Comparison of TAD Callers for Structural Analysis of Chromatin Structure from Interaction Hi-C Matrix

02-512, Fall 2021

Zhiyuan (Leon) Xu, Xiao (Katrina) Liu, Yixiao (Amy) Zhu, Jinru (Linda) Zhou, Shreya Varra

December 13, 2021

1 Abstract

Topologically associating domains (TAD) are self-interacting genomic domains. They are usually captured using genome-wide techniques such as Hi-C. Recently, TADs have gained increasing research interest, as new insights into the human genome structures have driven many discoveries of the underlying mechanisms of genetic expressions, functions, and diseases. In the 2018 paper Quantifying the similarity of topological domains across normal and cancer human cell types, Sauerwald and Kingsford analyzed the similarities of the superstructures between cancer and normal cells using the Armatus TAD caller, which as they argued, was one of the best tools available. We compared a range of TAD callers with a set of various metrics to see whether the same results can be replicated for HiCseg, Armatus, and the most recent spectralTAD.

2 Introduction

Conserved across multiple species and cell-types throughout evolution, TADs offer an insight into functional elements in the genome as there are evidence suggesting that all TADs represent functional domains. By conducting analysis on the TAD domains, researchers would be able to advance understanding of the geometry of chromatin structure, its relation to the regulation of gene expression, nuclear organization, cancer translocations, and copy number alterations in cancer. Often, disruptions in TAD boundaries is a signal for changes in gene expression as different regulatory elements comes into contact. Studies have shown TAD disruptions are often found in cancer cells and contributes to oncogenesis.

The three papers all presented their algorithm as an approach to identify the TAD boundaries. We intend to compare the accuracy as well as the efficiency for each algorithm to conclude which one is the most suitable tool for research. For example, to investigate how the boundaries are affected if a gene was deleted, we would need the algorithm to accurately identify the boundaries. If we were to compare the boundaries between numerous cancer and normal patients, then we would need a more efficient approach. [9]

3 Methods

3.1 Armatus[3]

The intuition behind Armatus algorithm is to separate the problem of finding TAD domains based on Hi-C matrices with different resolutions into two sub-problems, where the first step is to distinguish a set of boundaries based on data matrix of one given resolution, and the second step is to combine these result boundaries into one set of consensus boundaries. The two sub-problems are being solved by separate dynamic programming approaches. A brief introduction of the objectives and recursions for both steps is provided below.

1. (Resolution-specific domains) Given a $n \times n$ weighted adjacency matrix A and a resolution parameter $\gamma \geq 0$,The goal of this step is to generate the set of non-overlapping domain boundaries which optimize the following objective:

$$\max \sum_{[a_i,b_i]} \in D_\gamma q(a_i, b_i, \gamma)$$

where the function q is a function that captures the the quality of domains (level of interaction) defined as

$$q(k, l, \gamma) = s(k, l, \gamma) - \mu_s(l - k).$$

In this case, the function s represents the scaled density of the sub-graph between interval k and l, which defined formally as

$$s(k, l, \gamma) = \frac{\sum_{g=k}^{l} \sum_{h=g+1}^{l} A[g][h]}{(l - k)^\gamma}$$

and μ_s serves as an normalizing parameter function to shift the mean to zero.

Based on the definitions aboce, the derived recursion for dynamic programming is

$$OPT_1(l) = \max_{k<l} \{OPT_1(k - 1) + \max\{q(k, l, \gamma), 0\}\}.$$

2. (Consensus domains across resolutions) Given A and a set of resolutions $\Gamma = \{\gamma_1, \gamma_2, ...\}$ where we try to identify a set of the non-overlapping domains D_c that best represents all domains for each resolution. A persistent function is derived to formalize the objective for this problem:

$$p(a_i, b_i, \Gamma) = \sum_{\gamma \in \Gamma} \delta_i$$

where δ_i is a indicator variable that represents where interval $[a_i, b_i] \in D_\gamma$. In other words,

$$\delta_i = \begin{cases} 1 & \text{if } [a_i, b_i] \in D_\gamma \\ 0 & \text{otherwise.} \end{cases}$$

As a result, the objective is represented as

$$\max \sum_{[a_i,b_i]\in D_c} p(a_i, b_i, \Gamma)$$

The recursion used for dynamic programming is then expressed as

$$OPT_2(j) = \max\{OPT_2(j-1), OPT_2(c(j)) + p(a_i, b_j, \Gamma)\}$$

where $c(j)$ is the closest domain before j that does not overlap with j.

3.2 SpectralTAD[2]

The SpectralTAD algorithm makes use the premise that TADs are only found along the diagonal of the contact matrix. By using the sliding window, the algorithm limits the scope of data being processed and reduces the runtime significantly. The overall scheme of the procedure is done be sliding the window with a fixed size across the diagonal and within each window, the algorithm performs spectral clustering to find unique TADs by analyzing the spectrum. To find the TADs, SpectralTAD decomposes the contact matrix into eigenvectors, which are then projected to a unit circle to form clusters. Finally, we calculate the silhouette score to determine the number of clusters in each window to minimize distance within each cluster. The general procedure is shown as follows in pseudocode.

procedure SPECTRALTAD(w, C)
  for window W with size $w \times w$ in C **do**
    $D = diag(1^T C)$
    Get Laplacian matrix $\bar{L} = D^{-\frac{1}{2}} C D^{-\frac{1}{2}}$
    Decompose the matrix by solving $\bar{L}\bar{V} = \lambda\bar{V}$
    Normalize eigenvector matrix $\hat{V} = \frac{\bar{V}}{||\bar{V}||}$
    Project vectors onto unit circle

Derive silhuoette score $s = \frac{b-a}{\max(a,b)}$ where a is the mean distance between each cluster entry and the nearest cluster, and b is the mean distance between points in the cluster

Minimize $\bar{s_m} = \frac{\sum_{i=1}^{m} s_i}{m}$ to find m

Add selected TADs for the window to results

Return results

3.3 HiCSeg[5]

The 2-dimensional segmentation algorithm essentially collapses the data into a 1-dimensional segmentation problem. Hi-C data is symmetric, so this algorithm only considers the upper triangular part of each of the bounded regions in the matrix. The intensity of the interactions between positions i and j is denoted by Y_{ij}, which is drawn from a probability function with parameter μ_{ij}. In our matrix, each bounded region can be denoted as

$$D_k = \{(i,j) \mid t_{k-1} \leq i \leq j \leq t_k - 1\}$$

with all of the intensities within that region having a distribution of $\mathcal{N}(\mu_{ij})$. The areas outside of these regions are denoted by

$$E_0 = \{(i,j) \mid 1 \leq i \leq j \leq n \cap\}$$

as in the areas outside of the bounded regions. Each bounded region has a μ_k, and we end up with a general Hi-C data matrix:

The aim of this algorithm is to estimate the boundaries t_k for $`k \in [K]$ where K is the known number of blocks, which we accomplish using a maximum likelihood approach. Then, our log-likelihood becomes

$$\ell(Y) = \sum_{1 \leq i \leq j \leq n} \log(Y_{ij})$$

which reduces to

$$\ell(Y) = \sum_{k=1}^{K} \left(\sum_{i,j \in D_k} \ell_k(Y_{i,j}) + \sum_{i,j \in R_k} \ell_0(Y_{i,j} \right)$$

where R_k corresponds to the area in E_0 above the corresponding D_k. Now applying this to our dynamic programming approach yields

$$C(t_{k-1}, t_k - 1) = \sum_{i,j \in D_k} \ell_k(Y_{i,j}) + \sum_{i,j \in R_k} \ell_0(Y_{i,j}$$

for the cost function, which means we have to maximize with respect to t_k for $k \in [n+1]$

$$\sum_{k=1}^{K} C(t_{k-1}, t_k - 1)$$

Then, our objective function is defined as

$$I_L(\tau) = \max_{1=t_0<t_1<\ldots<t_L=\tau+1} \sum_{k=1}^{L} C(t_{k-1}, t_k - 1)$$

where $1 \leq L \leq K$ and $1 \leq \tau \leq n$. The value of this objective function represents the optimal segmentation of the submatrix made of the first τ rows and columns of Y into L blocks. We know that $I_1(\tau) = C(1, \tau)$ which means we can define the recursion for this objective function as

$$I_L(\tau) = \max_{1<t_{L-1}<\tau+1} I_{L-1}(t_{L-1} - 1) + C(t_{L-1}, \tau)$$

Because we create a subproblem for each of the 1, 2, ..., K number of blocks and for each of the n^2 possible permutations of boundaries, the runtime of this algorithm is in $O(Kn^2)$

3.4 Evalutation Metrics

To evaluate the results generated from each method, two main algorithms here were used to compute the accuracy using the real boundaries that we hand-annotated: Variation of Information[6] and Jaccard Index[4]. Each of them aims to calculate the similarity/dissimilarity between two sets of values, but the two algorithms use different approaches.

1. Variation of Information(VI): Variation of Information calculates the distance between the two clusters of data, hence a maximal value will occur when the two sets of boundaries are

disjoint, while a minimum value of 0 will occur when the two sets of boundaries completely overlap with each other. The formula of VI is shown below:

$$VI(C, C') = H(C) + H(C') - I(C, C')$$

Here C and C' each stands for one data cluster, and $I(C, C')$ is the intersection of the two clusters. The result returned by the formula indicates the distance between the two clusters, hence lower result means a higher overlap between the boundaries and therefore a higher accuracy.

2. Jaccard Index(JI): Jaccard Index, in contrast, calculate the similarity between the boundaries, by finding the intersection and the union of the two sets of values. When using JI, a maximal value of 1 occur when the two sets completely overlap with each other, and a minimal value of 0 would occur when the two sets are disjoint. The formula used by Jaccard Index is shown below:

$$JI(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Here A,B are the two sets of boundaries, and the way of obtaining the similarity is just dividing the intersection of the two sets by the union of them.

4 Data Sources

The data sources we used are HiC data for a breast cancer cell line (MCF7) and a non-tumorigenic epithelial cell line (MCF10a) from 2015 study by Rasim [1]. Both datasets contain ICE normalized HiC data in resolution $40,000$ and $250,000$ and their respective ground truths. We used the data with $40kb$ resolution for better defined TAD boundaries.

5 Results

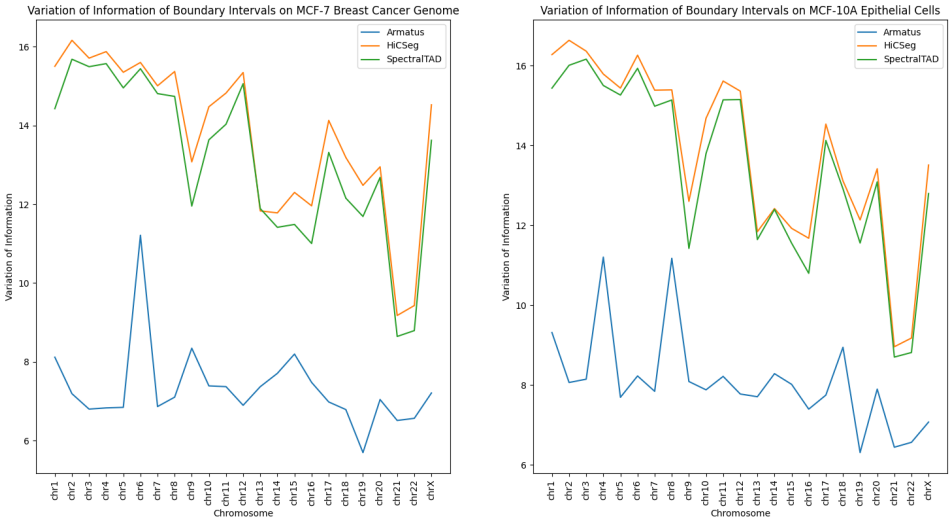

Figure 1: Variation of information between boundaries computed by Armatus[3], HiCSeg[5], and SpectralTAD[2] for MCF-7 breast cancer genome and MCF-10A epithelial cells[1].

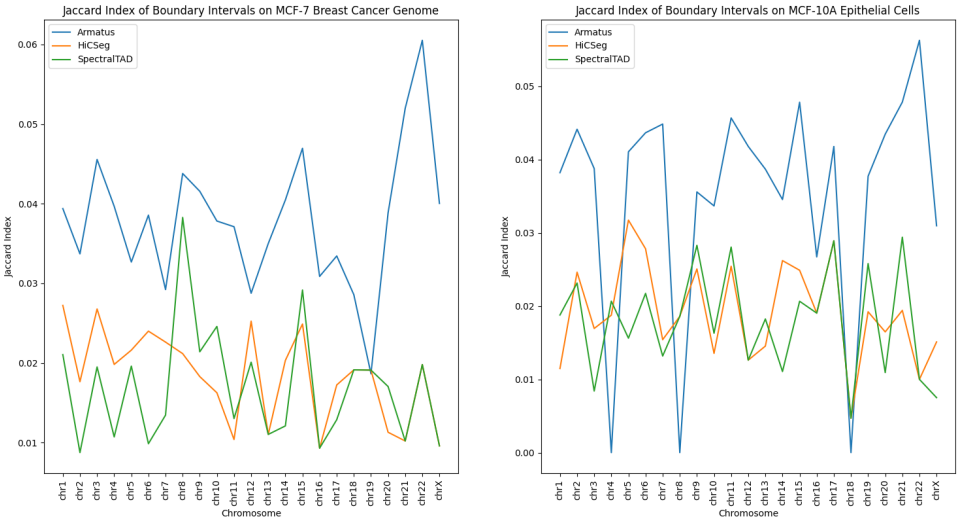

Figure 2: Jaccard index between sets of boundary endpoints computed by Armatus[3], HiCSeg[5], and SpectralTAD[2] for MCF-7 breast cancer genome and MCF-10A epithelial cells[1].

5.1 Accuracy

Both Variation of Information and Jaccard Index have shown similar results. Armatus has the highest accuracy among all three methods, followed by Spectral TAD and HiCSeg. As shown in the figures above, although Spectral TAD has a slightly higher accuracy, the overall trend is very similar compared to HiCSeg. However, the results from HiCSeg should be taken with a grain of salt, as the algorithm only returns endpoints of TADs identified, which are not necessarily well defined intervals. In the plot for Variation of Information (Figure 1), Armatus (shown in blue) is significantly lower than the other two, suggesting a lower difference between predicted boundaries and the true boundaries, leading to a higher accuracy. Similarly, in the plot for Jaccard Index (Figure 2), the line representing Armatus is generally higher than the other two, also indicating that there's a higher overlap between Armatus's prediction and the true boundaries. Some of the chromosomes (4, 8, and especially 18) can be seen as outliers. Looking at the actual outputs, Armatus skipped a large region, and the boundaries that were identified did not overlap with the true boundaries at all. We hypothesize that the algorithm may have returned a small portion of the actual boundaries identified.

5.2 Efficiency

Our timed executions showed results that are consistent with the calculated Big-O. With Armatus taking 349.6 seconds per cell line, SpectralTAD taking 58.09 seconds, and HiCSeg taking 6110.99 seconds, Armatus and SpectralTAD are suitable candidates for HiC calling on-the-fly, whereas HiCSeg's long runtime makes it impractical in cases of large chromosomal data. SpectralTAD's sliding window method drastically reduces the amount of data need to process, taking down the bottleneck, the matrix size from $O(N^2)$ to $O(N)$. By making use of similar techniques, we can potentially improve the runtime for other TAD callers.

6 Conclusion

From the results, we can see that the Armatus algorithm has the highest accuracy when comparing the variation of information on the boundary intervals. SpectralTAD and HiCSeg show same trends in their accuracies with SpectralTAD having a slightly higher overall accuracy. Also, when comparing the efficiency of the algorithms, we found that the HiCSeg algorithm has the highest runtime by far, whereas the SpectralTAD and Armatus algorithms required much less time to run. Overall, our results matched the results of the 2018 paper by Sauerwald and Kingsford in that Armatus was the best performing algorithm when looking at accuracy and efficiency.

7 Discussion

In this project, our group explores three different algorithms used for finding topological domains based on chromatin Hi-C matrix data. Hi-C data analysis provides an novel way to approach chromosome contacts, and the discoveries built upon it allows us to further examine the genome architecture, which facilitates the understanding of normal genome regulation[8]. Admittedly, our study do suffer slightly by limitations of our understanding for HiC data, level of adjustability for each algorithm, and available datasets that do not require excessive formatting. Furthermore, our accuracy metrics (JI and VI) might not fully characterize the differences in clusters for the results we examined, with some poor performances not necessarily detected. However, regardless of the algorithm chosen, there are technological barriers that limits the accuracy of the resulting boundaries, such as the limit of resolution of Hi-C data and microscopy techniques used in Hi-C data collection. Aside from this, the ultimate purpose of analyzing topological domains is to understand its implication and functionality, where the identification of boundaries is the first step of the analysis. The biological meaning represented by the organization of chromatin

layers remains obscure, and a solution to these challenges will advance our understanding of the chromatin structure and interactions[7]. Our project hopes to provide a guide line to the begin of such analysis, which will allow future researchers to target specifically for different data sources and individual experiment design.

8 Supplemental Information

Link to source code: https://github.com/katrina-liu/02-512-final-project

References

[1] A Rasim Barutcu, Bryan R Lajoie, Rachel P McCord, Coralee E Tye, Deli Hong, Terri L Messier, Gillian Browne, Andre J van Wijnen, Jane B Lian, Janet L Stein, Job Dekker, Anthony N Imbalzano, and Gary S Stein. Chromatin interaction analysis reveals changes in small chromosome and telomere clustering between epithelial and breast cancer cells. *Genome Biology*, 16(214), 2015.

[2] Kellen G. Cresswell, John C. Stansfield, and Mikhail G. Dozmorov. Spectraltad: an r package for defining a hierarchy of topologically associated domains using spectral clustering. *BMC Bioinformatics*, 21(319), 2020.

[3] Darya Filippova, Rob Patro, Geet Duggal, and Carl Kingsford. Identification of alternative topological domains in chromatin. *Algorithms for Molecular Biology*, 9(14).

[4] Paul Jaccard. The distribution of the flora in the alpine zone.1. *New Phytologist*, 11(2):37–50, 1912.

[5] Celine Lévy-Leduc, M. Delattre, T. Mary-Huard, and S. Robin. Two-dimensional segmentation for analyzing hi-c data. *Bioinformatics*, 30(17), September 1999.

[6] Marina Meilă. Comparing clusterings by the variation of information. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *Learning Theory and Kernel Machines*, pages 173–187, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[7] Koustav Pal, Mattia Forcato, and Francesco Ferrari. Hi-c analysis: from data generation to integration. *Biophysical Reviews*, 11(67-78).

[8] Amos Tanay and Giacomo Cavalli. Chromosomal domains: epigenetic contexts and functional implications of genomic compartmentalization. *Current Opinion in Genetics and Development*, 23(2):197–203, 2013. Genome architecture and expression.

[9] Anne-Laure Valton and Job Dekker. Tad disruption as oncogenic driver. *Current Opinion in Genetics and Development*, vol. 36, 2016: 34-40.