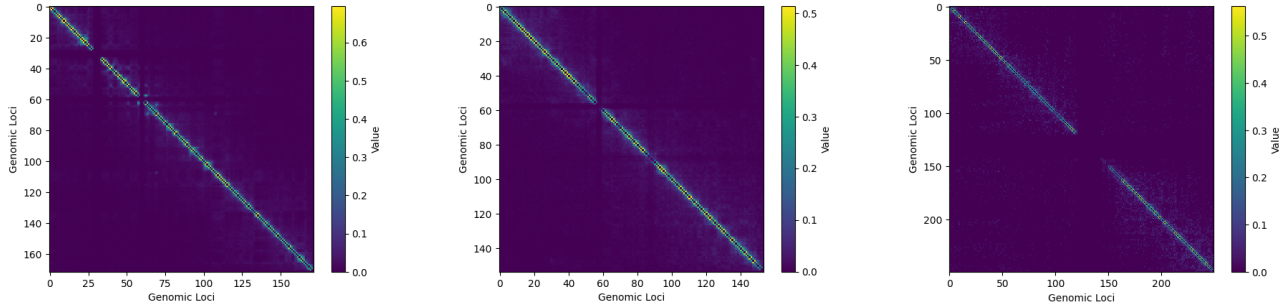


Figure 5: Concatenated Average Contacts per loci for four cell types.

plotted in Figure 6b, and Chromosome 1 of average *K562* is plotted in Figure 6c.



(a) Averaged C_4 of *Hela* (b) Averaged C_X of *HAP1* (c) Averaged C_1 of *K562*

Figure 6: Three averaged chromosomal matrices.

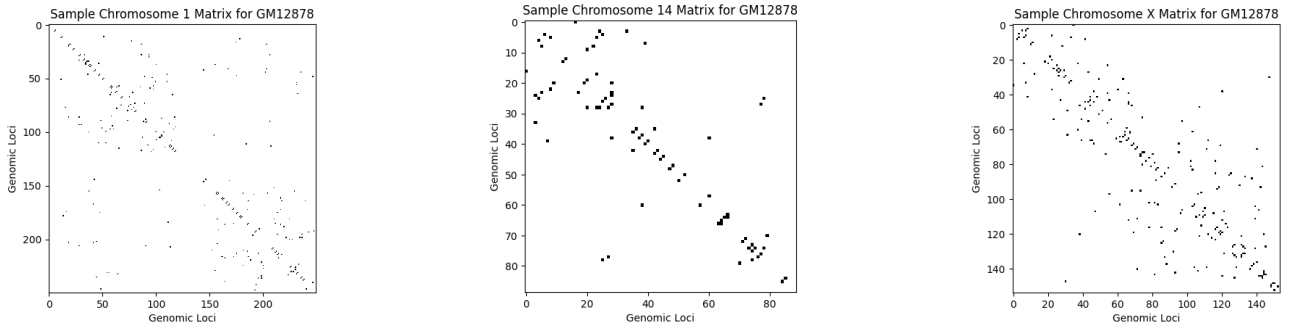
The following analysis was performed on a subsample of size $n = 172$, and was resampled 30 times to confirm the proposed method's robustness.

Following the methods described above, the preprocessed contact matrices were stored as $\{C^{(i)}\}$, where each matrix $C^{(i)}$ comprises chromosome-specific submatrices $\{C_j^{(i)}\}$ for $j \in \{1, 2, \ldots, 22, X\}$ and $i \in \{1, 2, \ldots, n\}$. Figure 7 shows three representative chromosomal submatrices from sample $C^{(14)}$. In these figures, black marks indicate the presence of contacts. Note that the contact density decreases with increasing distance from the diagonal.

After loading the chromosomal contact matrices for each cell, we computed a low-rank approximation for each chromosome using a rank $r = 50$. For each cell $C^{(i)}$, the approximation is denoted as

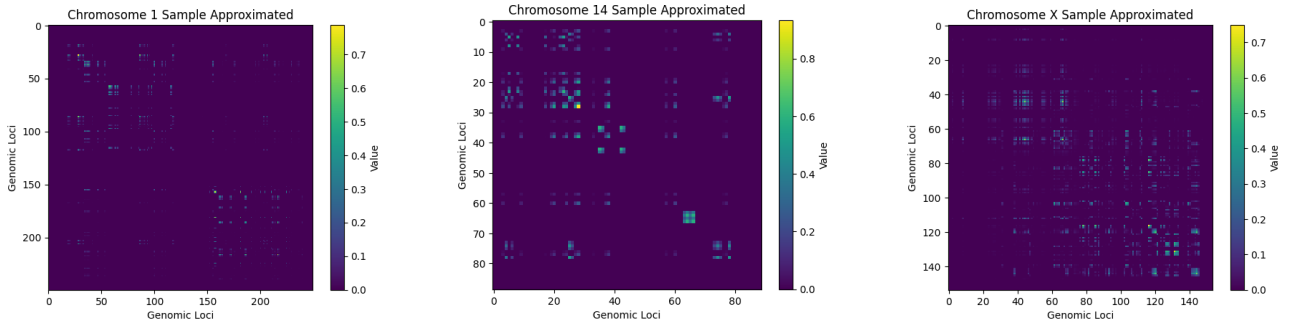
$$M^{(i)} = \text{diag}(M_1^{(i)}, M_2^{(i)}, \ldots, M_{22}^{(i)}, M_X^{(i)}),$$

where each $M_j^{(i)}$ is the block corresponding to the chromosome j from the optimal low-rank approximation of $C_j^{(i)}$. Figure 8 shows three approximated chromosomal matrices corresponding to those in Figure 7. The entries of these matrices, previously binary, are now continuous values reflecting contact intensity.



(a) $C_1^{(14)}$ (b) $C_{14}^{(14)}$ (c) $C_X^{(14)}$

Figure 7: Three samples of chromosomal matrices for the 14th sample cell of *GM12878* Cell Type



(a) $M_1^{(14)}$ (b) $M_{14}^{(14)}$ (c) $M_X^{(14)}$

Figure 8: The low rank approximated matrices corresponding to the three previous examples

The best rank is determined by finding the elbow point of the total approximation error

$$\frac{1}{n}\sum_{i=1}^{n}\|C^{(i)} - M^{(i)}\|_F^2$$

, identifying that using 72 rank approximation should help us find a balance between explainability in variance and data complexity. However, we used rank $r = 50$ as the optimal rank to construct the further analysis based on the empirical fact that $r = 50$ outperforms $r = 72$ in all cases in terms of clustering accuracy.

Using the set of low-rank approximated matrices $\{M^{(i)}\}$ as representations of the sample cells, we performed PCA to extract the dominant variance components. Each matrix $M^{(i)}$ was vectorized, and the resulting vectors were stacked to form a data matrix $P \in \mathbb{R}^{n\times 2965^2}$, where each element corresponds to a feature and $n$represents the size of subsample taken. The data were centred by subtracting the mean of each column, and PCA was then applied to determine the directions that preserve maximum variance. As shown in Figure 10, the first two principal components provide a clear two-dimensional separation among the four cell types, with *Hela* cells (in orange) distinctly isolated. In Figure 11, the histogram shows the cumulative variance explained by the top 20 principal components, and the red line overlay shows the variance explained by each principal component. The total variance explained by the top 20 is approximately 40%.

Figure 12 displays the detailed PCA loadings, where brighter regions indicate a stronger contribution to the observed variance. Across all chromosomes, we can observe a diagonal-block pattern with higher PCA loadings, having higher importance in helping us differentiate four cell types by carrying more information. Also, we can see that there appears to be a cross-shaped pattern colored in blue in most chromosomes, where lower PCA loadings are associated with. Ruling out the areas with lower PCA loadings and focusing on regions with significant PCA loadings might help us further extrapolate the characteristics of cell types.

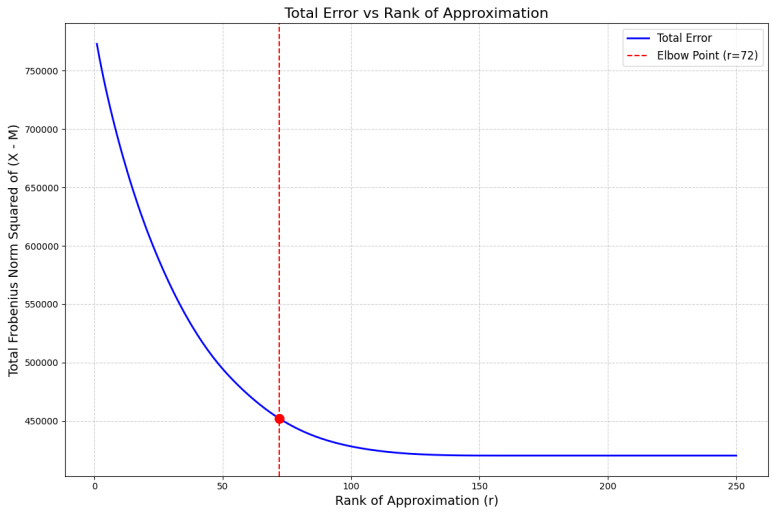


Figure 9: Total Error with respect to rank used for approximation

Subsequently, we applied Uniform Manifold Approximation and Projection (UMAP) to the PCA loadings to further enhance the visualization in a three-dimensional space. As illustrated in Figure 13b, both *Hela* and *K562* cells form distinct clusters, effectively differentiating them from the other groups. More details of the effect of number of principal components taken on the embedding results can be fount in Figure 14.

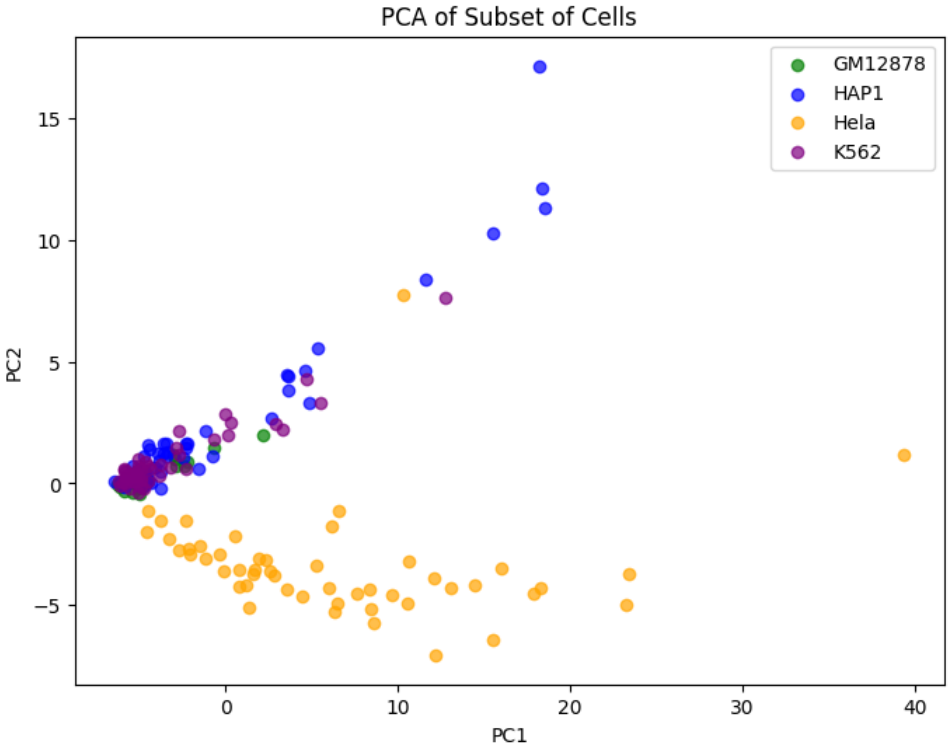


Figure 10: Representing top 2 principal components of the approximated matrices

The number of principal components used for dimensionality reduction on the approximated matrices was chosen based on the performance of subsequent embedding and clustering. After manually testing various options, it was found that using 20 principal components provided the best overall results. Detailed comparisons of the clustering accuracy using 10, 20, and 50 principal components for both the original and approximated matrices are presented in Table 2, where each round of the experiment is performed on 30 subsamples of size $n = 175$. The Table 2 shows the mean clustering

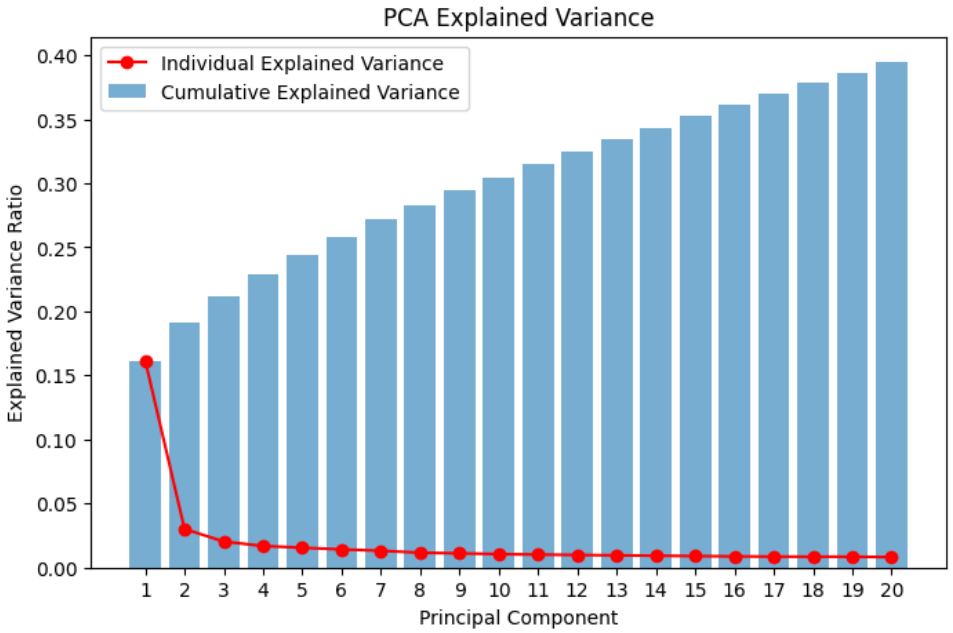


Figure 11: Total Variance explained by PCs and the variance explained by each PC

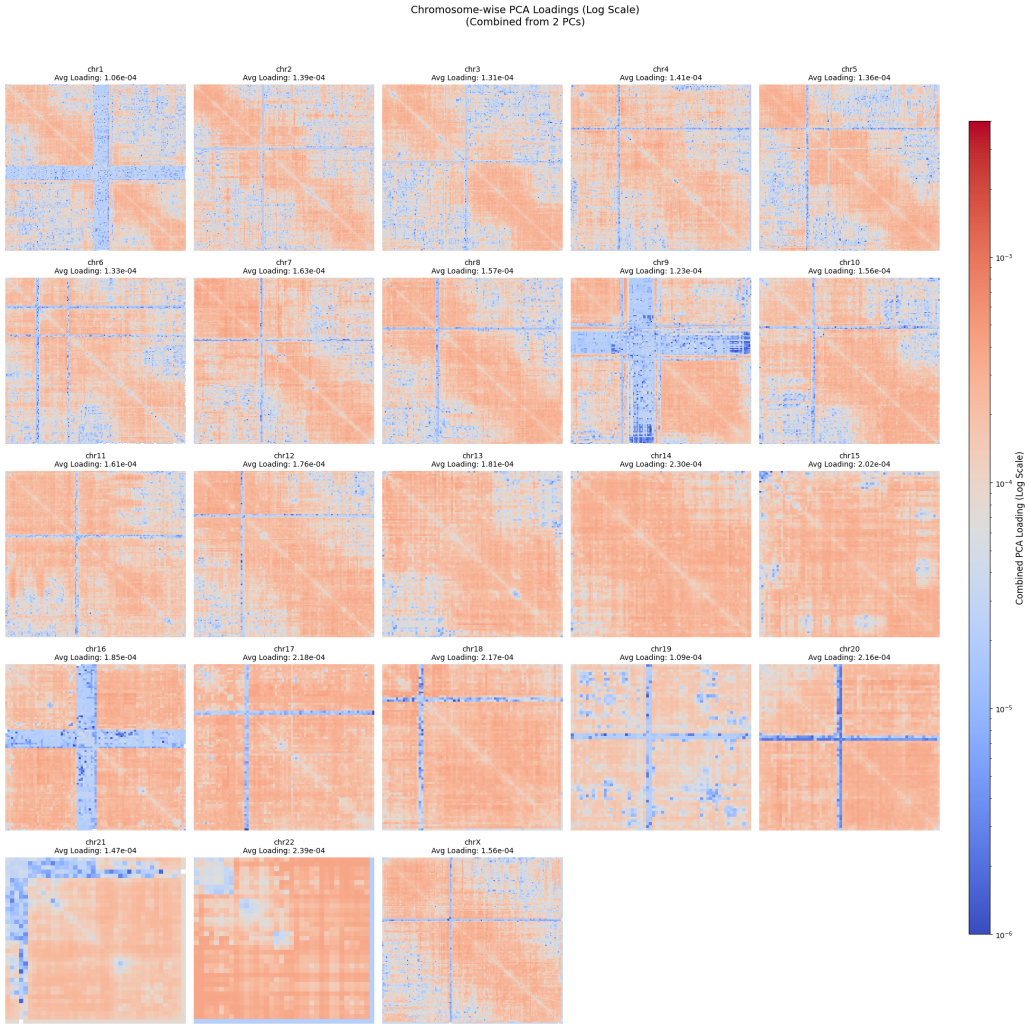
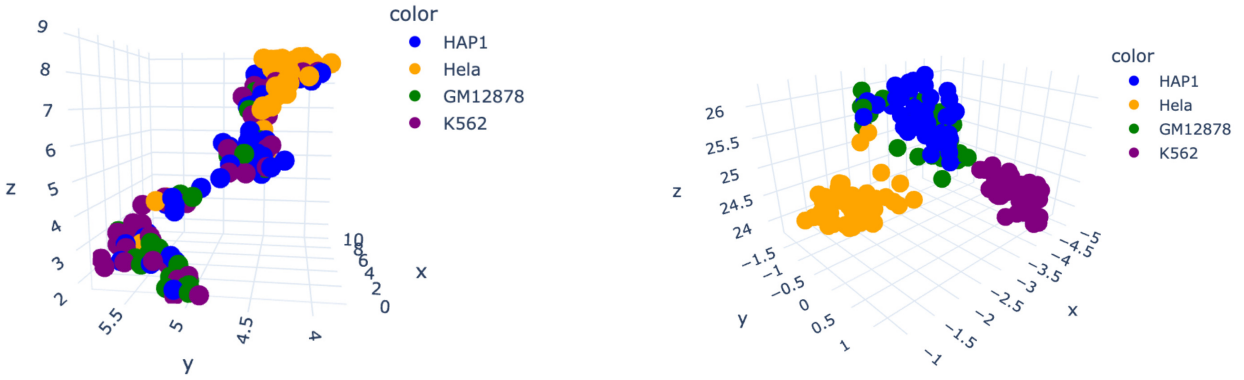


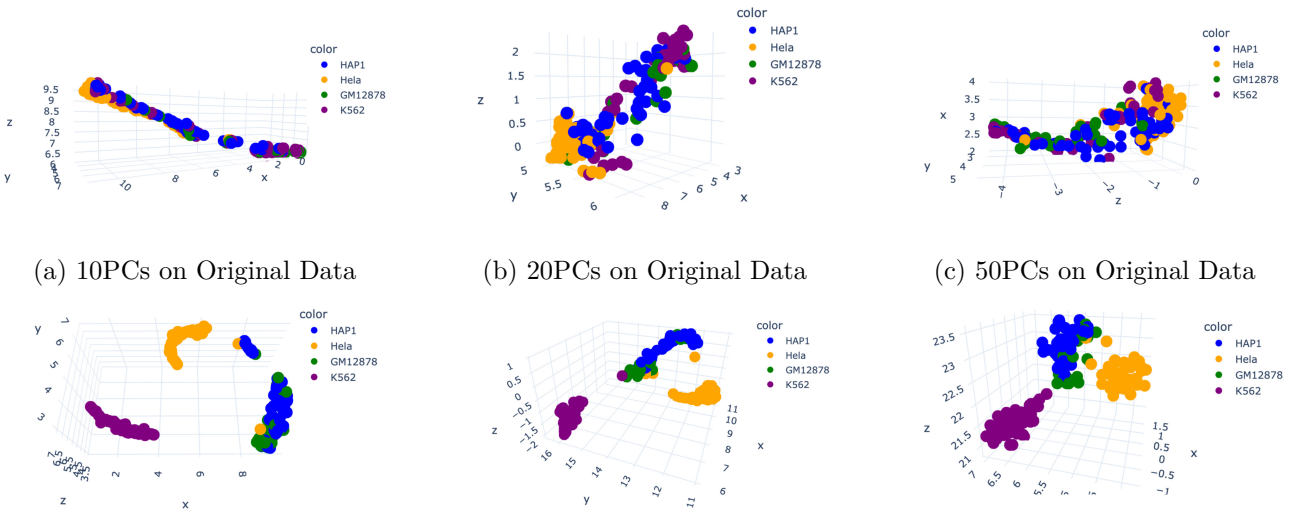
Figure 12: Loadings of PCA on matrix



(a) UMAP of original samples

(b) UMAP of processed samples

Figure 13: UMAP embeddings comparisons



(a) 10PCs on Original Data

(b) 20PCs on Original Data

(c) 50PCs on Original Data

(d) 10PCs after Approximation

(e) 20PCs after Approximation

(f) 50PCs after Approximation

Figure 14: UMAP embeddings for original and approximated data. The top row displays embeddings generated from the original data, while the bottom row shows those based on low-rank approximated data. The first, second, and third columns correspond to analyses performed using 10, 20, and 50 principal components, respectively.

accuracy and corresponding 95% confidence interval.

	10PCs	20PCs	50PCs
Original	45.48% (±1.06%)	47.38% (±1.35%)	50.48% (±1.46%)
Approximated	79.96% (±1.88%)	80.14% (±1.76%)	76.69% (±2.19%)

Table 2: Clustering accuracy comparison using different numbers of principal components

To quantify visual separability of the cell types in 3D embeddings obtained by UMAP we apply K-Means clustering and assess the clustering performance as discussed in Section 3.4. We further compare the performance with applying K-Means clustering to the principal components representations of cells on the raw contact data, which results in an accuracy of only about 51%. However, after processing the cells using low-rank approximation, PCA reduction, and UMAP embeddings, the clustering accuracy increases dramatically to 82%. This substantial improvement demonstrates the effectiveness of the proposed methods in enhancing clustering performance.

5 Discussion

In our study, we observed that the selection of rank in the low-rank approximation stage has a significant impact on clustering performance. While the traditional elbow method in the loss curves provides a convenient heuristic for choosing the rank, our results indicate that alternative rank selections can sometimes yield superior performance. This suggests that a fixed rank might not be optimal for all chromosomes, as each contributes differently to the overall heterogeneity among cells. Future work should focus on developing dynamic, chromosome-specific strategies for rank selection to further enhance clustering accuracy.

Moreover, our proposed methods, which combine low-rank approximation, PCA reduction, and UMAP embeddings, led to a remarkable improvement in clustering accuracy—from 51% with the raw matrix representations to 82% after processing. This dramatic increase underscores the potential of our approach to effectively capture and leverage the underlying structure in the data for better cell clustering.

It is important to note, however, that the significance of our experimental results may be somewhat limited by dataset imbalance. Some cell types were underrepresented, which likely affected the overall performance of the clustering algorithms. In fact, when focusing solely on the binary classification and clustering of the two cell types with the largest sample sizes, we observed even higher accuracy. Therefore, employing a more balanced dataset in future experiments could provide even more compelling evidence of the method's effectiveness and generalizability.

References

[1] HJ Kim, GG Yardımcı, G Bonora, V Ramani, J Liu, et al. Capturing cell type-specific chromatin compartment patterns by applying topic modeling to single-cell hi-c data. *PLOS Computational Biology*, 16(9):e1008173, 2020.

[2] Jie Liu, Dejun Lin, Galip Gürkan Yardımcı, and William Stafford Noble. Unsupervised embedding of single-cell hi-c data. *Bioinformatics*, 34(13):i96–i104, Jul 2018.

[3] Qiao Liu, Wanwen Zeng, Wei Zhang, Sicheng Wang, Hongyang Chen, Rui Jiang, Mu Zhou, and Shaoting Zhang. Deep generative modeling and clustering of single cell Hi-C data. *Briefings in Bioinformatics*, 24(1):bbac494, 2023.

[4] Vijay Ramani, Xinxian Deng, Ruolan Qiu, Kevin L Gunderson, Frank J Steemers, Christine M Disteche, William S Noble, Zhijun Duan, and Jay Shendure. High-resolution hi-c data (ramani et al. 2017). `https://pages.stat.wisc.edu/~sshen82/bandnorm/Ramani2017/`, 2017. Accessed: 17-Feb-2025.

[5] Vijay Ramani, Xinxian Deng, Ruolan Qiu, Kevin L Gunderson, Frank J Steemers, Christine M Disteche, William S Noble, Zhijun Duan, and Jay Shendure. Massively multiplex single-cell Hi-C. *Nature Methods*, 14(3):263–266, 2017.

[6] Rui Zhang, Tianming Zhou, and Jian Ma. Multiscale and integrative single-cell Hi-C analysis with Higashi. *Nature Biotechnology*, 40:254–261, 2022.

[7] Jingtian Zhou, Jianzhu Ma, Yusi Chen, Chuankai Cheng, Bokan Bao, Jian Peng, Terrence J. Sejnowski, Jesse R. Dixon, and Joseph R. Ecker. Robust single-cell Hi-C clustering by convolution- and random-walk–based imputation. *Proceedings of the National Academy of Sciences*, 116(28):14011–14018, 2019.

Gram-Schmidt & QR Decomposition

In $\mathbb{R}^2$, two arbitrary vectors x_1, x_2. Define one orthonormal vector $v_1 = x_1$, find projection of x_2 onto v_1 as $Proj_{v_1}x_2$. Note that $v_2 = x_2 - Proj_{v_1}x_2$ and $Proj_{v_1}x_2 = \beta_{21}v_1$. Since $v_2 \cdot v_1 = 0$ and $(x_2 - \beta_{21}v_1) \cdot v_1 = 0$, resulting in $\beta_{21} = \frac{x_2 \cdot v_1}{v_1 \cdot v_1}$. Therefore we have $v_2 = x_2 - \frac{x_2 \cdot v_1}{v_1 \cdot v_1}v_1$ as the second orthonormal vector.

In $\mathbb{R}^3$, three arbitrary vectors x_1, x_2, x_3. By previous step we can find $v_1 = x_1$ and $v_2 = x_2 - \frac{x_2 \cdot v_1}{v_1 \cdot v_1}v_1$ which span the space of $span(x_1, x_2)$. Projecting x_3 onto $span(v_1, v_2)$, we have $v_3 = x_3 - Proj_{x1,x2}x_3 = x_3 - (\beta_{31}v_1 + \beta_3 2v_2)$. By the fact $v_3 \perp v_1, v_3 \perp v_2$ we have $v_3 \cdot v_1 = v_3 \cdot v_2 = 0$, resulting in $v_3 = x_3 - \frac{x_3 \cdot v_1}{v_1 \cdot v_1}v_1 - \frac{x_3 \cdot v_2}{v_2 \cdot v_2}v_2$.

In $\mathbb{R}^n$, n arbitrary vectors $x_1, x_2, \cdots, x_n$. Following the previous $n-1$ steps we have the $v_1, \cdots v_{n-1}$ orthogonal vectors, which spans $span(x_1, x_2, \cdots, x_{n-1})$. Projecting x_n onto $span(v_1, \cdots, v_{n-1})$ we have $v_n = x_n - \frac{x_n \cdot v_1}{v_1 \cdot v_1}v_1 - \frac{x_n \cdot v_2}{v_2 \cdot v_2}v_2 - \cdots - \frac{x_n \cdot v_{n-1}}{v_{n-1} \cdot v_{n-1}}v_{n-1}$.

Above process inductively holds but is not tidy enough in terms of normalizing the vectors. The following process links Gram-Schmidt with the idea of regression, which provides a more intuitive interpretation of Gram-Schmidt.

Note that during each step we are finding the projection of a new original vector on to the existing span of previous normal vectors. In $\mathbb{R}^2$, assume that we already have $X_1 = V_1$, and

$$\begin{bmatrix} | \\ X_2 \\ | \end{bmatrix}$$

represents the second original vector. Now we assume that we are finding the projection of this X_2 onto V_1, which is equivalent to minimizing the Frobenius error between the projected vector and the

original vector. Now this problem can be transformed to an minimization task:

$$\min_{\beta_1} \|X_2 - V_1\beta_1\|_F^2$$

, which is directly related with the regression concept and its closed form solution can be found directly by $\beta_1 = (V_1^T V_1)^{-1} V_1^T X_2$. Taking this idea to the $\mathbb{R}^n$ case, where X_n represents the $n-th$ original vector, and matrix V has form

$$\begin{bmatrix} | & | & & | \\ V_1 & V_2 & \cdots & V_{n-1} \\ | & | & & | \end{bmatrix}$$

has previous found $n-1$ orthogonal vectors as columns. Therefore the tasks is to find the solution β_{n-1} for

$$\min_{\beta} \|X_n - V\beta_{n-1}\|_F^2$$

, and similarly the close form solution is $\beta_{n-1} = (V^T V)^{-1} V^T X_n$. Here the $(V^T V)^{-1}$ is a $n-1$ by $n-1$ diagonal matrix with each V_i's norm squared $\|V_i\|^2$ on the diagonal entries. Generally speaking, the resulting V matrix can be found using $X\beta^{-1}$, where β is the matrix with coefficients β_i as column. For instance the first column should be with 1 on first row with trailing 0s, the second column should be with $-\beta_1$ on first row, 1 on second, and trailing 0s. This forms the upper triangular matrix β, serves as the transformation from original matrix X to the Gram-Schmidt result V. Here, if we reorder the equation, we have $X = V\beta$, which is exactly the QR decomposition of matrix X, with $V = Q$ as the basis matrix and $\beta = R$ as the upper triangular matrix with coefficients recorded.

Positive Semi Definite Matrix

A matrix A is positive semi-definite means that the $x^T Ax \geq 0$ for any column vector x, and A has to be symmetric. The property guarantees that the matrix A has non-negative eigenvalues, which can be useful when determining the best number of principal components that can represent a contact matrix.

Low Rank Approximation of Matrix

Given an original matrix $X \in \mathbb{R}^{m\times n}$, we can find low rank approximated matrix M', where $rk(M') \leq r$, such that $M' = \underset{rk(M)\leq r}{\operatorname{argmin}} \|X - M\|_F^2$, where M' is the truncated SVD given rank r.

Step one is to transform into SVD. Assume M where $rk(M) \leq r$, let $M = UAV^T$, here $A \in \mathbb{R}^{m\times n}$ with $rk(M) \leq r$. So we have

$$\|X - M\|_F = \left\|U\Sigma V^T - UAV^T\right\|_F = \left\|U(\Sigma - A)V^T\right\|_F$$

, since U and V^T are orthogonal matrices, we have $\|X - M\|_F = \|\Sigma - A\|_F$.

Step two is to minimize, where we want to find the best A with rank less than r that minimizes $\|\Sigma - A\|_F$. Σ is diagonal, A can only be diagonal to minimize the distance to Σ, otherwise off-diagonal non-zero entries will only enlarge the difference. Thus A is diagonal, with $a_{ii} = \sigma_{ii}$, if $i \leq r$ and 0 otherwise. Simply saying A is the matrix resulting from setting $\sigma_{ii} = 0$ when $i > r$ in Σ.

Step three is to find the Frobenius norm of distance. $A = \Sigma_r = diag(\sigma_1, \sigma_2, \cdots \sigma_r, 0\cdots)$, so we have

$$\|\Sigma - A\|_F = \|\Sigma - \Sigma_r\|_F = \sum_{i=1}^{p} (\sigma_{ii} - a_{ii})^2 = \sum_{i=r+1}^{p} (\sigma_{ii}^2)$$

.

Step four is to prove the distance found is the minimum. Let A' be the matrix with rank $\leq r$, so

$$\|\Sigma - A'\|_F^2 = \sum_{i=1}^{p} (\sigma_{ii} - a'_{ii})^2 + \sum_{i\neq j} \left|a'^2_{ij}\right|$$

. For off diagonal entries $a'_{ij} = 0$ when $i \neq j$. Only when having $a'_{ii} = \sigma_i$ if $i \leq r$ and choosing $a'_{ii} = 0$ minimizes the above summation term. So Σ_r is the only matrix that minimizes $\|\Sigma - A\|_F$. We have the low rank approximation $M' = \underset{M}{\operatorname{argmin}} \|X - M\|_F^2 = U\Sigma_r V^T$, and the minimized $\|X - M'\|_F^2 = \sum_{i=r+1}^{p} \sigma_i^2$.

Singular Value Decomposition

Given a matrix m by n X, we can perform SVD to have the decomposition $X = U\Sigma V^T$, where U has size m by m, Σ in m by n, and V^T in n by n. The columns of V form an orthogonal basis for $\mathbb{R}^n$, the row space of X. The columns of U, form the orthogonal basis for $\mathbb{R}^m$, the column space of X. Σ is the diagonal matrix which contains the singular values on the diagonal entries ordered from largest to smallest in value.

Steps to perform SVD: Step 1, $X^T X = V\Sigma^2 V^T$. Step 2, Solve for $det(X^T X - \lambda I) = 0$, finding $\lambda_1, \cdots$. Step 3, for each λ_i, substitute it into $X^T X - \lambda_i I$, to find a vector v_i that is orthogonal to the row space, i.e. the null space of row vectors, then normalize it to be the i-th column of V. For all the λ_i we found we take $\sigma_i = \sqrt{\lambda_i}$ on the diagonal of Σ. After V and Σ are found we can also find U.

U represents the left singular vectors, serving as a column orthonormal matrix that captures the principal directions in the original data space. These eigenvectors form a transformed coordinate system. Each column of U acts as a basis vector, and are arranged in order of decreasing importance, corresponding to the magnitude of singular values.

Σ is a diagonal matrix composed of non-negative singular values. These values are sorted in descending order, serving as a measure of each principal direction's significance. The diagonal entries essentially scale the singular vectors, providing a precise representation of how much information or variance each direction contributes to the overall data. Larger singular values indicate more dominant patterns, while smaller values suggest less critical variations.

V, consisting of right singular vectors, complements the U matrix by representing the principal directions in the feature space. Orthonormal in nature, the V matrix's rows provide a comprehensive mapping of how the original data can be projected and transformed across different feature dimensions. Each row captures a specific directional characteristic of the data, offering insights into the underlying geometric and statistical properties of the original matrix.

Eigen Decomposition

Given a square $n \times n$ matrix A, we can perform an eigen decomposition when A is diagonalizable). This decomposition can be written as

$$A = V\Lambda V^{-1},$$

where V is an $n \times n$ matrix whose columns are the eigenvectors of A, and Λ is an $n \times n$ diagonal matrix containing the corresponding eigenvalues along its diagonal. Λ is diagonal matrix with entries $\lambda_1, \lambda_2, \ldots, \lambda_n$ (the eigenvalues of A). These are usually arranged in descending or ascending order, depending on the application. V is the matrix of eigenvectors. Each column $\mathbf{v}_i$ is an eigenvector associated with the eigenvalue λ_i. When A is diagonalizable, these eigenvectors form a basis for $\mathbb{R}^n$

Steps to perform Eigen Decomposition:

1. Form the characteristic polynomial: Solve

$$\det(A - \lambda I) = 0$$

   to find all eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$.

2. Find each eigenvector: For each eigenvalue λ_i, solve

$$(A - \lambda_i I)v_i = 0$$

to find its corresponding eigenvector(s) v_i. Normalize or scale the vectors appropriately.

3. Construct V and Λ:

$$V = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}.$$

4. Decompose:

$$A = V\Lambda V^{-1}.$$

The diagonal entries of Λ, λ_i, indicate how A *scales* the corresponding eigenvector v_i. A larger eigenvalue implies a stronger stretching in the direction of its eigenvector, while an eigenvalue of smaller magnitude implies less influence in that direction. The columns of V form a set of basis vectors that diagonalize A. Each column v_i is the eigenvector associated with λ_i. In geometric terms, $Av_i = \lambda_i v_i$, meaning v_i is a direction in which A acts as a simple scaling by λ_i.
Special Case: Symmetric Matrices

- If A is real and symmetric, it can be orthogonally diagonalized:

$$A = Q\Lambda Q^T,$$

  where Q is an orthonormal matrix ($Q^TQ = I$). This simplifies the inverse to $Q^{-1} = Q^T$.

- The eigenvectors in this case are orthonormal and can be chosen to be perpendicular to each other, greatly simplifying many computations in numerical linear algebra.

PCA Steps

Given the original data X_0 in $p \times n$ with p features and n observations, we can first center the data by $X_{ik} = X_{0ik} - \bar{X}_k$, where column mean is 0. The centered data X has $\sum_{i=1}^{n} X_{ik} = 0, \forall k$. Finding the top m principal components is the process of finding the m directions $\alpha_1, \alpha_2, \cdots, \alpha_m$ such that the projected data onto these directions maintain variance the most.

For a direction e^T through the mean centered point $\bar{X}$, the projection of a data point X_i is $e^T(X_i - \bar{X})$. The projected variance is

$$\frac{1}{n}\sum_{i=1}^{n}[e^T(X_i - \bar{X})((X_i - \bar{X})e)] = e^T\Sigma e$$

. The first principal component e_1 can be found by $e_1 = \underset{e}{\text{argmax}}\ e^T\Sigma e,\ s.t.\ e^Te = 1$. Using Lagrange Multiplier we can find $\lambda = e^T\Sigma e$, where λ_1 is the max projection variance, and e_1 be the first principal.

Based on the first principal component we can find the second pc as well. Given the first pc e_1, e_2 can be formulated as $e_2 = \underset{e_2}{\text{argmax}}\ e^T\Sigma e,\ s.t.\ e^Te = 1, e_1^Te = 0$. Also by Langrange Multiplier we can find $\Sigma e_2 = \lambda_2 e_2$.

K-Means

In the $p-th$ dimension, the K-Means optimization task is $\underset{c_1,c_2,\cdots,c_n}{\text{minimize}} f(c_1, c_2, \cdots, c_n)$, where $f(c_1, c_2, \cdots, c_n)$ is defined as

$$\sum_{i=1}^{n} \|x_i - c\|_2^2 = \sum_{i=1}^{n}\sum_{j=1}^{p}(x_{ij} - c_j)^2$$

. Take $\frac{\partial f}{\partial c_j} = -2\sum_{i=1}^{n}(x_{ij} - c_j) = 0$, obviously the solution to this is $c_j = \bar{x_j}$. Therefore similarly we can have the result $c = (\bar{x_1}, \bar{x_2}, \cdots, \bar{x_n})$. Since $\frac{\partial^2 f}{\partial c_j^2} = 2n > 0$, so we have the c as global minimum.

The steps for the K-Means algorithm can be generalized as follows:

1. For a fixed set of $\{C_k\}$, find $C_k = \frac{1}{|C_k|}\sum_{x_i \in C_k} x_i$.

2. For given centroids $\{c_k\}$, assign each x_i to each cluster and find the one that minimizes $\|x_i - c_k\|_2^2$, where each x_i is assigned properly and $\sum_{x_i \in C_k} \|x_i - c_k\|_2^2$ is minimized.

3. The term $\sum_{k=1}^{K}\sum_{x_i \in C_k} \|x_i - c_k\|_2^2$ decreases or stays the same after a given iteration. Minimizing pairwise distance is equivalent to minimizing the distance to the centroid, which proof can be performed through setting:

$$S_k = \sum_{i,i' \in C_k} \|x_i - x_{i'}\|_2^2 = \sum_{i,i' \in C_k} [\|x_i - c_k\|_2^2 + \|x_{i'} - c_k\|_2^2 - 2(x_i - c_k)^T(x_{i'} - c_k)].$$