國 立 政 治 大 學 資 訊 科 學 系

碩 士 學 位 論 文

(Proposal)

# **染色質區室分析之探索與優化**

**The exploration and optimization of the chromatin compartment analysis**

指導教授：張 家 銘 博士

研究生：程 至 榮

中 華 民 國 113 年 1 月 11 日

## **Abstract**

Since the first Hi-C contact matrix was published, countless researchers have utilized this method to detect the 3D structure of the whole genomes, especially for identifying the transcriptionally active and inactive loci among the chromosomes (Roughly defined as the A and B compartments). Although there are several options for identifying these compartments, the most acceptable way in practice is still the well-known PCA-based identification due to its simplicity and robustness. However, with the higher resolution provided by these contact matrices, thanks to the improvement of DNA sequencing coverage, the computational effort and memory requirement for the traditional PCA becomes intractable nowadays. Here, we proposed a heuristic approach for approximating the pattern of the first principal component (PC1) without doing PCA, which is crucial in labeling A/B compartments. Besides, we discuss the intuition behind the relationship between PC1 and A/B compartments and how the explained variance of the PC1 might impact the result of compartment identification to provide a caveat for future study.

**Keywords**: Hi-C, Chromatin compartments analysis, Principal Component Analysis (PCA)

## **Content**

1. Introduction
   1. Hi-C
   2. Chromatin Compartment Analysis
2. Materials and Methods
   1. Datasets
   2. PC1-pattern Approximation Algorithm
   3. Explained Variance Comparison
3. Results
   1. PC1-pattern Approximation Results
   2. Explained Variance Comparison Results
4. Future Works
5. References

## **Introduction**

### **Hi-C**

The Hi-C technique was first proposed in *Science* (2009) by Erez Lieberman-Aiden et al. [1], which is adapted from the chromosome conformation capture (3C) method [2] for analyzing the folding principle and spatial organization of the genome. The 3C-like methodologies study the long-range interactions between genomics regions (i.e., “loci”, sometimes we call it “genomics bins”) via cross-linking the interacting sites with formaldehyde, digesting the chromatin with a restriction enzyme to create the ligation-products, and quantifying these ligation-products through Polymerase Chain Reaction (PCR) [11]. Compared to 3C and the other adaptations such as 4C [3] and 5C [4], Hi-C could probe the genomics proximity ligation for the whole genomes without using any locus-specific PCR, and it allowed massively pair-end sequencing with the purification of invalid interaction pairs (“chimeric reads”) [1][7].

Hi-C provides a so-called “contact matrix” representing the interaction frequency between each pair of separated genomics bins. This matrix presents lots of information and demonstrates a variety of studies about the chromosome territories [1], chromatin compartments analysis [1], chromatin loops [10], and Topologically Associated Domains (TADs) [12][13], which reflects the importance of the Hi-C technique. Based on a definition stated before, the “resolution” of a Hi-C matrix means “The smallest locus size such that 80% of loci have at least 1,000 contacts” (Rao et al., 2014) [10], which represents the finest scale a matrix could be inspected. The first contact matrix constructed by Erez Lieberman-Aiden et al. in 2009 merely divides the chromatin into one megabase per genomics bin (1Mb) [1]; thanks to the advancement of DNA sequencing the resolution of matrix in 2014 had already improved to 1 Kb resolution with 4.9 billion contacts [10], and a recent publication in 2023 even resolved the resolution to 500 base-pair (500 bp), produced a super large matrix with about 33 billion contacts [14]. Since the matrix size is proportional to the number of locus bins in the chromosome [5], analyzing a contact matrix with a higher resolution will dramatically increase the computational expense, such as the requirement of RAM storage and computing time. A high expense makes analyzing a fine-scale contact matrix difficult, especially when it requires a complicated process. The chromatin compartments analysis on the Hi-C matrix is a great example reflecting this problem since the commonly used method for the compartments analysis is the PCA-based identification with a complex eigen-decomposition.

### **Chromatin compartments analysis**

In 2009, Erez Lieberman-Aiden et al. confirmed that the genome can be partitioned into two sets of chromatin domains (open and closed) [1]. The open-domain is more accessible and contains the transcriptionally-active chromatin; on the other hand, the closed-domain contains the transcriptionally-repressed chromatin [15]. Furthermore, by exploring the Hi-C matrix, they surprisingly found out the enrichment of interaction signal will cluster into “plaids”, indicating that the loci could be separated into at least two sets (Labeled as A and B), in which the contacts between AA or BB sets will be enriched, and the contacts between AB will be depleted. Erez Lieberman-Aiden et al. defined these two sets of loci as the “compartment”, and they concluded that A compartment is closely related to the open-domain and B compartment is closely related to the closed-domain.

For the identification of A and B compartments, a series of data preparations are needed for the downstream analysis, which we term the “chromatin compartments analysis”. According to the previous studies [1][5], we explain the classical PCA-based chromatin compartments analysis step by step:

1. The raw contact matrix should be normalized due to the reason that the “visibility” of the contact matrix might be impacted by systematic biases such as GC content and the locus mappability [6][7]. There are several studies about matrix normalization; for example, the Iterative Correction and Eigenvector decomposition (ICE) [8] and the Knight-Ruiz (KR) matrix balancing [9] are widely applied to correct these biases.
2. Since Erez Lieberman-Aiden et al. found that the probability of contact will decrease monotonically as the genomic distance between loci increases, they proposed the sequence proximity should be accounted for during the analysis [1]. The distance effect will be considered by dividing the observed contacts with the genome-wide average contact at that distance (i.e., expected contacts) between each loci pairs (bin-pairs) [5], which transforms the raw contact matrix into the Observed/Expected matrix (O/E matrix).
3. As we can notice that the checkerboard-like O/E matrix exists multiple plaids, especially densely gathered along the diagonal of the matrix, Erez Lieberman-Aiden et al. [1] assumed the short-distanced loci pairs would have a similar interaction profile. Hence, they will be highly correlated in terms of contact frequency. In this step, the locus-to-locus correlation between each pair of loci should be calculated, and a Pearson correlation matrix from the previously constructed O/E matrix will be derived. The plaid-pattern is sharpened in the Pearson correlation matrix, and the actual meaning represented by the entities in the matrix is also converted from the contact interaction strength to the locus correlations [5].
4. In practice, the Principal Component Analysis (PCA) will be implemented on the Pearson correlation matrix, and the A and B compartments along the chromosome are labeled by the corresponding sign of the PC1 vector (positive or negative). Generally, the positive regions of PC1 are used to identify the A compartment (GC-content rich); however, because the signs of the eigenvector are mathematically arbitrary [25], sometimes a “flipped” of PC1 is needed, according to the GC-content distribution of the chromosome (i.e. inspect the “GC” distribution in the fasta file, measure the percentage of guanine or cytosine in DNA to locate the GC-content rich compartment; invert the sign of all the entries in PC1 to make the GC-content rich compartments labeled by the positive regions of PC1) [16].

Following the above instructions, we can easily identify the A and B compartments. Owing to its simplicity and straightforward process, there are multiple studies and software tools implemented the PCA-based identification for its compartments analysis, such as POSSUMM [14], FAN-C [16], HOMER [17], Cooler [18], Juicer [19] and many more. However, as mentioned in the previous section, the computing expense will increase extensively as the resolution gets finer. It is intractable for some laboratories to explore the fine-scale contact matrix with the chromatin compartment analysis. For example, we tested the execution time of calculating PC1 for the human chromosome 1 at 25 Kb on the GM12878 contact matrix [20][21]. The tool we used was the classical Hi-C analysis tool Juicer [19], with normal computing resources (Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz, 16GiB RAM). It already took about 70 minutes to finish the PC1 calculation at this coarse-scale resolution. Besides, according to the previous study [5], a dense matrix of human chromosome 1 at 1 Kb takes about 500 GB of memory, which is way more than the requirement of general conditions and prohibits the application for most labs.

Although a study in 2023 proposed an idea to fasten the PC1 calculation [14], some tricky problems still exist for PCA-based compartment analysis nowadays. Here we list some of these issues that we’re interested in below:

1. The Principal Component Analysis (PCA) is a statistical method originally designed for dimensional-reduction, and the reason why we can use the sign positive or negative of the PC1 vector to identify the A/B compartment is still unexplained. As a result, we cannot directly compare the different Hi-C datasets through the score of the PC1 vector since the biological meaning of this score is still mysterious [22].
2. The PC1 may not always reflect the compartment segregation but reflects the separation of chromosome arms. In some cases, the PC2 should be used to match the compartment interval correctly [1][5][23]. For example, Erez Lieberman-Aiden et al. in 2009 described that for GM12878 cells at 1 Mb resolution, PC2 should correspond to the compartmental structure for chromosomes 4 and 5 [1].
3. Although PC1 represents the principal component with the largest variance, the actual explained variance of PC1 has never been discussed, which might have an unignorable influence on compartmental identification. In our experiments, we found out that for GM12878 cells at 1 Mb resolution, the explained variance of PC1 for chromosomes 4 and 5 are only about 0.62 and 0.67, respectively, which might explain the reason why we cannot directly use the PC1 to identify the chromosome compartments.

In our study, we decompose the process of PCA computation by steps and propose that in the chromatin compartment analysis, there’s no need to explicitly compute the first principal component since all we need is the sign-pattern of PC1 for the A and B compartment identification. We use a fast and heuristic approach to approximate the sign-pattern of PC1 without doing PCA on the Pearson correlation matrix, and we’ll further illustrate that the relative-magnitude of our approximation almost perfectly matches the PC1 if both vectors are Z-score normalized. According to our PC1-pattern approximation approach, we then discuss the intuition behind the reason why the sign-pattern of PC1 will nearly correspond to the sign-pattern of the Pearson correlation matrix. Besides, we also compare the explained variance of PC1 to PC3 for GM12878 and K562 cells at 1Mb and 100Kb to point out that the percentage of variance explained by the principal components might also be considered.

## **Materials and Methods**

### **Datasets**

We used the GM12878 and K562 human cells provided by Rao S, Huntley M, and Lieberman Aiden E [20][21], with the support of Juicer [19] for calculating the Pearson correlation matrix and the first principal component as the ground truth, directly through the preprocessed *.hic* files:

* *GSE63525\_GM12878\_insitu\_primary+replicate\_combined\_30.hic*
* *GSE63525\_K562\_combined\_30.hic*

All the matrices are KR normalized by the following command lines:

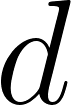
* Pearson correlation matrix

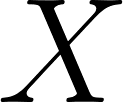
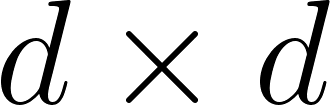
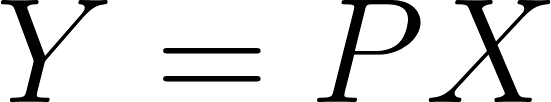
*$ java -Xmx13824m -Xms13824m -jar juicer\_tools.jar pearsons KR GSE63525\_GM12878\_insitu\_primary\_replicate\_combined\_30.hic 1 BP 1000000 pearsons\_1Mb.txt -p*

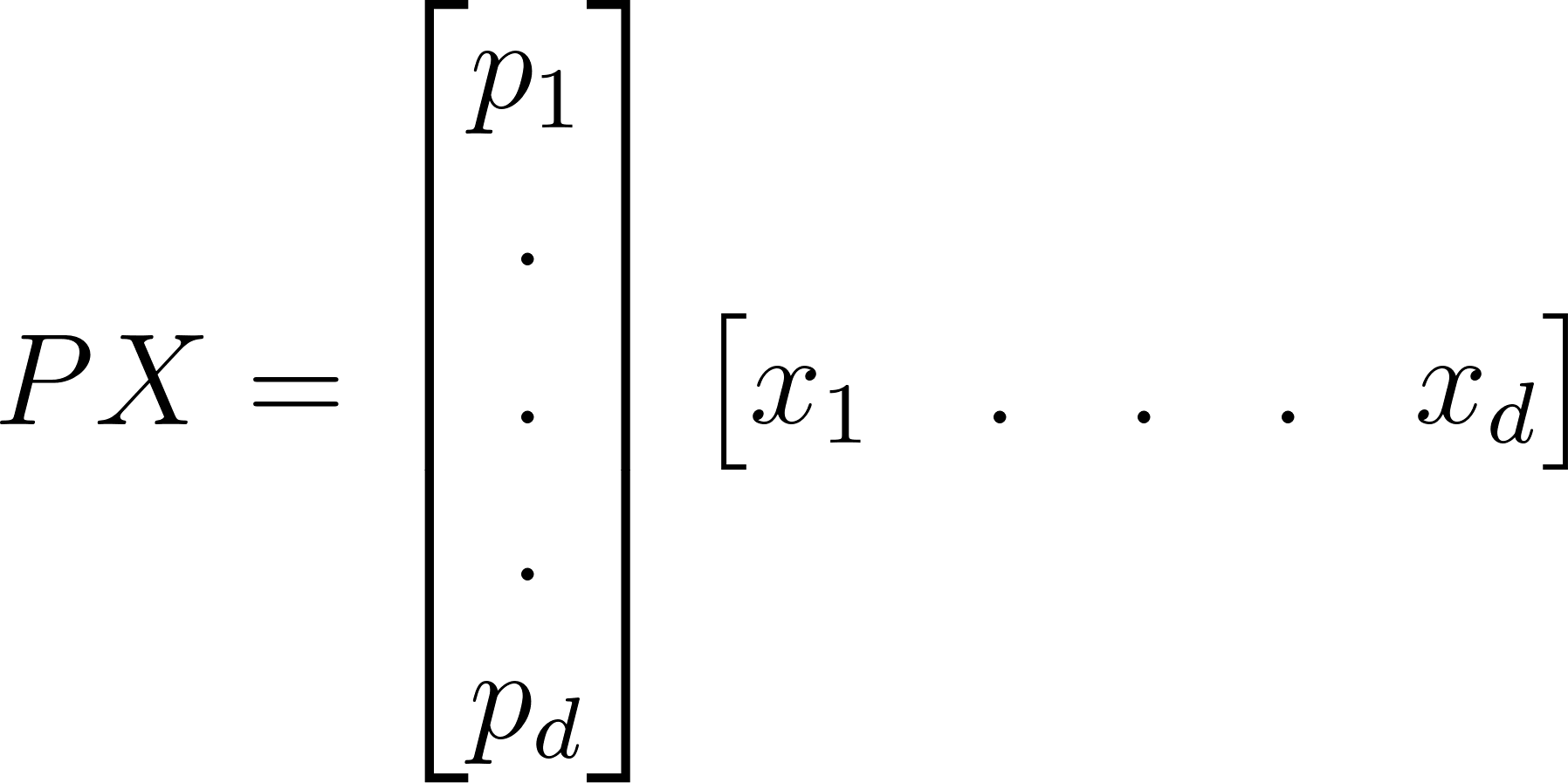
* First principal component

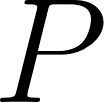
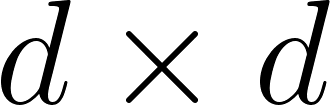
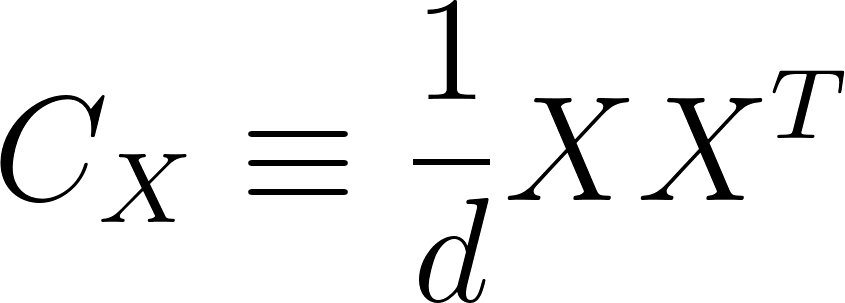
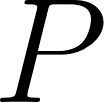
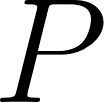
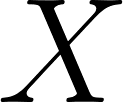
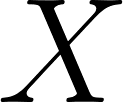
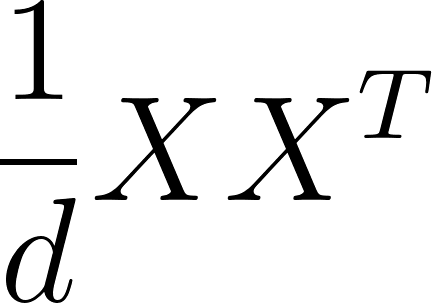
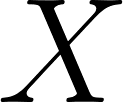
*$ java -Xmx13824m -Xms13824m -jar juicer\_tools.jar eigenvector KR GSE63525\_GM12878\_insitu\_primary\_replicate\_combined\_30.hic 1 BP 1000000 pc1\_1Mb.txt -p*

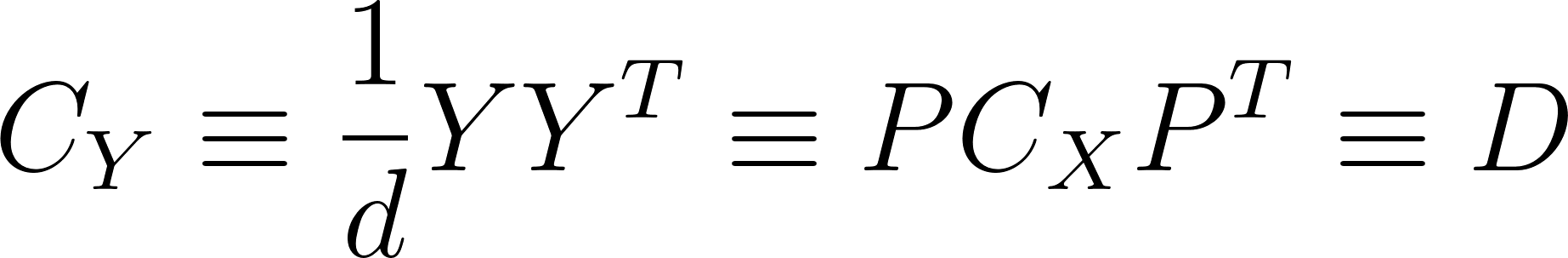
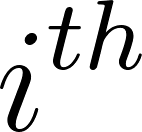
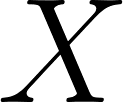
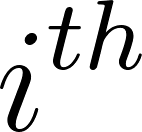
### **PC1-pattern Approximation Algorithm**

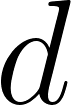
By defining [](https://www.codecogs.com/eqnedit.php?latex=d#0) as the number of genomics bins according to the matrix resolution, the derivation of PCA from Jonathon Shlens, 2014 (page 6, section V) [24] can be adjusted as follows:

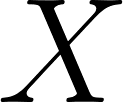
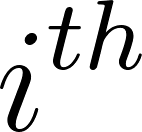
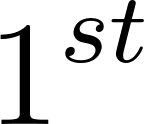
1. Set [](https://www.codecogs.com/eqnedit.php?latex=X#0) as the [](https://www.codecogs.com/eqnedit.php?latex=d%20%5Ctimes%20d#0) Pearson correlation matrix, set [](https://www.codecogs.com/eqnedit.php?latex=Y%20%3D%20PX#0) such that

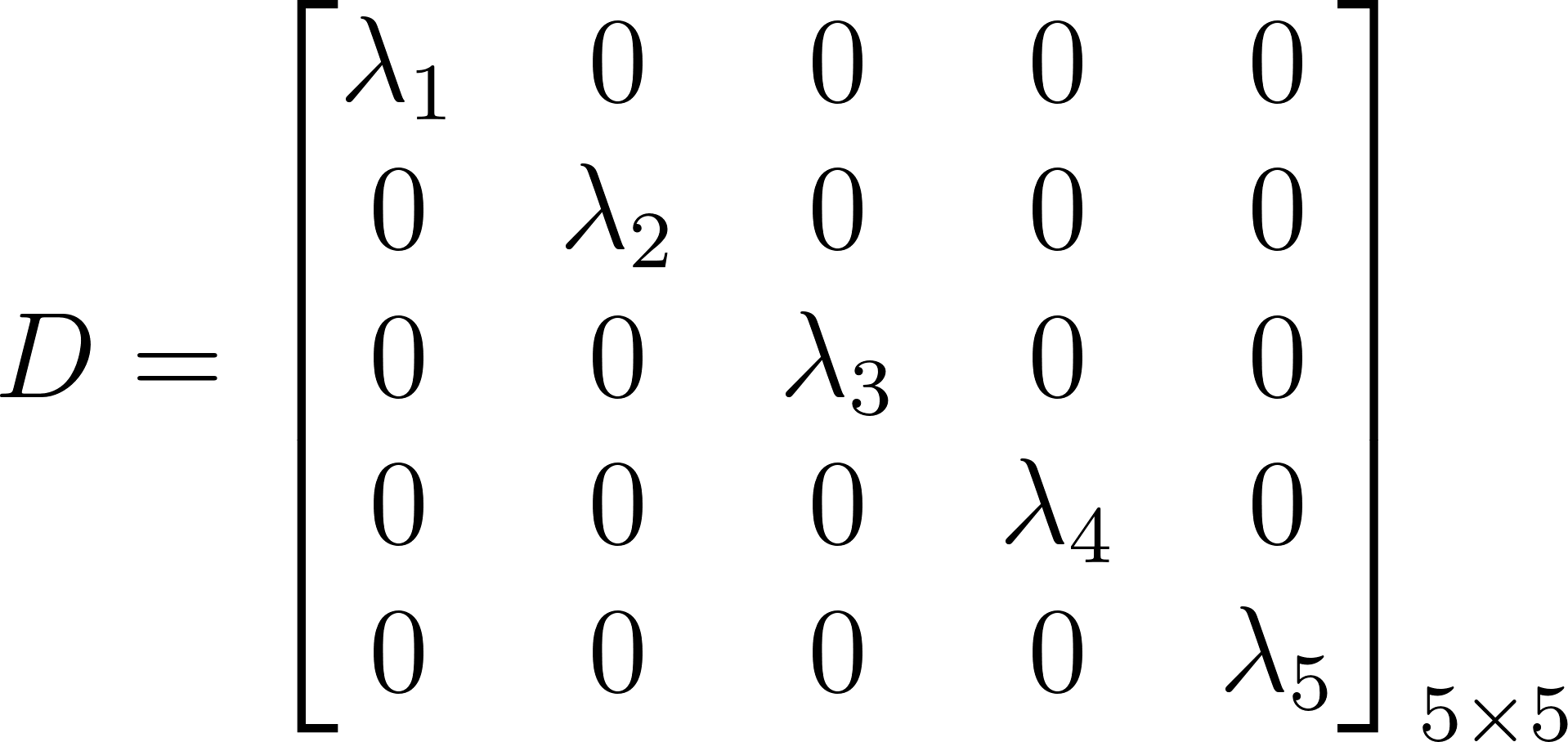
[](https://www.codecogs.com/eqnedit.php?latex=PX%20%3D%20%5Cbegin%7Bbmatrix%7D%20p_%7B1%7D%5C%5C%5C%5C%20.%5C%5C%5C%5C%20.%5C%5C%5C%5C%20.%5C%5C%5C%5C%20p_%7Bd%7D%20%5Cend%7Bbmatrix%7D%20%5Cbegin%7Bbmatrix%7Dx_%7B1%7D%20%26%20.%20%26%20.%20%26%20.%20%26%20x_%7Bd%7D%20%5Cend%7Bbmatrix%7D#0)

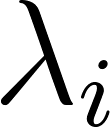
, where the selected [](https://www.codecogs.com/eqnedit.php?latex=P#0) is a [](https://www.codecogs.com/eqnedit.php?latex=d%20%5Ctimes%20d#0) orthonormal matrix that diagonalizes the covariance matrix [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7D%20%5Cequiv%20%5Cfrac%7B1%7D%7Bd%7DXX%5E%7BT%7D#0) (The trick is to set [](https://www.codecogs.com/eqnedit.php?latex=P#0) as the eigenvectors of [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7D#0) , which implies the rows of [](https://www.codecogs.com/eqnedit.php?latex=P#0) are the principal components of [](https://www.codecogs.com/eqnedit.php?latex=X#0)). Note that the mean for each dimension of the Pearson correlation matrix will be subtracted off before assigning to [](https://www.codecogs.com/eqnedit.php?latex=X#0), which is also called “zero-means”, makes the [](https://www.codecogs.com/eqnedit.php?latex=%5Cfrac%7B1%7D%7Bd%7DXX%5E%7BT%7D#0)mathematically equals to the covariance matrix of [](https://www.codecogs.com/eqnedit.php?latex=X#0).

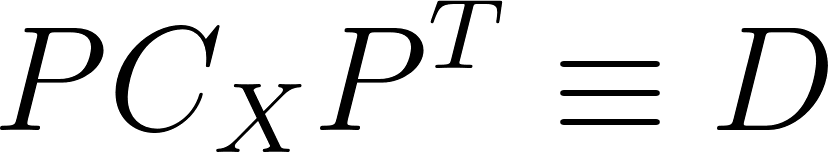
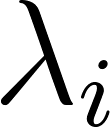
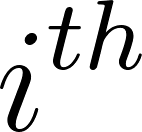
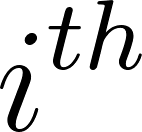
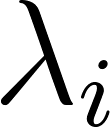
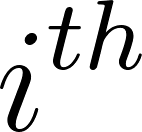
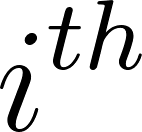
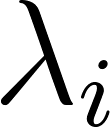
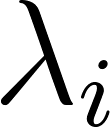
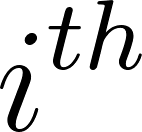
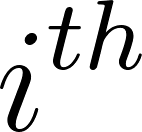
1. The covariance matrix [](https://www.codecogs.com/eqnedit.php?latex=C_%7BY%7D#0) can be rewritten as [](https://www.codecogs.com/eqnedit.php?latex=C_%7BY%7D%20%5Cequiv%20%5Cfrac%7B1%7D%7Bd%7DYY%5E%7BT%7D%20%5Cequiv%20PC_%7BX%7DP%5E%7BT%7D%20%5Cequiv%20D#0), where [](https://www.codecogs.com/eqnedit.php?latex=D#0) is a rank-ordered diagonal matrix. The [](https://www.codecogs.com/eqnedit.php?latex=i%5E%7Bth%7D#0) diagonal value of [](https://www.codecogs.com/eqnedit.php?latex=D#0) represent the the variance of [](https://www.codecogs.com/eqnedit.php?latex=X#0) along the [](https://www.codecogs.com/eqnedit.php?latex=i%5E%7Bth%7D#0) principal component.

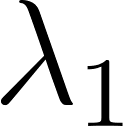
We further trace back the formula of [](https://www.codecogs.com/eqnedit.php?latex=C_%7BY%7D%20%5Cequiv%20PC_%7BX%7DP%5E%7BT%7D%20%5Cequiv%20D#0) from end to start by a “reverse derivation”. Here, we illustrate our idea with an example (Assume the number of genomics bins [](https://www.codecogs.com/eqnedit.php?latex=d#0) is 5):

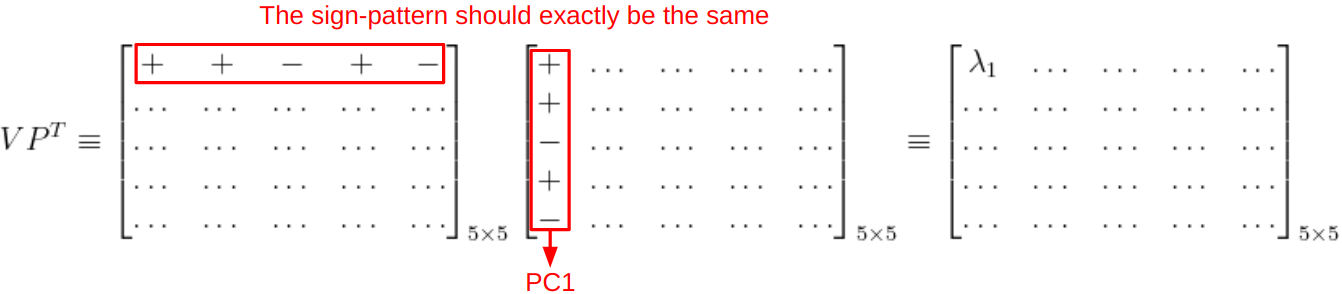
1. First, we already knew the diagonal matrix [](https://www.codecogs.com/eqnedit.php?latex=D#0) represents the variance of [](https://www.codecogs.com/eqnedit.php?latex=X#0) along the [](https://www.codecogs.com/eqnedit.php?latex=i%5E%7Bth%7D#0) principal components. Besides, the goal of PCA is to successively find the local maximum for each diagonal value in the order from the [](https://www.codecogs.com/eqnedit.php?latex=1%5E%7Bst%7D#0) principal component to the [](https://www.codecogs.com/eqnedit.php?latex=d%5E%7Bth%7D#0) , and make sure all PCs are orthogonal to each other [25][26]. We set [](https://www.codecogs.com/eqnedit.php?latex=D#0) as

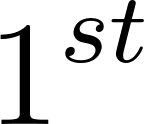
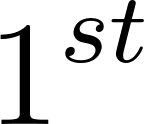
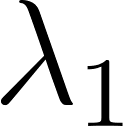
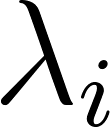
[](https://www.codecogs.com/eqnedit.php?latex=D%20%3D%20%5Cbegin%7Bbmatrix%7D%5Clambda%20_%7B1%7D%20%26%200%20%26%200%20%26%200%20%26%200%20%5C%5C%5C%5C%200%20%26%20%5Clambda%20_%7B2%7D%26%200%20%26%200%20%26%200%20%5C%5C%5C%5C%200%20%26%200%20%26%20%5Clambda%20_%7B3%7D%26%200%20%26%200%20%5C%5C%5C%5C%200%20%26%200%20%26%200%20%26%20%5Clambda%20_%7B4%7D%20%26%200%20%5C%5C%5C%5C%200%20%26%200%20%26%200%20%26%200%20%26%20%5Clambda%20_%7B5%7D%20%5Cend%7Bbmatrix%7D_%7B5%20%5Ctimes%205%7D#0)

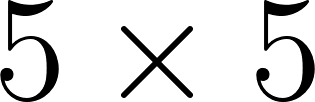
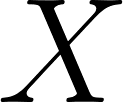
, where each of the [](https://www.codecogs.com/eqnedit.php?latex=%5Clambda_%7Bi%7D#0) is at their maximum (i.e., there is nothing larger).

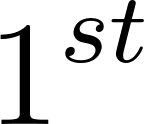
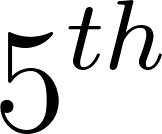
1. We define [](https://www.codecogs.com/eqnedit.php?latex=V#0) as the matrix multiplication of [](https://www.codecogs.com/eqnedit.php?latex=PC_%7BX%7D#0), and consider the relationship between [](https://www.codecogs.com/eqnedit.php?latex=V#0) and [](https://www.codecogs.com/eqnedit.php?latex=P%5E%7BT%7D#0) in [](https://www.codecogs.com/eqnedit.php?latex=PC_%7BX%7DP%5E%7BT%7D%20%5Cequiv%20D#0). Since we have known each of the [](https://www.codecogs.com/eqnedit.php?latex=%5Clambda_%7Bi%7D#0) is at their maximum, the sign-pattern of the [](https://www.codecogs.com/eqnedit.php?latex=i%5E%7Bth%7D#0) row of [](https://www.codecogs.com/eqnedit.php?latex=V#0)and the sign-pattern of the [](https://www.codecogs.com/eqnedit.php?latex=i%5E%7Bth%7D#0) column of [](https://www.codecogs.com/eqnedit.php?latex=P%5E%7BT%7D#0) should be the same (i.e., the matrix multiplication of these two vectors will derive the maximum [](https://www.codecogs.com/eqnedit.php?latex=%5Clambda_%7Bi%7D#0)). This statement can be proved by contradiction:
2. If the sign-pattern of the [](https://www.codecogs.com/eqnedit.php?latex=i%5E%7Bth%7D#0) row of [](https://www.codecogs.com/eqnedit.php?latex=V#0)and the sign-pattern of the [](https://www.codecogs.com/eqnedit.php?latex=i%5E%7Bth%7D#0) column of [](https://www.codecogs.com/eqnedit.php?latex=P%5E%7BT%7D#0) are NOT the same, we declare the matrix multiplication [](https://www.codecogs.com/eqnedit.php?latex=%5Clambda_%7Bi%7D#0) is the maximum.
3. However, we can always find a larger [](https://www.codecogs.com/eqnedit.php?latex=%5Clambda_%7Bi%7D#0) by calculating the matrix multiplication of the [](https://www.codecogs.com/eqnedit.php?latex=i%5E%7Bth%7D#0) row of [](https://www.codecogs.com/eqnedit.php?latex=V#0)and the [](https://www.codecogs.com/eqnedit.php?latex=i%5E%7Bth%7D#0) column of [](https://www.codecogs.com/eqnedit.php?latex=P%5E%7BT%7D#0) with the same sign-pattern, since the multiplication of each entry-pairs is all positive and leads to a larger summation. Hence, we come across a contradiction.

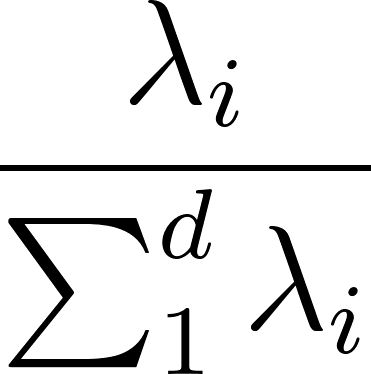
Here, we illustrate calculating the [](https://www.codecogs.com/eqnedit.php?latex=%5Clambda%20_%7B1%7D#0) (Figure 1):



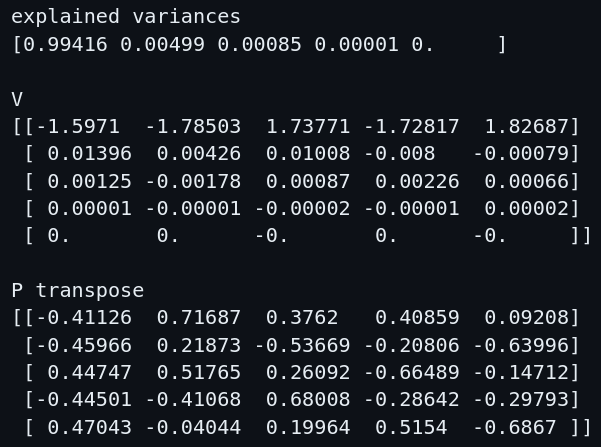
**Figure 1.** In this example, the [](https://www.codecogs.com/eqnedit.php?latex=1%5E%7Bst%7D#0) row of [](https://www.codecogs.com/eqnedit.php?latex=V#0) and the [](https://www.codecogs.com/eqnedit.php?latex=1%5E%7Bst%7D#0) column of [](https://www.codecogs.com/eqnedit.php?latex=P%5E%7BT%7D#0) (i.e. PC1) is for calculating the [](https://www.codecogs.com/eqnedit.php?latex=%5Clambda%20_%7B1%7D#0), the sign-pattern of these two vectors should be the same. The sign-pattern of the corresponding vectors in [](https://www.codecogs.com/eqnedit.php?latex=V#0) and [](https://www.codecogs.com/eqnedit.php?latex=P%5E%7BT%7D#0) for calculating [](https://www.codecogs.com/eqnedit.php?latex=%5Clambda_%7Bi%7D#0) is also identical vise-versa.

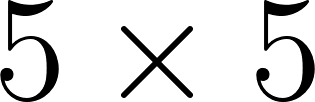
1. Next, as in the example below (Figure 2), we perform PCA on a simulated [](https://www.codecogs.com/eqnedit.php?latex=5%20%5Ctimes%205#0) Pearson correlation matrix [](https://www.codecogs.com/eqnedit.php?latex=X#0) from the O/E matrix in a NumPy format [27]:

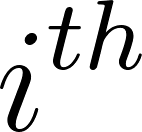
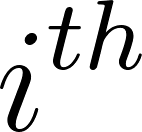
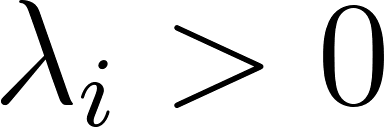
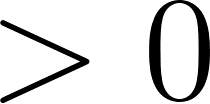
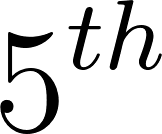
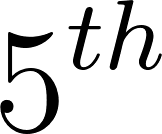
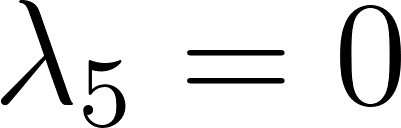
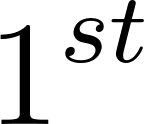
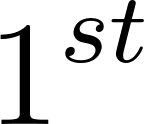
* The “explained variances” vector is for the [](https://www.codecogs.com/eqnedit.php?latex=1%5E%7Bst%7D#0) to the [](https://www.codecogs.com/eqnedit.php?latex=5%5E%7Bth%7D#0)principal component, from left to right. The formula for the explained variances is:

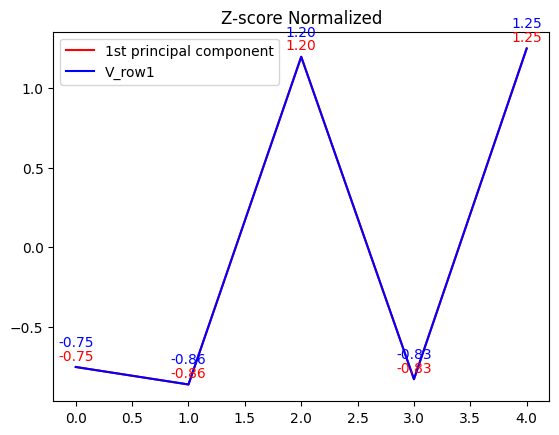
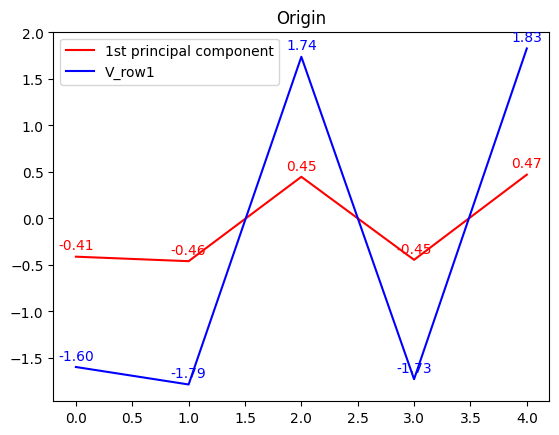
[](https://www.codecogs.com/eqnedit.php?latex=%5Cfrac%7B%5Clambda%20_%7Bi%7D%7D%7B%5Csum_%7B1%7D%5E%7Bd%7D%20%5Clambda%20_%7Bi%7D%7D#0)

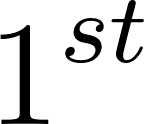
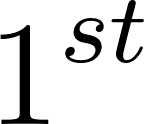
* The matrix format of [](https://www.codecogs.com/eqnedit.php?latex=V#0) and [](https://www.codecogs.com/eqnedit.php?latex=P%5ET#0) are consistent with the general mathematical expression.

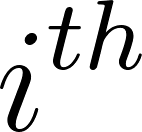
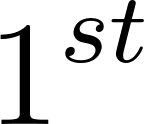
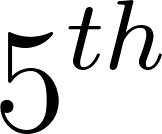
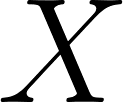
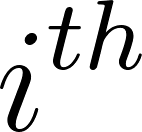
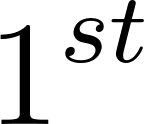
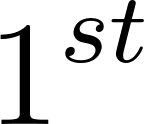
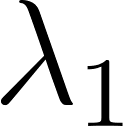


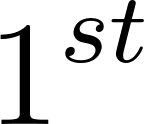
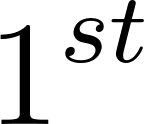
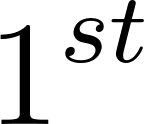
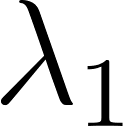
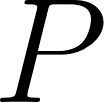
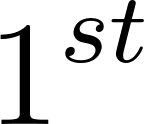
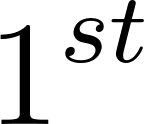
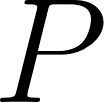
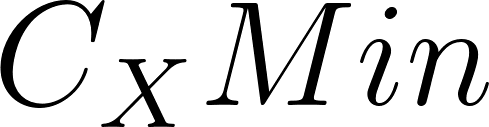
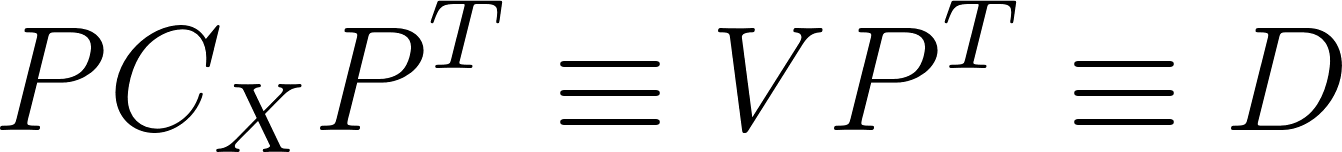
**Figure 2.** Simulated [](https://www.codecogs.com/eqnedit.php?latex=5%20%5Ctimes%205#0) [](https://www.codecogs.com/eqnedit.php?latex=V#0) and [](https://www.codecogs.com/eqnedit.php?latex=P%5ET#0) during the PCA, the explained variance showed on the top.

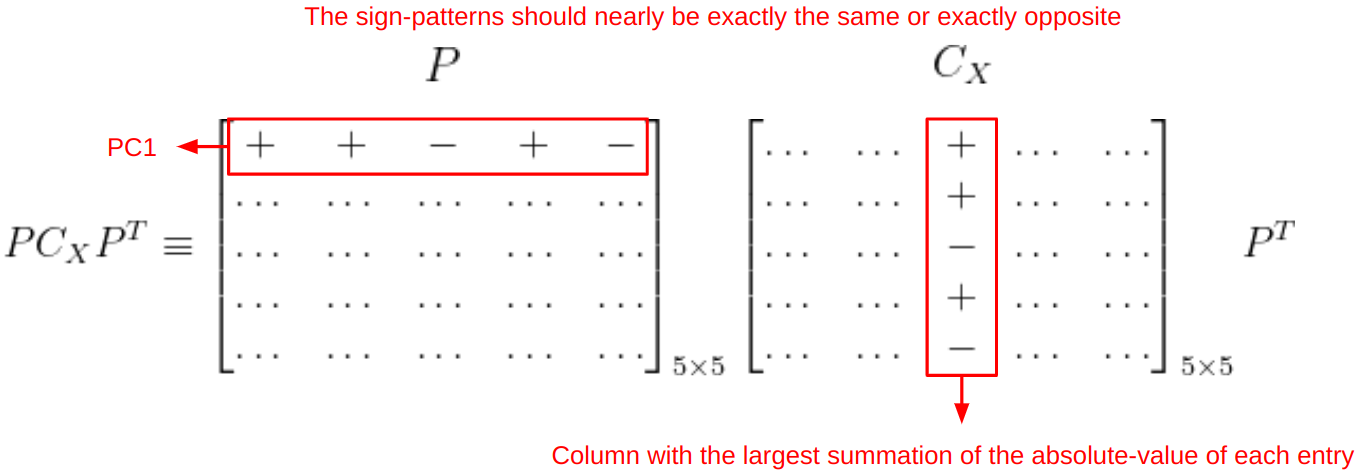
This example demonstrates that the sign-patterns of the [](https://www.codecogs.com/eqnedit.php?latex=i%5E%7Bth%7D#0) row in [](https://www.codecogs.com/eqnedit.php?latex=V#0) and [](https://www.codecogs.com/eqnedit.php?latex=i%5E%7Bth%7D#0) column in [](https://www.codecogs.com/eqnedit.php?latex=P%5ET#0) are identical, and the relative-magnitude are also similar if the corresponding [](https://www.codecogs.com/eqnedit.php?latex=%5Clambda_%7Bi%7D%20%3E%200#0) (i.e., the corresponding explained variance [](https://www.codecogs.com/eqnedit.php?latex=%3E0#0)). In the case of the [](https://www.codecogs.com/eqnedit.php?latex=5%5E%7Bth%7D#0) row in [](https://www.codecogs.com/eqnedit.php?latex=V#0) and [](https://www.codecogs.com/eqnedit.php?latex=5%5E%7Bth%7D#0) column in [](https://www.codecogs.com/eqnedit.php?latex=P%5ET#0), since the corresponding [](https://www.codecogs.com/eqnedit.php?latex=%5Clambda_%7B5%7D%20%3D%20%200#0) , the sign-patterns between these two vectors are somewhat arbitrary. The following plots visualize the comparison between the [](https://www.codecogs.com/eqnedit.php?latex=1%5E%7Bst%7D#0) row in [](https://www.codecogs.com/eqnedit.php?latex=V#0) and [](https://www.codecogs.com/eqnedit.php?latex=1%5E%7Bst%7D#0) column in [](https://www.codecogs.com/eqnedit.php?latex=P%5ET#0) (i.e., PC1), without and with the Z-score normalization (Figure 3):

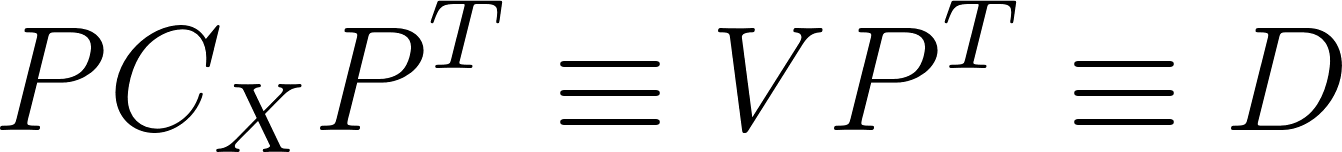


**Figure 3.** The plot on the left shows both the sign-pattern and relative-magnitude of the [](https://www.codecogs.com/eqnedit.php?latex=1%5E%7Bst%7D#0) row of [](https://www.codecogs.com/eqnedit.php?latex=V#0) (blue line) versus PC1 (red line) in Figure 2, with the values rounded to the second decimal. The plot on the right shows the Z-score normalized plot of these two vectors. We can see that the patterns of the [](https://www.codecogs.com/eqnedit.php?latex=1%5E%7Bst%7D#0) row in [](https://www.codecogs.com/eqnedit.php?latex=V#0) and PC1 are nearly the same. In the Z-score normalized plot, they are even overlapped.

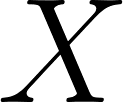
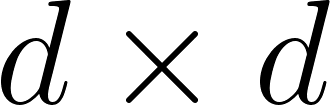
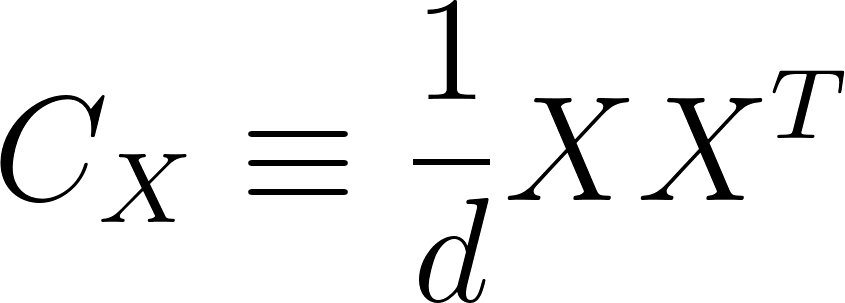
Moreover, the relative-magnitude between rows in [](https://www.codecogs.com/eqnedit.php?latex=V#0) are rank-ordered in a descending manner, corresponding to the [](https://www.codecogs.com/eqnedit.php?latex=i%5E%7Bth%7D#0) principal component (i.e., the [](https://www.codecogs.com/eqnedit.php?latex=1%5E%7Bst%7D#0) row has the highest magnitude, and the [](https://www.codecogs.com/eqnedit.php?latex=5%5E%7Bth%7D#0) row has the lowest magnitude). On the contrary, there’s no fixed pattern for the relative-magnitude between columns in [](https://www.codecogs.com/eqnedit.php?latex=P%5E%7BT%7D#0). We hypothesize that [](https://www.codecogs.com/eqnedit.php?latex=V#0) the key factor influences the diagonal values in [](https://www.codecogs.com/eqnedit.php?latex=D#0) (i.e., the variance of [](https://www.codecogs.com/eqnedit.php?latex=X#0) along the [](https://www.codecogs.com/eqnedit.php?latex=i%5E%7Bth%7D#0) PC), as our interest in the [](https://www.codecogs.com/eqnedit.php?latex=1%5E%7Bst%7D#0) principal component, we then consider the rationale behind how the [](https://www.codecogs.com/eqnedit.php?latex=1%5E%7Bst%7D#0) row of [](https://www.codecogs.com/eqnedit.php?latex=V#0) makes the corresponded [](https://www.codecogs.com/eqnedit.php?latex=%5Clambda_%7B1%7D#0) the maximum.

1. Based on our intuition, each entry of the [](https://www.codecogs.com/eqnedit.php?latex=1%5E%7Bst%7D#0) row in [](https://www.codecogs.com/eqnedit.php?latex=V#0) should be at its maximum or minimum, hence the matrix multiplication of the [](https://www.codecogs.com/eqnedit.php?latex=1%5E%7Bst%7D#0) row in [](https://www.codecogs.com/eqnedit.php?latex=V#0) and the [](https://www.codecogs.com/eqnedit.php?latex=1%5E%7Bst%7D#0) column in [](https://www.codecogs.com/eqnedit.php?latex=P%5E%7BT%7D#0)will lead to the maximum [](https://www.codecogs.com/eqnedit.php?latex=%5Clambda_%7B1%7D#0) (Since these two vectors have the same sign-patterns, the matrix multiplication of these two vectors will only have a positive effect on the total variance. i.e., The positive-positive and negative-negative multiplication of each entry-pair will only produce a positive result). Besides, we consider the proportion of each entry in [](https://www.codecogs.com/eqnedit.php?latex=V#0) will “contribute” to the total variance; the most-contributed entry is the largest absolute-value entry.
2. We assume that we can trace back the sign-pattern of the first principal component with the most-contributed entry in [](https://www.codecogs.com/eqnedit.php?latex=V#0), hence we move backward to consider the matrix multiplication of [](https://www.codecogs.com/eqnedit.php?latex=P#0) and [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7D#0). To produce a maximum or minimum most-contributed entry in the [](https://www.codecogs.com/eqnedit.php?latex=1%5E%7Bst%7D#0) row [](https://www.codecogs.com/eqnedit.php?latex=V#0), we conclude that the sign-pattern of [](https://www.codecogs.com/eqnedit.php?latex=1%5E%7Bst%7D#0) row [](https://www.codecogs.com/eqnedit.php?latex=P#0) (i.e., PC1) should nearly be the same or exactly opposite with a specific column of [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7D#0), in which the summation of the absolute-value of each entry in this specific column is the largest compare to the other columns of [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7D#0) (We define it as “[](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7DMax#0)”), for the reason that the matrix multiplication of these two vectors should contribute the largest amount to the total variance; on the contrary, the column of [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7D#0) with the smallest summation of the absolute-value of each entry (We define it as “[](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7DMin#0)”) might poorly match the PC1 sign-pattern. As a result, since our target is only to figure out the sign-pattern of PC1 for the compartment identification, there’s no need to perform the PCA but to utilize the [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7D#0) to get the approximated sign-pattern of PC1. Here, we provide an illustration for [](https://www.codecogs.com/eqnedit.php?latex=PC_%7BX%7DP%5E%7BT%7D%20%5Cequiv%20VP%5E%7BT%7D%20%5Cequiv%20D#0) in our conclusion (Figure 4):



**Figure 4.** An illustration for concluding our idea in calculating [](https://www.codecogs.com/eqnedit.php?latex=PC_%7BX%7DP%5E%7BT%7D%20%5Cequiv%20VP%5E%7BT%7D%20%5Cequiv%20D#0) shows that we don’t have to use PCA to get the sign-pattern of PC1 explicitly.

With the explanation above, we conclude that there’s no need to perform PCA to get the approximate sign-pattern of PC1. Besides, we also found a similar relative-magnitude between our approximation and the actual PC1. We define the approximation as the “approximated PC1-patten”, here we list our algorithm applying to the chromatin compartments analysis:

1. Set [](https://www.codecogs.com/eqnedit.php?latex=X#0) as the [](https://www.codecogs.com/eqnedit.php?latex=d%20%5Ctimes%20d#0) zero-means Pearson correlation matrix derived from the Hi-C O/E matrix, according to the matrix resolution (Note that the zero-means process is needed since we assume [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7D%20%5Cequiv%20%5Cfrac%7B1%7D%7Bd%7DXX%5E%7BT%7D#0), without this step, the result approximated PC1-patten will have a slight difference).
2. Construct the covariance matrix [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7D#0), calculate the summation of the absolute-value of each entry for each column in [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7D#0), and select [](https://www.codecogs.com/eqnedit.php?latex=C_%7Bx%7DMax#0) as the approximated PC1-patten.
3. Inspect the GC-content distribution of the chromosome corresponding to the approximated PC1-patten, and flip the sign of the approximated PC1-patten if needed (In practice, we tend to label the positive-sign entries as the A compartments, which are usually GC-content rich [16]).
4. Perform Z-score normalization to the approximated PC1-pattern (Optional).

In our experiments, we tested the correctness of our approximated PC1-pattern algorithm with Juicer’s PC1 on GM12878 and K562 human cells for all 23 chromosomes at the resolutions of 100 Kb and 25 Kb. For the comparison, we flipped the sign of the approximated PC1-pattern if the Pearson correlation coefficient between the approximated PC1-pattern and Juicer’s PC1 is negative, counted the entries with the same sign in these two vectors, and summarized the correction rate for our algorithm. Besides, we performed the Z-score normalization on both vectors for better visualization and also produced heatmaps by *seaborn* python package [28] to compare the difference between matrices during the algorithm.

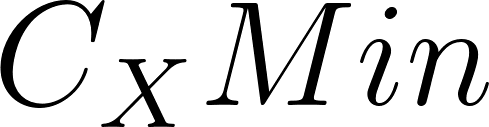
### **Explained Variance Comparison**

We explored the explained variances for the top three principal components on GM12878 and K562 Pearson correlation matrix at the resolution of 1Mb and 100Kb. All the *NaN* values in Juicer’s outputs were removed in advance, and the PCA was performed with the support of the *Scikit-learn* [29] and *Pandas* [30] Python packages.

## **Results**

### **PC1-pattern Approximation Results**

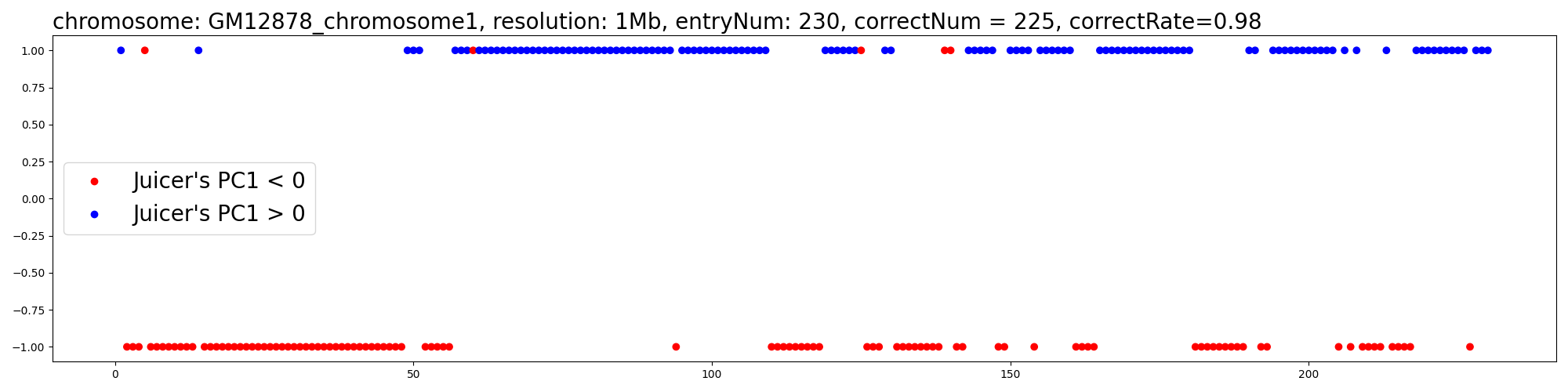
Figure 5 shows the plot results of the PC1-pattern approximation with the GM12878 and K562 zero-mean Pearson correlation matrix on chromosome 1, compared with the Juicer’s PC1. Due to a finer resolution of track, which will need a larger plot for visualization, we only show the scatter plots and line plots at the resolution of 1Mb for demonstration.

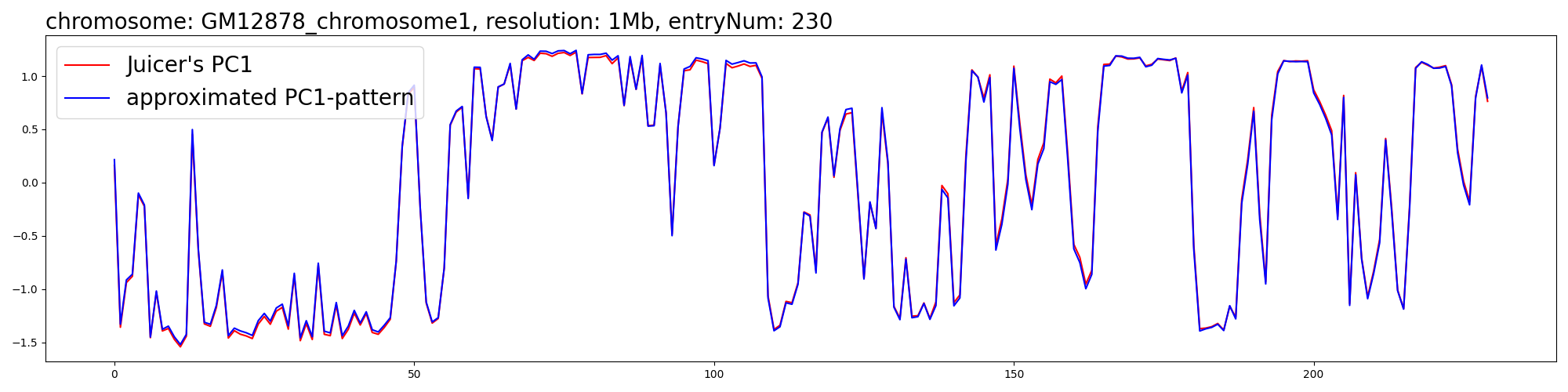
We can see that in Figure 5 (A) and Figure 5 (C), both the sign-pattern and relative-magnitude of the approximated PC1-pattern and Juicer’s PC1 are nearly perfectly matched for GM12878 and K562, and the correct rate is also pretty well based on [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7DMax#0). On the other hand, in Figure 5 (B) and Figure 5 (D) although the trend of relative-magnitude in the line plots looks alike, they still don't overlap with each other and have poor correct rates in the scatter plots based on [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7DMin#0). We further quantify the correctness of the sign-pattern:

[](https://www.codecogs.com/eqnedit.php?latex=Correct%5C%3Brate%3D%5Cfrac%7BNumber%5C%3Bof%5C%3Bentries%5C%3Bin%5C%3Bthe%5C%3Bapproximated%5C%3BPC1%20%5Ctextendash%20pattern%5C%3Bwhich%5C%3Bhas%5C%3Bthe%5C%3Bsame%5C%3Bsign%5C%3Bas%5C%3Bthe%5C%3BJuicer's%5C%3BPC1%7D%7BNumber%5C%3Bof%5C%3Bthe%5C%3Btotal%5C%3Bentries%5C%3Bin%5C%3Bthe%5C%3Bapproximated%5C%3BPC1%20%5Ctextendash%20pattern%7D#0)

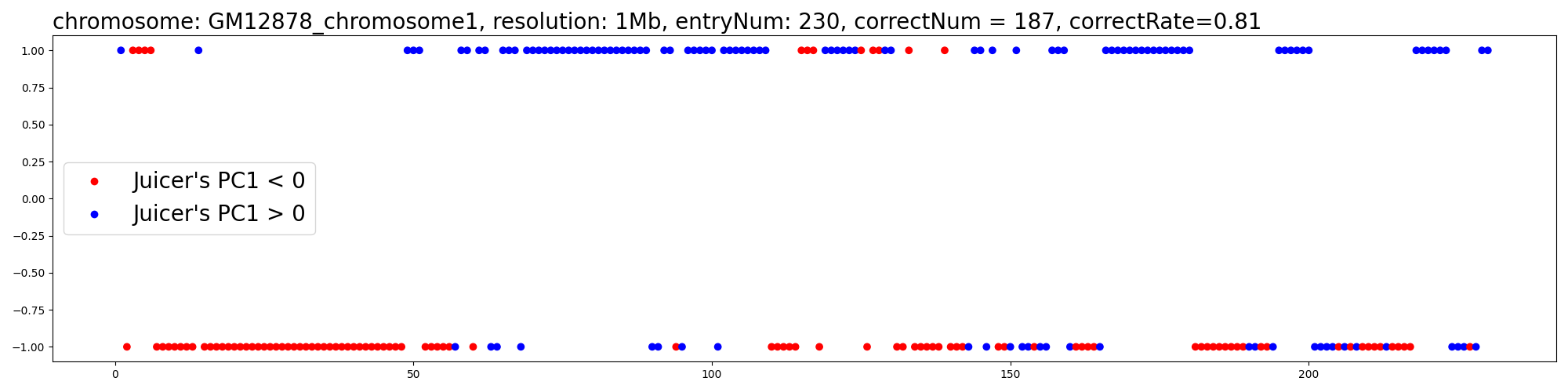
If the sign-pattern of the approximated PC1-pattern has the same sign as the Juicer’s PC1, the blue dots have a value of 1 or the red dots with -1. The correct rate of the scatter plot for (A) and (B) is 0.98 and 0.81, and for (C) and (D) is 0.98 and 0.68, respectively. These results seem to indicate our assumption is correct. We then further performed the same experiments for GM12878 and K562 at the resolution of 100 Kb and 25 Kb, and summarized the results for all 23 chromosomes in Table 1.

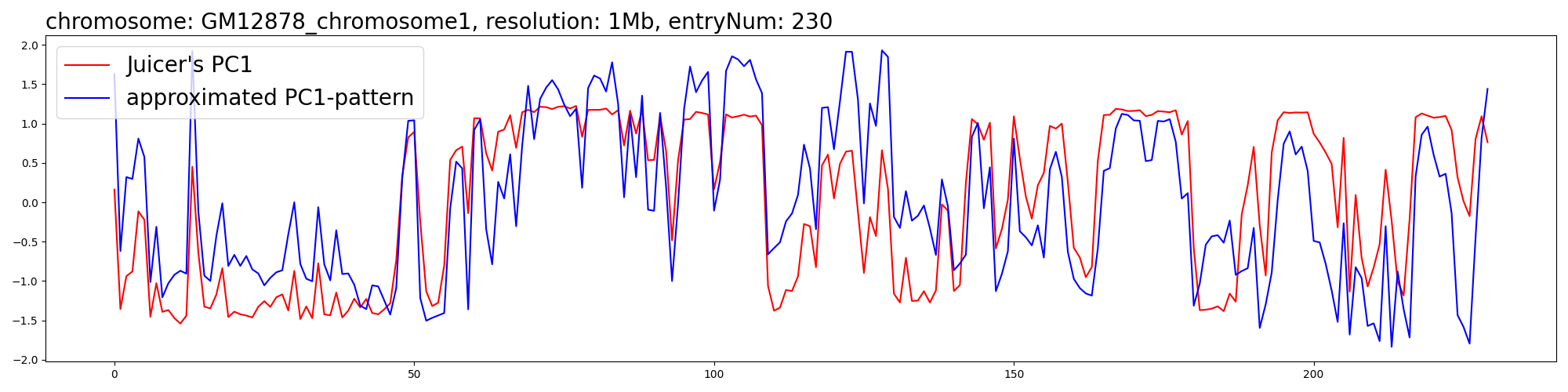
**(A)**



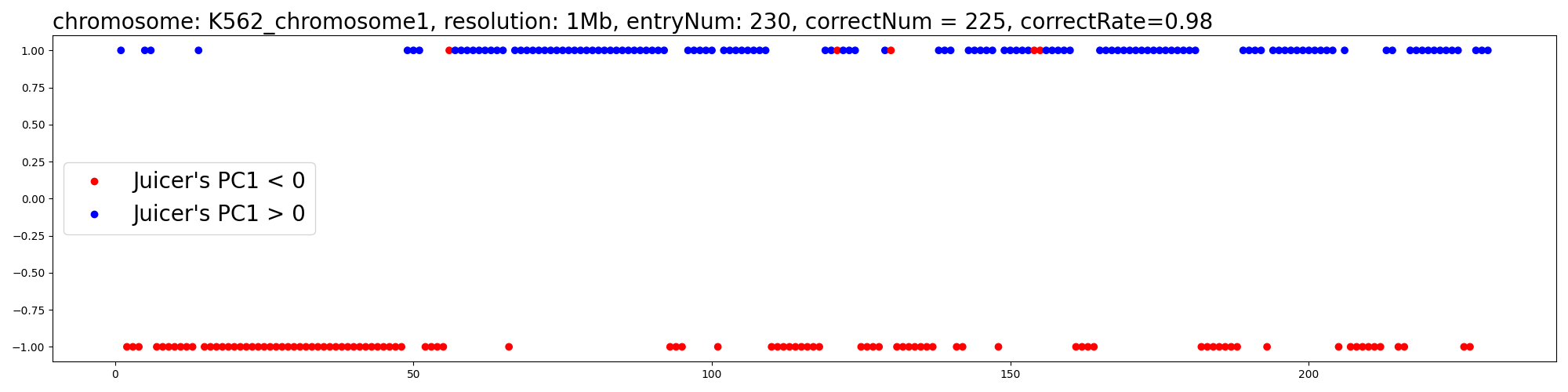


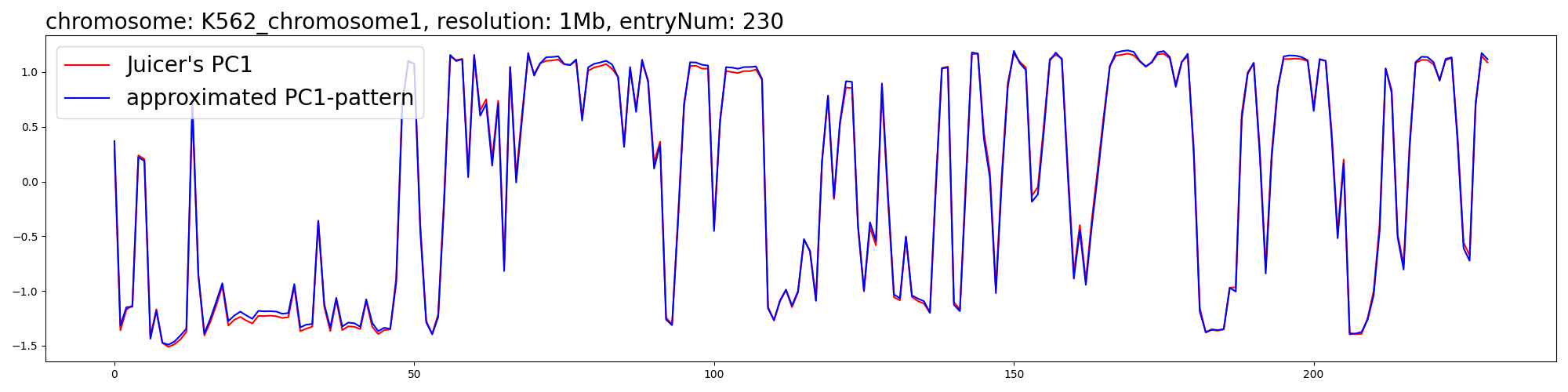
**(B)**



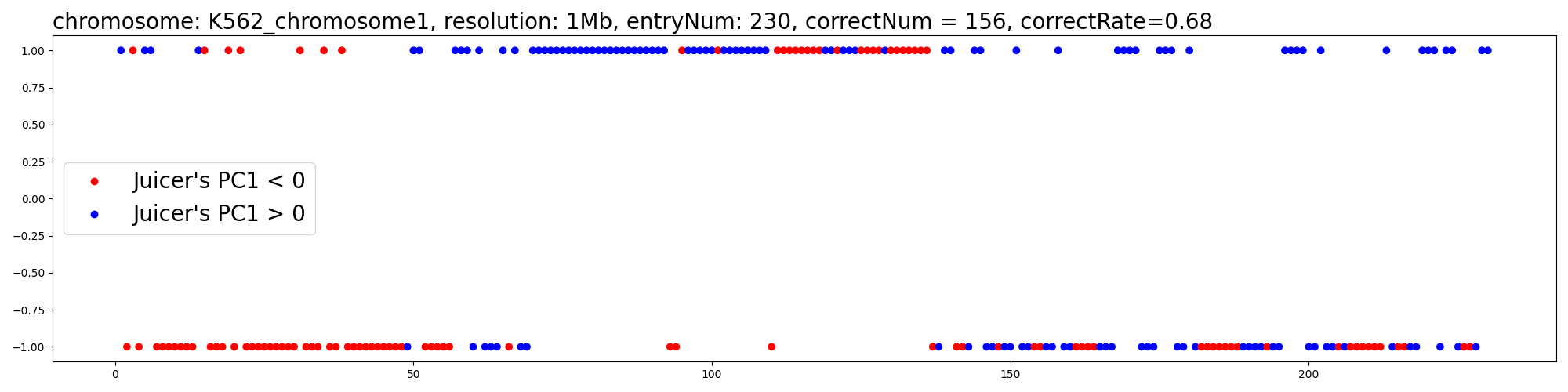


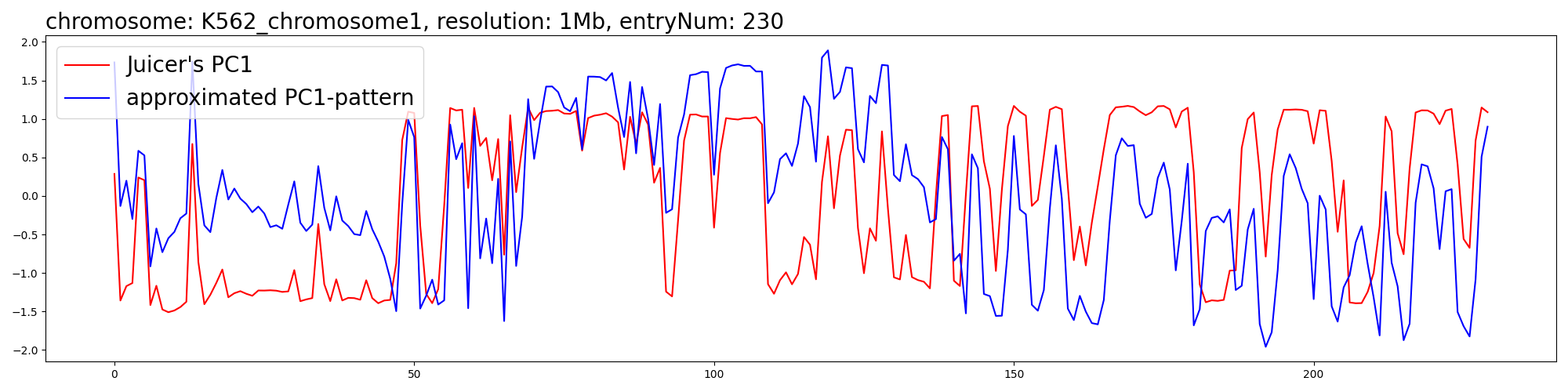
**(C)**

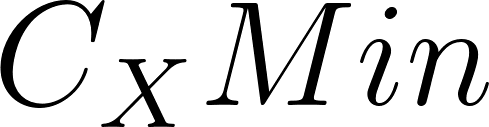
****

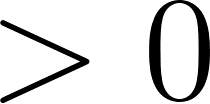
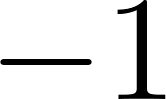
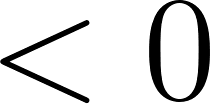
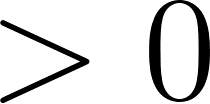
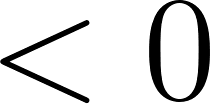
****

**(D)**

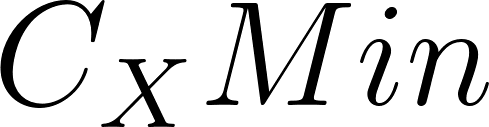
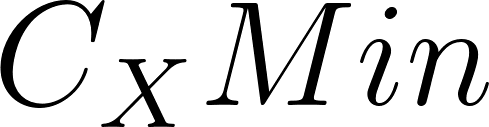
****

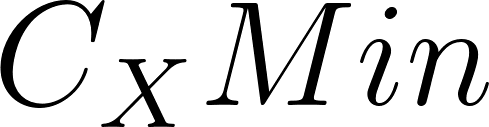
****

**Figure 5.** The scatter plots and line plots for the PC1-pattern approximation comparison with the GM12878 (A) (B) and K562 (C) (D) on chromosome 1, at the resolution of 1Mb. In (A) and (C), we select [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7DMax#0) as our approximated PC1-pattern. For the comparison, (B) and (D) [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7DMin#0) are selected. In the scatter plot, we compare the raw track of the approximated PC1-pattern with Juicer’s PC1 without normalization. All the genomics bins (i.e., entries) are shown as dots. For each dot:

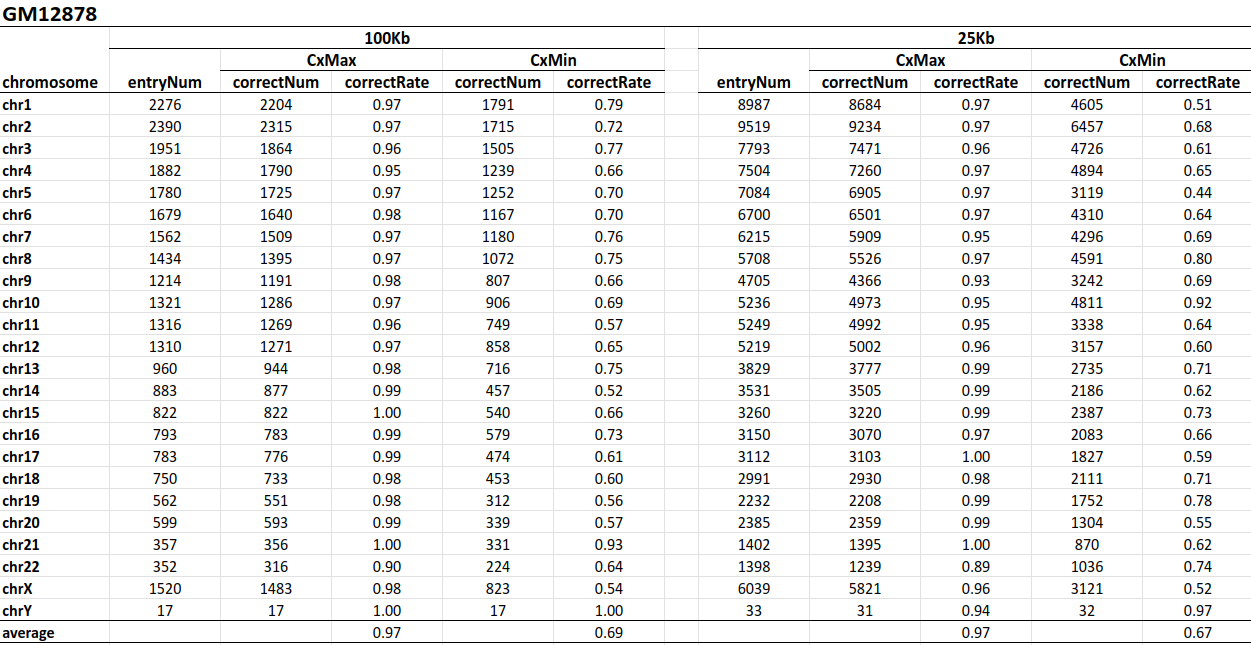
* The value is set as [](https://www.codecogs.com/eqnedit.php?latex=1#0) if the corresponding approximated PC1-pattern entry [](https://www.codecogs.com/eqnedit.php?latex=%3E0#0) , is set as [](https://www.codecogs.com/eqnedit.php?latex=-1#0) if entry [](https://www.codecogs.com/eqnedit.php?latex=%3C0#0).
* The color is set as blue if the corresponding Juicer’s PC1 entry [](https://www.codecogs.com/eqnedit.php?latex=%3E0#0) , is set as red if entry [](https://www.codecogs.com/eqnedit.php?latex=%3C0#0).

In the line plot, we normalize the Z-score of both the approximated PC1-pattern and Juicer’s PC1, and compare the relative-magnitude between these two tracks.

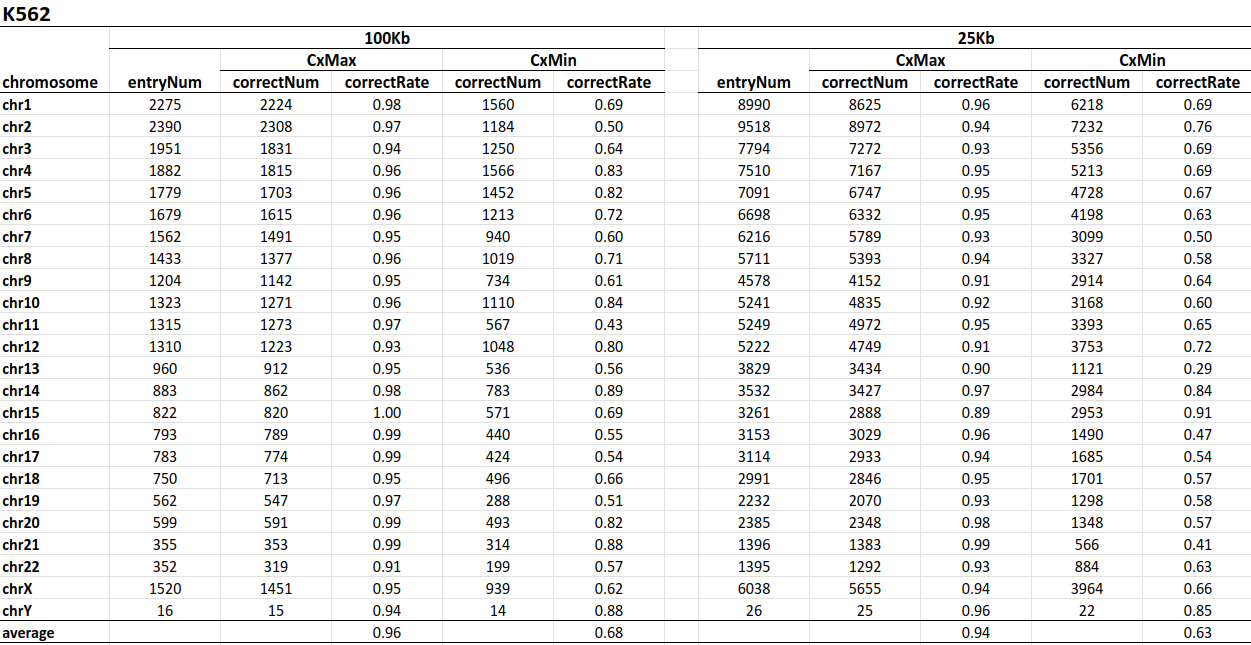
In Table 1, we compare the difference between selecting the [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7DMax#0) or [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7DMin#0) as the approximated PC1-pattern; the correct rates for all the 23 chromosomes in GM12878 and K562 are pretty good if we choose the [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7DMax#0) (Both higher than 0.94 on average at 100 Kb and 25 Kb, for GM12878 and K562), and not ideal if we choose the [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7DMin#0) (Both lower than 0.69 in average at 100 Kb and 25 Kb, for GM12878 and K562). Note that our algorithm is extremely fast. In the experiments on both GM12878 and K562, it takes less than 1 minute to finish the PC1-pattern approximation for all the 23 chromosomes at 100 Kb and about 5 minutes at 25 Kb, respectively (Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz, 16GiB RAM).

**Table 1.** The summarization of the difference between selecting the [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7DMax#0) or [](https://www.codecogs.com/eqnedit.php?latex=C_%7BX%7DMin#0) as the approximated PC1-pattern, in GM12878 (A) and K562 (B) at the resolution of 100 Kb and 25 Kb. The correct rates are all rounded to the second decimal.

**(A)**

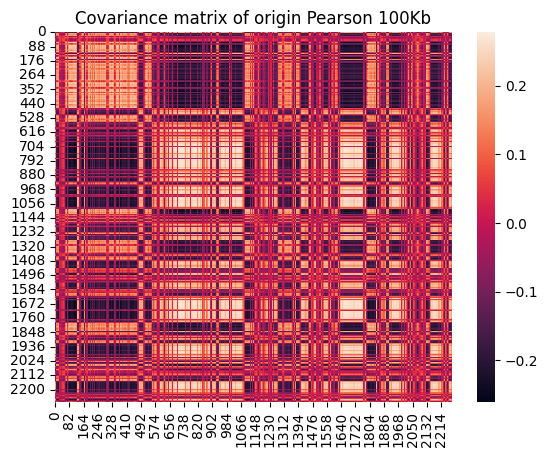
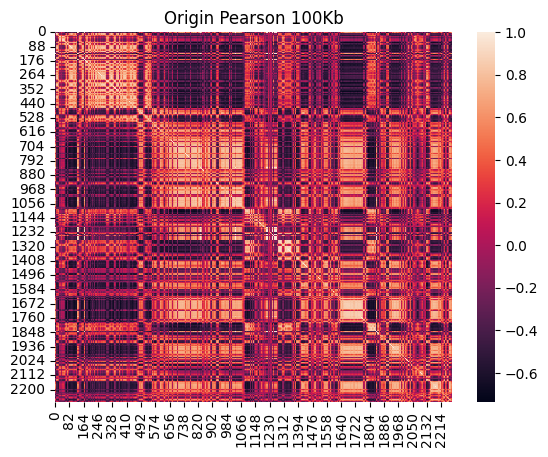


**(B)**

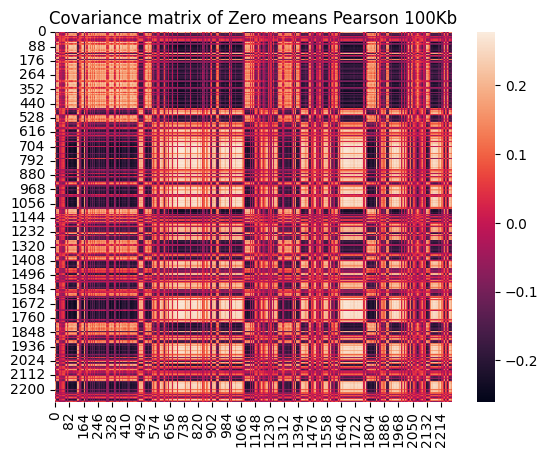
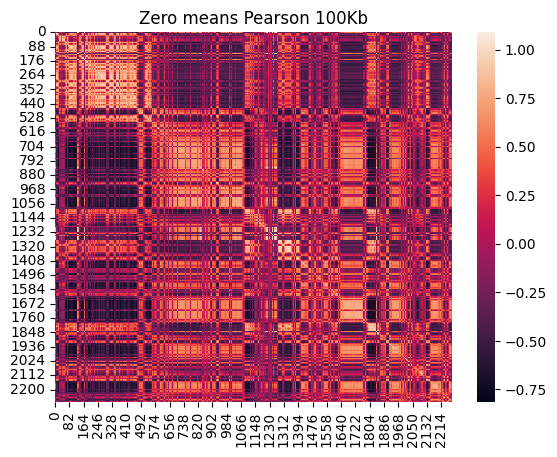
****

Finally, we compare the heatmaps of the GM12878 Pearson correlation matrix and the corresponding covariance matrix at 100 Kb resolution, with and without zero-means the Pearson matrix (Figure 6 A and B), in case some of the Hi-C tools won’t center the data before PCA (We find it will makes a slight difference on the PC1-pattern approximation). The heatmap-pattern between these matrices is almost identical. However, the signal in the covariance matrix might be slightly blurred, or we call it over “sharpened”, as the vocabulary used by Erez Lieberman-Aiden et al. [1]. These look-alike heatmaps reflect that our approximated PC1-pattern extracted from the covariance matrix can also fit the plaid-pattern on the origin Pearson matrix, and it might imply the reason why PC1 can reflect the compartment intervals should be explained mathematically. What’s more, since the Juicer’s PC1 can’t match well with the covariance of the Pearson matrix if the summation of the absolute-value of each entry for that column is small, the Juicer’s PC1 might hence not always match well in the origin Pearson matrix, too.

**(A)**



**(B)**



**Figure 6.** Heatmaps of the GM12878 at 100Kb without (A) and with (B) zero-means the Pearson matrix.

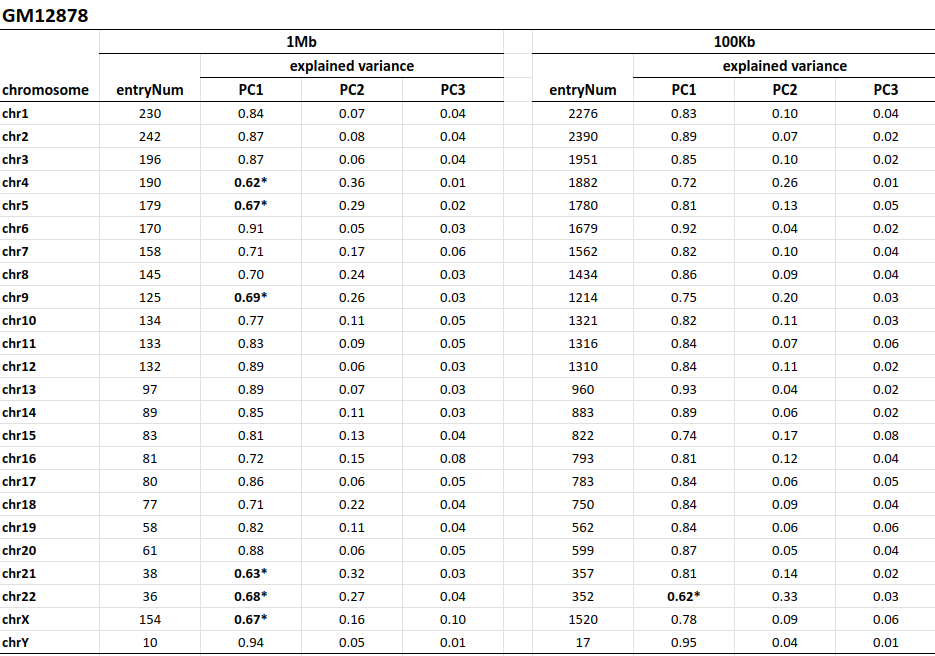
### **Explained Variance Comparison Results**

We summarize the explained variance calculated for GM12878 and K562 at resolutions of 1 Mb and 100 Kb in Table 2. Note that the PC1 calculated by Scikit-learn were all confirmed to have a high cosine similarity of nearly 1 or -1 between the Juicer’s PC1; since the signs of the eigenvector are mathematically arbitrary by PCA, sometimes the sign-pattern will be the exact opposite [25].

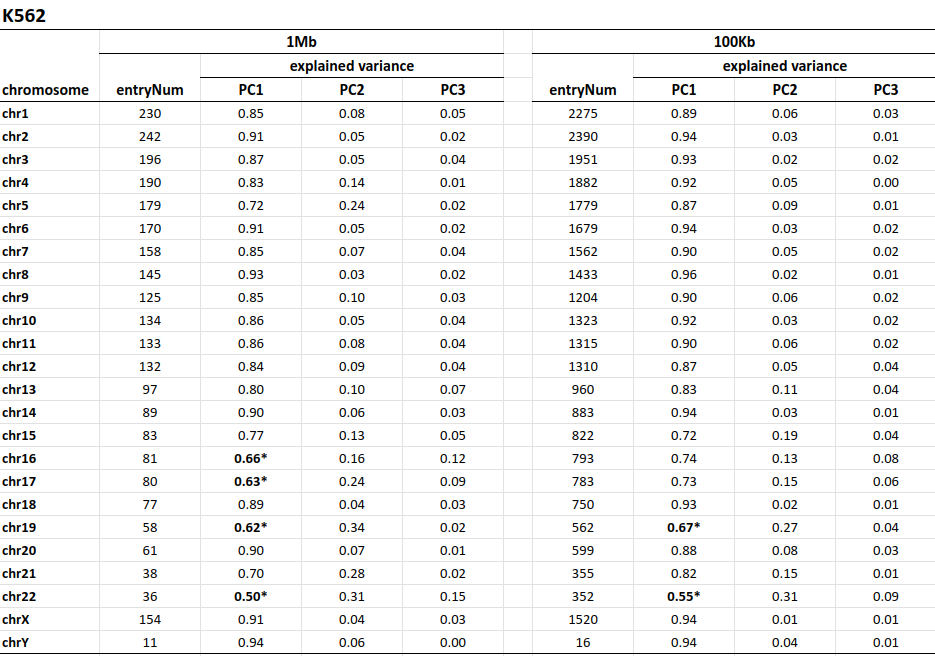
First, we notice at a coarse resolution such as 1 Mb, the variances explained by the PC1 for some chromosomes are quite low. We emphasize the blocks in Table 2 with the explained variance in PC1 lower than 0.7 by asterisk, and find that chromosomes 4 and 5, which were mentioned by Erez Lieberman-Aiden et al. [1] that the PC2 should be used for compartment identification but not the PC1, are included in the low PC1 explained variance blocks; besides, we also notice the number of emphasized blocks decreases as the resolution getting finer at 100 Kb in Table 2 (A) and Table 2 (B), accompanied with a general PC1 explained variance increases for all chromosomes. Except for the chromosome arms effect mentioned in the previous study [23][31], we assume that the explained variance also influences compartment identification, which should be considered, especially when the matrix resolution is quite coarse.

**Table 2.** Explained variance summarization for the top three principal components for GM12878 (A) and K562 (B) at resolutions of 1 Mb and 100 Kb. The blocks with PC1 explained variance lower than 0.7 are emphasized by asterisk.

**(A)**



**(B)**



## **Future Works**

We decided to compare the correlation between our approximation PC1 with the histone modifications [32], which are characteristic of the transcriptional activity for compartmental domains. Besides, we will further tweak our algorithm, test it on a higher resolution matrix, and determine if we can use these signals for a finer subcompartment identification [10].

## 

## **References**

1. Erez Lieberman-Aiden\*, Nynke L. van Berkum\*, et al. “Comprehensive mapping of long-range interactions reveals folding principles of the human genome.” *Science* 326 (2009). GScholar Citations: 1626. Cover Article.
2. Dekker J, Rippe K, Dekker M, Kleckner N. Capturing chromosome conformation. *Science*. 2002 Feb 15;295(5558):1306-11. doi: 10.1126/science.1067799. PMID: 11847345.
3. Zhao Z, Tavoosidana G, Sjölinder M, Göndör A, Mariano P, Wang S, Kanduri C, Lezcano M, Sandhu KS, Singh U, Pant V, Tiwari V, Kurukuti S, Ohlsson R. Circular chromosome conformation capture (4C) uncovers extensive networks of epigenetically regulated intra- and interchromosomal interactions. *Nat Genet.* 2006 Nov;38(11):1341-7. doi: 10.1038/ng1891. *Epub* 2006 Oct 8. *PMID*: 17033624.
4. Dostie J, Richmond TA, Arnaout RA, Selzer RR, Lee WL, Honan TA, Rubio ED, Krumm A, Lamb J, Nusbaum C, Green RD, Dekker J. Chromosome Conformation Capture Carbon Copy (5C): a massively parallel solution for mapping interactions between genomic elements. *Genome Res*. 2006 Oct;16(10):1299-309. doi: 10.1101/gr.5571506. *Epub* 2006 Sep 5. *PMID*: 16954542; *PMCID*: PMC1581439.
5. Kalluchi A, Harris HL, Reznicek TE, Rowley MJ. Considerations and caveats for analyzing chromatin compartments. *Front Mol Biosci*. 2023 Apr 5;10:1168562. doi: 10.3389/fmolb.2023.1168562. *PMID*: 37091873; *PMCID*: PMC10113542.
6. Yaffe, E., and Tanay, A. (2011). Probabilistic modeling of Hi-C contact maps eliminates systematic biases to characterize global chromosomal architecture. *Nat. Genet*. 43 (11), 1059–1065. doi:10.1038/ng.947
7. Servant, N., Varoquaux, N., Lajoie, B. R., Viara, E., Chen, C. J., Vert, J. P., et al. (2015). HiC-pro: An optimized and flexible pipeline for Hi-C data processing. *Genome Biol*. 16, 259. doi:10.1186/s13059-015-0831-x
8. Imakaev, M., Fudenberg, G., McCord, R. P., Naumova, N., Goloborodko, A., Lajoie, B.

R., et al. (2012). Iterative correction of Hi-C data reveals hallmarks of chromosome

organization. *Nat. Methods* 9 (10), 999–1003. doi:10.1038/nmeth.2148

1. Knight, P. A., and Daniel, R. (2013). A fast algorithm for matrix balancing. *IMA J. Numer. Analysis* 33 (3), 1029–1047. doi:10.1093/imanum/drs019
2. Rao SS, Huntley MH, Durand NC, Stamenova EK, Bochkov ID, Robinson JT, Sanborn AL, Machol I, Omer AD, Lander ES, Aiden EL. A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*. 2014 Dec 18;159(7):1665-80. doi: 10.1016/j.cell.2014.11.021. Epub 2014 Dec 11. Erratum in: *Cell*. 2015 Jul 30;162(3):687-8. *PMID*: 25497547; *PMCID*: PMC5635824.
3. Gavrilov, A., Eivazova, E., Pirozhkova, I., Lipinski, M., Razin, S., Vassetzky, Y. (2009). Chromosome Conformation Capture (from 3C to 5C) and Its ChIP-Based Modification. In: *Collas, P. (eds) Chromatin Immunoprecipitation Assays. Methods in Molecular Biology*, vol 567. *Humana Press*, *Totowa*, *NJ*. https://doi.org/10.1007/978-1-60327-414-2\_12
4. Dixon, J., Selvaraj, S., Yue, F. et al. Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature* 485, 376–380 (2012). https://doi.org/10.1038/nature11082
5. Nora, E., Lajoie, B., Schulz, E. et al. Spatial partitioning of the regulatory landscape of the X-inactivation centre. *Nature* 485, 381–385 (2012). https://doi.org/10.1038/nature11049
6. Harris, H.L., Gu, H., Olshansky, M. et al. Chromatin alternates between A and B compartments at kilobase scale for subgenic organization. *Nat Commun* 14, 3303 (2023). https://doi.org/10.1038/s41467-023-38429-1
7. Zhao, Y., Zheng, D. & Cvekl, A. Profiling of chromatin accessibility and identification of general cis-regulatory mechanisms that control two ocular lens differentiation pathways. *Epigenetics & Chromatin* 12, 27 (2019). https://doi.org/10.1186/s13072-019-0272-y
8. Kruse, K., Hug, C.B. & Vaquerizas, J.M. FAN-C: a feature-rich framework for the analysis and visualization of chromosome conformation capture data. *Genome Biol* 21, 303 (2020). https://doi.org/10.1186/s13059-020-02215-9
9. Heinz S, Benner C, Spann N, Bertolino E et al. Simple Combinations of Lineage-Determining Transcription Factors Prime cis-Regulatory Elements Required for Macrophage and B Cell Identities. *Mol Cell* 2010 May 28;38(4):576-589. *PMID*: 20513432
10. Abdennur, N., and Mirny, L.A. (2020). Cooler: scalable storage for Hi-C data and other genomically labeled arrays. *Bioinformatics*. doi: 10.1093/bioinformatics/btz540.
11. Neva C. Durand, Muhammad S. Shamim, Ido Machol, Suhas S. P. Rao, Miriam H. Huntley, Eric S. Lander, and Erez Lieberman Aiden. "Juicer provides a one-click system for analyzing loop-resolution Hi-C experiments." *Cell Systems* 3(1), 2016.
12. Rao SS, Huntley MH, Durand NC, Stamenova EK et al. A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell* 2014 Dec 18;159(7):1665-80. *PMID*: 25497547
13. Sanborn AL, Rao SS, Huang SC, Durand NC et al. Chromatin extrusion explains key features of loop and domain formation in wild-type and engineered genomes. *Proc Natl Acad Sci U S A* 2015 Nov 24;112(47):E6456-65. PMID: 26499245
14. Xiaobin Zheng, Yixian Zheng, CscoreTool: fast Hi-C compartment analysis at high resolution, *Bioinformatics*, *Volume* 34, Issue 9, May 2018, Pages 1568–1570, https://doi.org/10.1093/bioinformatics/btx802
15. Wen Z, Zhang W, Zhong Q, Xu J, Hou C, Qin ZS and Li L (2022) Extensive Chromatin Structure-Function Associations Revealed by Accurate 3D Compartmentalization Characterization. *Front. Cell Dev. Biol*. 10:845118. doi: 10.3389/fcell.2022.845118
16. Jonathon Shlens. A Tutorial on Principal Component Analysis. 2014. *arXiv*:1404.1100
17. Jolliffe Ian T. and Cadima Jorge 2016Principal component analysis: a review and recent developmentsPhil. *Trans. R. Soc. A*.3742015020220150202

http://doi.org/10.1098/rsta.2015.0202

1. Hung-yi Lee, NTU. (2020, Nov 18). ML Lecture 13: Unsupervised Learning - Linear Methods. *YouTube*. https://youtu.be/iwh5o\_M4BNU?si=LXabu4jMu8YDFHgg
2. Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. *Nature* 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2. (Publisher link).
3. Waskom, M. L., (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60), 3021, https://doi.org/10.21105/joss.03021.
4. Pedregosa et al., JMLR 12, pp. 2825-2830, 2011. Scikit-learn: Machine Learning in Python, Pedregosa et al., *JMLR* 12, pp. 2825-2830, 2011. *arXiv*:1201.0490
5. McKinney, W., & others. (2010). Data structures for statistical computing in python. In Proceedings of the 9th *Python in Science Conference* (Vol. 445, pp. 51–56).
6. Muller, H., Gil, J., and Drinnenberg, I. A. (2019). The Impact of Centromeres on Spatial Genome Architecture. *Trends Genet*. 35, 565–578. doi:10.1016/j.tig.2019.05.003
7. Nichols MH, Corces VG. Principles of 3D compartmentalization of the human genome. *Cell Rep*. 2021 Jun 29;35(13):109330. doi: 10.1016/j.celrep.2021.109330. *PMID*: 34192544; *PMCID*: PMC8265014.