MEET-EU 2021
Final Project Report

BANANA-TAD: OBTAINING A CONSENSUS FROM MULTIPLE TAD CALLING ALGORITHMS

TEAM SA-1

AUTHORS:

Abakarova Marina
Crouzet Simon
Le Goffic Liam
Trang Alexis
Zhong Yann

28th of February 2022

Abstract

Topologically Associating Domains (TADs) are a key biological structure in elucidating the mechanisms of chromosomal interactions, notably with regards to gene expression. However, the lack of a clear consensus on how a TAD is actually defined presents itself as an obstacle in furthering research in this field. Here, we present BananaTAD, a method that incorporates the calls from 4 existing TAD detection algorithms in order to score them and give a list of consensus TADs at a given resolution.

Acknowledgements

We would like to thank Elodie Laine, Léopold Carron and Juliana Bernades for their guidance throughout this project.

Contents

1 Introduction

1.1 TAD: Topologically Associating Domains

DNA, also known as Deoxyribonucleic Acid, can be considered the basic building block of all life on this planet, and can reach immense lengths and complexity. The size of the human genome is roughly 3.2 billion base pairs of nucleotides and comfortably fits inside a simple nucleus. This is in part due to the presence of chromatin: a DNA-protein complex supported and regulated by various factors and actors that allows for extreme compaction.

In recent years, more and more research into chromosomal organisation has revealed the presence of regions known as Topologically Associating Domains (TADs), which are densely packed and have a high degree of interaction of sequences within themselves. This is because parts of sequences which are normally far apart by virtue of their distances on a sequence can find themselves interacting with each other by forming loops and grouping with each other in local proximity, such as illustrated on figure 1: such regions can be considered TADs.

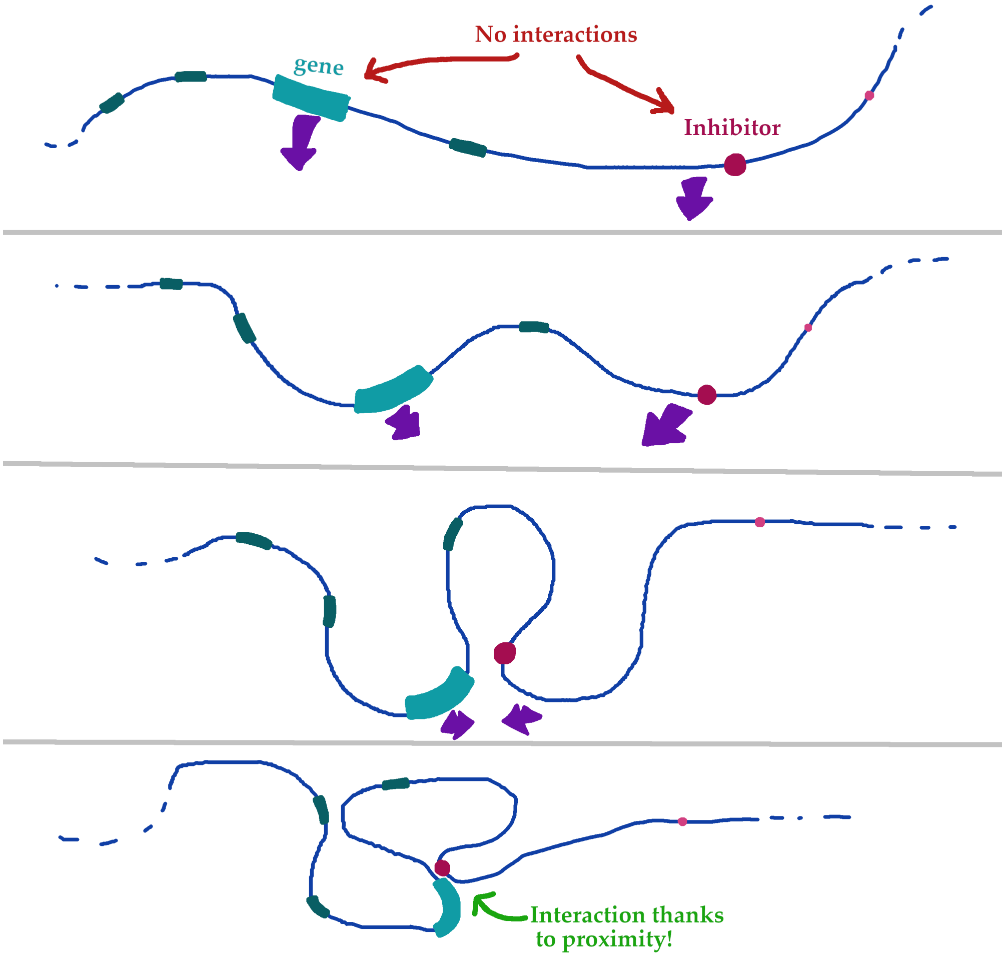


Figure 1: Simplified example of proximity chromosomal interaction due to loop formation. An inhibitor for a specific gene may only find activation once a loop has been formed, causing normally distant interactions to become proximal.

TADs are important because their presence is often associated to the regulation of gene expression and their disruption is linked to oncogenic onset [1], to cite on example. Additionally, TADs have also been shown to be conserved across different species [2], meaning that they are not limited to homo sapiens in this regard. The accurate detection of TADs, however, is currently still the subject of much research and debate. They are typically "called" (in other words, identified) through the analysis of raw Hi-C (also known as chromosome conformation capture) sequencing data by various algorithms and pipelines such as Arrowhead [3] or Topdom [4], historically considered to be strong contenders for a gold standard method. Yet, there is still much debate on the strict definition of a TAD, notably with regards to its size. For example, some pipelines go about identifying TADs as distinct non overlapping entities while others seek to identify TADs through different levels and sub-levels, giving rise to the notion of sub-TADs.

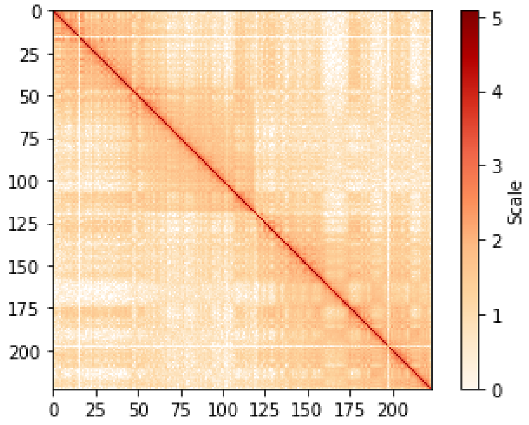


Figure 2: Contact matrix of GM12878 at 100 kb resolution (x and y axis at 20 Million base pairs) on log10 scale

The principal input to such pipelines is raw coordinate data which can be visualised as a contact matrix, as shown in figure 2 as well as figure 3, in a much more simplified manner. A contact matrix is a matrix that shows the entirety of the contacts within a chromosome as log scales of interactions. Both the x and y axis represent the same coordinates displayed according to the resolution chosen for visualising this matrix. For the matrix shown above, the resolution is 100kb, so 0 to 25 on the figure goes from 0 to 2.5 million base pairs in actuality. The darker the colors shown, the stronger the interactions at a specific region, and visually speaking TADs can be roughly visualised as squares along the diagonal line (or simply triangles above or below the diagonal line, since the matrix is symmetrical).

With this intuition, it is relatively simple to visualise the presence of a TAD by "looking for triangles" (or squares) along the diagonal, but in reality, many problems associate themselves to this question.

For example, if multiple such triangles overlap, do we consider merging them into a bigger TAD, or nesting them into each other as TADs at differing levels, knowing that TAD size categorization is also poorly defined? These questions were important to answer, but also to examine from the point of view of the consequent number of existing algorithms in literature.

1.2 Aims and scope of the project

Consequently, in this project, we sought to examine the TAD calls from multiple established methods on contact matrices in order to get an overall list of resulting consensus TADs after putting them through a scoring system of our own. To do so, we separately re-implemented a number of these methods, notably Topdom, TADTree, OnTAD, and TADBit, and integrated them within a centralised code making use of abstract classes. Then, TAD calls were given a specific score based on metrics and integrated into our final output list of TADs. In the rest of this report, we outline our methods, results and discussion related to our BananaTAD implementation.

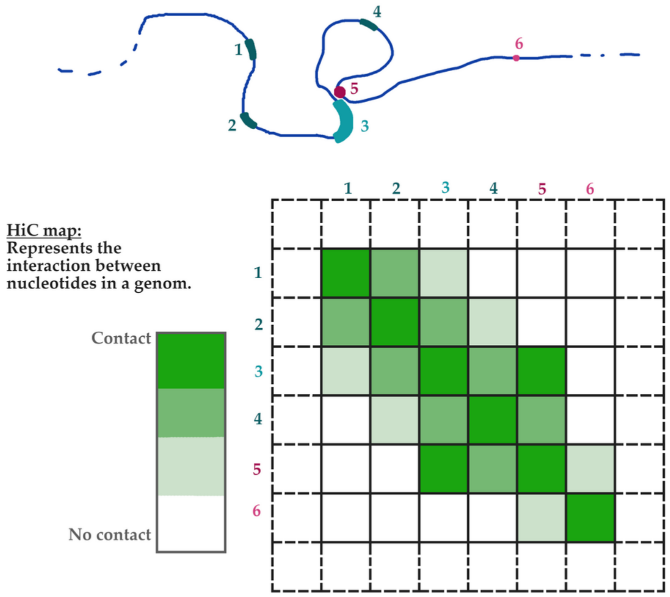


Figure 3: Simplified representation of a contact matrix. As this was done for illustrative purposes, the color scales may not be fully accurate, but serve to represent that coordinates 3 and 5, for example, have heightened interactions due to local proximity.

2 Methods

2.1 Pipelines and algorithms used in consensus

Rola et al., in their 2017 paper examining 7 different TAD prediction tools (Armatus, Arrowhead, DomainCaller, HiCSeg, TADBit, TADTree and Topdom) [5], state that "the lack of a consensus on what exactly a TAD is, combined with the variability in TAD prediction tools, lead to inconsistencies across studies and are an impediment to research in the field". There was consequently real motivation in the creation of a system that could consider and score the TADs predicted by multiple methods. We then considered that the next logical step was to incorporate the results from said system in order to extract a list of TADs we call "consensus TADs".

In this section, we will rapidly outline the principles behind each of the methods that we incorporated into our pipeline.

Topdom [4] was the first considered methods as it was suggested early on as one of the most used and referenced algorithms. To put it briefly, Topdom is a deterministic method that makes use of a sliding diamond shaped window that moves along the diagonal axis in increments of predetermined bin sizes to calculate local minima from average contact frequency within the window. The local minima found along the diagonal axis are a measure of indication of how likely that position is a TAD boundary (that is, the leftmost or rightmost coordinate of a TAD).

Topdom was originally written in Python, so was thus chosen to be the very first pipeline in our reimplementation and subsequent integration into an abstract class, which will be detailed in a further section. We re-implemented Topdom for 100kb and 25kb TAD prediction.

TADTree [6] was the second pipeline considered in our work. TADTree came up in literature review as a python-based algorithm that was able to detect both TADs and sub-TADs as a set of nested hierarchies. TADTree makes use of dynamic programming in order to optimize an objective function and create optimal TAD-forests within the contact matrix. Due to the nature of the algorithm, TADTree's time complexity was much greater than any of the other programs at $O(S^5)$ where S is the maximum TAD size defined in the input, and could thus only be realistically re-implemented on 100kb data, as runtime on higher resolutions was much too consequent.

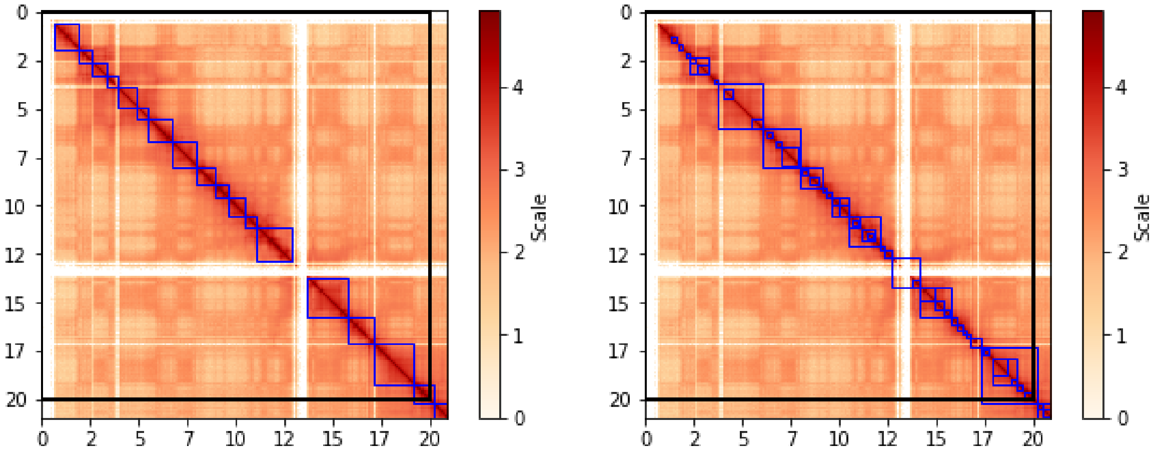


Figure 4: GM12878 from 0 to 2 million base pairs (100kb resolution)
Left: Topdom predictions. Right: TADTree predictions

OnTAD [7] was the third pipeline considered. Through literature review and personal research, we found that OnTAD was a promising method that derives parts of its principles from Topdom and makes use of a similar sliding diamond-shaped window. Additionally, despite its C++ implementation, due to its ease of use (either by virtue of Docker or a direct executable and a simple chmod command to grant access permission), we were not required to directly re-implement this pipeline as the previous two, rather it was integrated into the aforementioned abstract class.

OnTAD was run and tuned on 25kb data.

TADBit[8] was the fourth and final pipeline that we considered and re-implemented, and was the only one out of the four pipelines that featured a Bioconda package. Essentially, TADBit is based on a breakpoint detection algorithm that looks at slices of matrix data and computes log likelihood for said slice. A dynamic programming algorithm thus looks for the decomposition of the Hi-C contact matrix into optimal slices, with an additional border score assigned to suspected TAD borders to consider whether they are robust or not.

TADBit was run and tuned on 100 kb data.

Finally, **Arrowhead** [3] was used as the baseline gold standard for evaluating the TADs obtained from the 4 algorithms mentioned above. Arrowhead was implemented by Rao et al. and relies on detecting domain corners as pixel coordinates on the Hi-C contact matrix, and subsequently replacing said domains with an arrowhead-shaped motif (hence the name). As a result, we based our scoring and consensus system off of Arrowhead as they were provided to us for use as ground truth relative to our implementations of the other prediction algorithms.

The Arrowhead data was provided to us at a resolution of 5kb.

Training and validation was done on five cell lines provided by Rao et al. [3], namely GM12878, HMEC, HUVEC, IMR90, and NHEK. All cell lines included chromosomes 1 to 22 as well as chromosome X.

2.2 Scoring system

In order to properly integrate the 4 methods mentioned above into a relevant consensus, we developed a scoring system that includes using CCCTC-binding factor (CTCF) that are biological markers present to a certain degree on TAD boundaries. Additionally, we also used parts of the gold standard Arrowhead dataset to obtain two relevant metrics that were included in the scoring.

Both the CTCF as well as the gold standard derived metrics were split into a training and validation dataset.

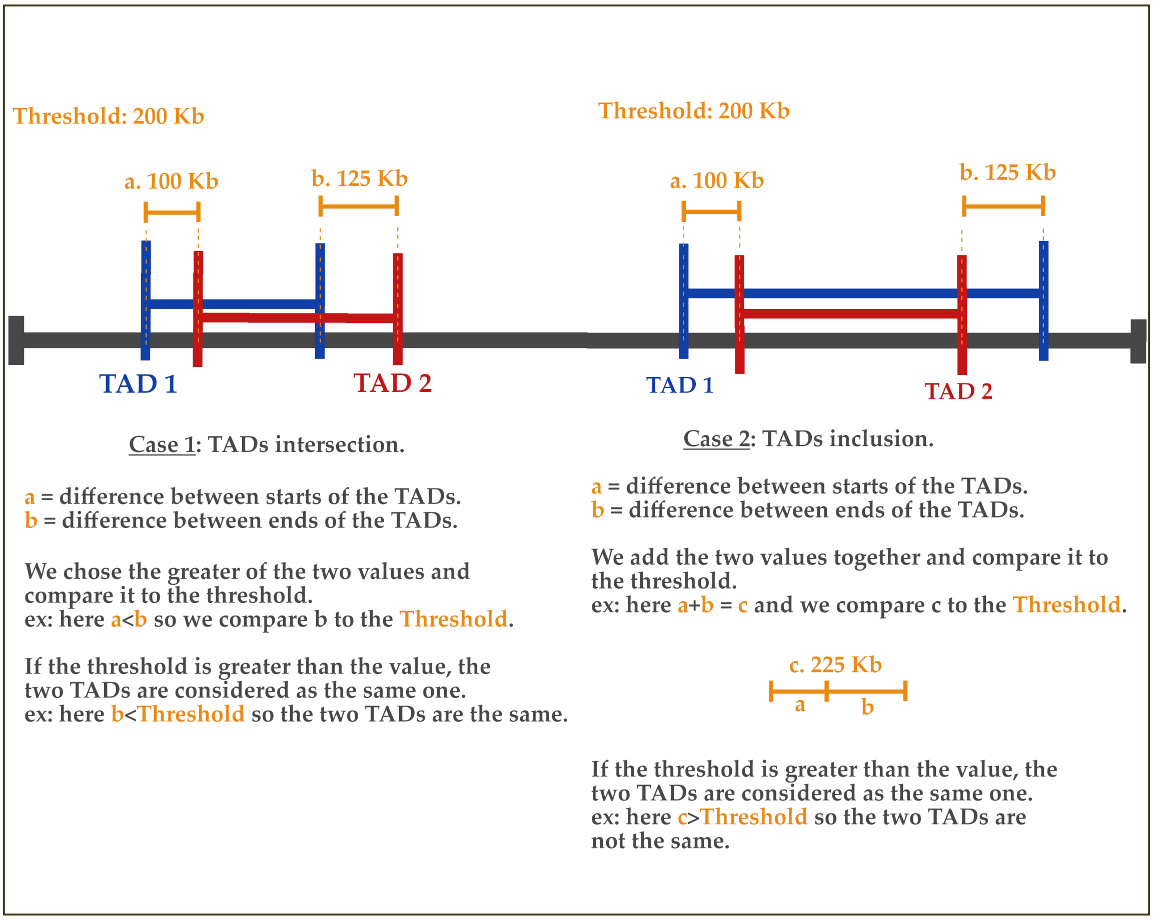
2.2.1 TAD prediction rates - M_{gt} and M_{pred}

The most significant part of our scoring was in calculating the percentage of called TADs found in the ground truth TAD list, as well as the percentage of ground truth TADs found in the list of TADs from the called method, which we respectively called the metrics M_{gt} and M_{pred}:

$$M_{gt} = \frac{TP}{TP + FN} \quad M_{pred} = \frac{TP}{TP + FP}$$

Where TP corresponds to True Positives, or TADs predicted by the method that were indeed found in the ground truth, FP corresponds to False Positives, or TADs predicted by the method that did not exist in the ground truth, and FN corresponds to False Negatives, or TADs that were not predicted by the method but were present in the ground truth. It is worth noting that M_{gt} and M_{pred} are roughly equivalent to recall and precision in typical machine learning problems, but here we also lack the True Negative parameter.

In order to consider if the TAD predicted by a pipeline such as Topdom is the same one predicted by the Arrowhead results that were provided to us as validation, for example, we decided to define a gap value that would allow for a certain margin of error. As shown in figure 10, there were two scenarios for consideration.



Threshold: 200 Kb
a. 100 Kb
b. 125 Kb
TAD 1
TAD 2
Case 1: TADs intersection.
a = difference between starts of the TADs.
b = difference between ends of the TADs.
We chose the greater of the two values and compare it to the threshold.
ex: here a<b so we compare b to the Threshold.
If the threshold is greater than the value, the two TADs are considered as the same one.
ex: here b<Threshold so the two TADs are the same.
Threshold: 200 Kb
a. 100 Kb
b. 125 Kb
TAD 1
TAD 2
Case 2: TADs inclusion.
a = difference between starts of the TADs.
b = difference between ends of the TADs.
We add the two values together and compare it to the threshold.
ex: here a+b = c and we compare c to the Threshold.
c. 225 Kb
a
b
If the threshold is greater than the value, the two TADs are considered as the same one.
ex: here c>Threshold so the two TADs are not the same.

Figure 5: Illustrative example for considering if a TAD from a tested method was the same as Arrowhead ground truth, based on cases where they intersect with or include each other

In both cases, we set a threshold value, such as 200 kilobases (this particular value was incidentally determined to yield the greatest number of matching TADs). Because we know that there will be almost no cases where the TADs called will be exactly equal in boundaries, especially since different methods work at different resolution, the decision is taken based on which of the two cases we encounter.

1. **The TADs intersect:** In this case, we choose the greater of the difference between the start of the TADs and the end of the TADs. If the chosen value is lower than the assigned threshold, then we consider that the TAD predicted by the method is the same as that as the one in the ground truth.

2. **One TAD includes another:** In this case, we instead add together the difference between the start of the TADs and the end of the TADs. If the sum is lower than the assigned threshold, then we consider that the TAD predicted by the method is the same as that as the one in the ground truth.

It is also worth noting that these two metrics were tuned to be the greatest possible for each single method by adjusting the methods' parameters. For example, one of the most important parameters for Topdom was the window size. At 100kb resolution, the best window size was found to be equal to 2 (multiplied by the resolution), while at 25kb it was found to be 3.

2.2.2 CTCF peak correspondence - C_m

CTCF and cohesin are two proteins known to engage in DNA loop formation: TAD boundaries are consequently known to be enriched in CTCF binding sites at high rates [2][9]. Rola et al. also found through their aforementioned TAD tool analysis that predicted boundaries for all tools exhibited CTCF site enrichment, with the strongest for Topdom and TADTree, two algorithms we use in our method. Thus, we immediately considered the use of CTCF markers in order to demarcate TAD boundaries and add a measure of confidence to our consensus predictions.

CTCF enriched positions were obtained from the ENCODE project for GM12878, HMEC, HUVEC, IMR90 and NHEK in the BED format [10]. These files all contained the precise base level resolution parts of chromosomes that were enriched in CTCF. Thus, when a TAD was predicted by one of the four methods and any of its two borders (leftmost or rightmost) were within a CTCF enriched part of the chromosome we take as input, we considered that there was a CTCF correspondence.

Then, for all the TADs predicted by a method, we calculated the rate of CTCF correspondence TADs relative to all the predicted TADs, and called it C_m, where the m subscript indicated the method used. For example, if 60 out of 120 TADs were CTCF enriched, then C_m for that method was simply equal to 0.5.

2.2.3 Adapted accuracy - A_m

For a given method, once the TAD prediction rates were calculated, we were then able to assign it a particular score based on the following simple formula, which we call adapted accuracy:

$$A_m = \frac{M_{gt} + M_{pred}}{2}$$

Where M_{gt} and M_{pred} are the TAD prediction rates, and A_m is the adapted accuracy (score) for this given method.

2.3 Consensus weight method

2.3.1 Weight attribution

Once the adapted accuracy as well as the CTCF peak correspondence were both calculated, we then attributed a given consensus weight to each of our methods, corresponding to the following formula:

$$W_m = \alpha A_m * \beta C_m$$

Where W_m is the consensus weight for that method, A_m is the adapted accuracy and C_m is the CTCF peak correspondence. α and β are additional tuning factors that we played with, but we decided to keep as equal to 1 as modifying them made little to no difference. Below, a table showing the consensus weights calculated for each of the four methods varying on resolution:

Method	TADTree	Topdom	OnTAD	TADBit
25kb	N/A	0.33656	0.84340	N/A
100kb	0.86019	0.32533	N/A	0.27248

2.3.2 Consensus TADs and threshold value tuning

After obtaining the consensus weights, the next step was to apply said weights to every chromosome that was part of the training set. For example, if we randomly took chromosome 15 of GM12878 at 100kb to be part of the training set, the consensus weight for the algorithms would be respectively 0.86019 for TADTree, 0.32533 for Topdom, and 0.27248 for TADBit(OnTAD was not applicable for 100kb).

Then, for all the methods combined, we considered every TAD that was called. A TAD is defined by the coordinates of its leftmost border and its rightmost border. If a TAD was called by Topdom at positions x and y, then the coordinates x and y would get a weight of 0.32533, while x+1, x-1, y+1, y-1 a score of $\frac{0.32533}{2}$, and so on and so forth as we go further away from the borders. This is repeated for every TAD called by every method at that resolution, with every new weight summing into the position. Thus the weighting at a specific position follows the formula:

$$s(i) = \sum_{m=1}^{4} W_m(i) \qquad W_m(i) = \frac{W_m}{2^{|x-i|}}$$

Where m goes from 1 to 4 (1 for TADTree, 2 for Topdom, 3 for OnTAD, 4 for TADBit), W_m is the consensus weight for each of those methods, i is the position we're considering and x the position of the nearest predicted boundary to i.

Once the summed weight for every position of a chromosome has been calculated, we then considered that a border was acceptable to be part of the consensus TAD output if the sum at that position exceed a threshold value, after a multiplicative factor was applied to the sum.

The default threshold value was 20 for 100kb and 35 for 25kb, both of which were chosen after tuning them to get the greatest possible values for our metrics M_{gt} and M_{pred}.

Finally, note that threshold acceptance was done per boundary, not per TAD. This means that the very first coordinate to pass the threshold would automatically be considered a leftmost boundary and that the next one would be the rightmost boundary, even though those two boundaries might not necessarily have been associated to the same TAD on all methods.

3 Results

3.1 Results based on defined metrics

We produced TADs for our test set that include 7 chromosomes of each cell line and resolution. In total, we had 70 test chromosomes to analyse. In table 1, the mean values for the two metrics, length (in base pair) and the number of TADs found altogether/in total, are shown. The values are very similar to our train set results (table 2 in appendix A.1) with 160 chromosomes. We also compared the mean length with ArrowHead TADs (see Fig.6), shown to be very similar.

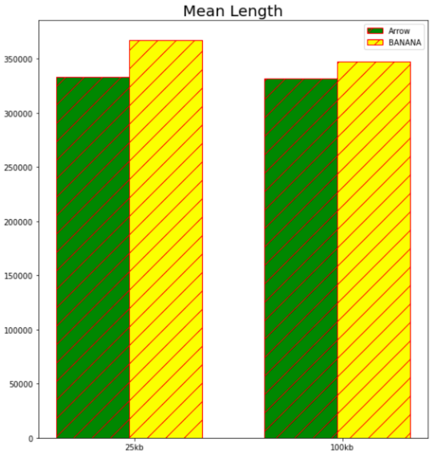


Figure 6: Plot ArrowHead vs Consensus on test set

Table 1: Comparison ArrowHead vs Consensus results on test set

	Mean M_{gt}	**Mean M_{pred}**	**Quantity**	**Mean Length**
25kb	0.749	0.605	20088	367230.4
100kb	0.795	0.568	22897	347552.1

4 Discussion

4.1 Comparison with other teams

The Warsaw team 1 analyzed their TADs by chromosome in terms of number, mean length, the area of the matrix covered by the intersection of ground truth and TopDom TADs compare to the area covered by the union of TADs, and the mean percent of match between two closest TADs between the two methods in one chromosome. We evaluated our consensus TADs with the same metrics and represented the results in the following plots.

We can see Banana-TAD outscoring WA1's results based on all type of metrics used. Banana-TAD's M_{gt} is 4 time better than WA1's M_{gt} and Banana-Tad's M_{pred} outscale WA1's M_{pred}. Banana-TAD seems clearly more able to correctly predict gold standard TADs while predicting less unknown TADs. These results were expected, regarding the fact we defined them ourselves and used them to tune Banana-TAD. Nevertheless, Banana-TAD also outperforms WA1's $Coverage$ and $Matching$, which is really promising regarding these results were proposed by WA1 and we were not aware of these metrics before comparing our performances with WA1 and we therefore never tuned our algorithm with them.

Performances on 25kb and 100kb resolution are mostly similar.

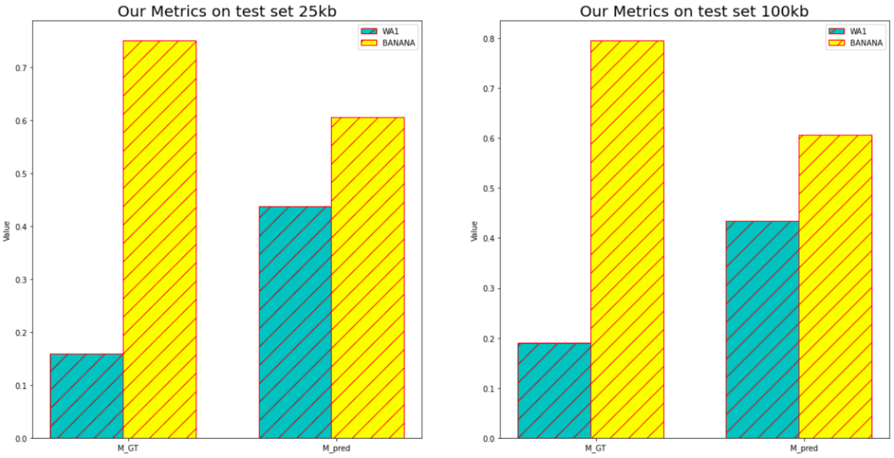


Figure 7: Comparison of our metrics' values between the two teams for 25kb and 100kb

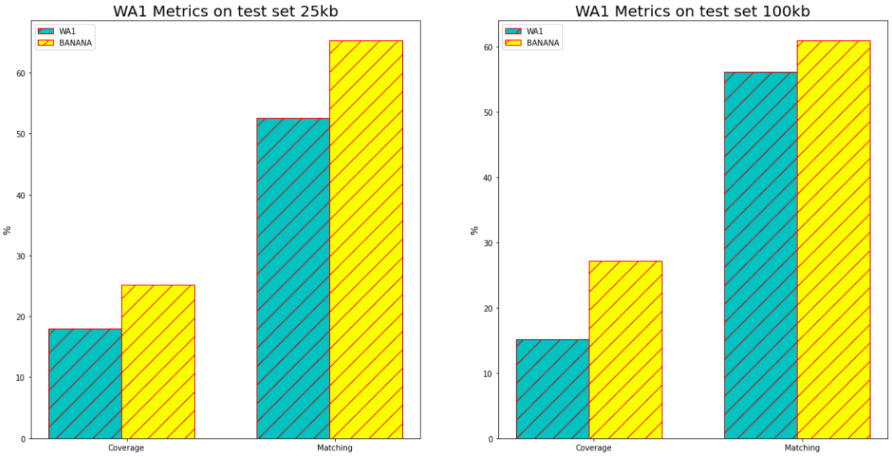


Figure 8: Comparison of WA1 metrics' values between the two teams for 25kb and 100 kb

4.2 On the Consensus Weight approach

Our weighted method has proven its ability to correctly predict TADs, despite a heavy computational time. During the *threshold* tuning, we saw the positive influence of each algorithm on the final performances.

Scores might be an interesting improvement to usual TAD caller as a confidence level about TAD's presence, as already proposed in some of them - including OnTAD [7]. It is also interesting to notice that CTCF peaks correspondence C_m and Adapted accuracy A_m of each TAD caller are highly correlated, confirming the previous studies showing the enrichment of CTCF binding sites at high rates around TAD boundaries [2] [9].

4.3 About Gaussian models

We analyze TADs length starts by the detection of the TADs by using different methods such as: TadTree, TopDom, or ArrowHead (described earlier).

We fetch these datas on a large number of TADs by performing TAD detection algorithms on all chromosomes of an organism. This will provide us with a sufficient test set. The resulting list of TADs are then processed by a fitting algorithm that relies on the Least

Square Minimization principle. The algorithm will create a Gaussian function that will try to approach our model. With a wide enough test set of TADs this method might accurately predict the distribution of number of TADs for each size in a genom.

We can see that different techniques produce different predicted models. For example the method used to create the previous TADs was TopDom with a window of 5. But the TADs found by using the ArrowHead method are different, because these results include smaller TADs. The model on which the fitting will be done is thus a double peak one. It is composed of 2 gaussian models merged together.

5 Conclusion

Overall, our approach to solving this problem yielded satisfactory results relative to our testing set, as well as when comparing to two other partner teams. We lacked the necessary time and computing power to properly implement all methods on all resolutions, which would likely have yielded more interesting results.

6 Bibliography

[1] Anne-Laure Valton and Job Dekker. TAD disruption as oncogenic driver. 36:34–40.

[2] Jesse R. Dixon, Siddarth Selvaraj, Feng Yue, Audrey Kim, Yan Li, Yin Shen, Ming Hu, Jun S. Liu, and Bing Ren. Topological domains in mammalian genomes identified by analysis of chromatin interactions. 485(7398):376–380.

[3] Suhas S.P. Rao, Miriam H. Huntley, Neva C. Durand, Elena K. Stamenova, Ivan D. Bochkov, James T. Robinson, Adrian L. Sanborn, Ido Machol, Arina D. Omer, Eric S. Lander, and Erez Lieberman Aiden. A 3d map of the human genome at kilobase resolution reveals principles of chromatin looping. 159(7):1665–1680.

[4] Hanjun Shin, Yi Shi, Chao Dai, Harianto Tjong, Ke Gong, Frank Alber, and Xianghong Jasmine Zhou. TopDom: an efficient and deterministic method for identifying topological domains in genomes. 44(7):e70–e70.

[5] Rola Dali and Mathieu Blanchette. A critical assessment of topologically associating domain prediction tools. 45(6):2994–3005.

[6] Caleb Weinreb and Benjamin J. Raphael. Identification of hierarchical chromatin domains. 32(11):1601–1609.

[7] Lin An, Tao Yang, Jiahao Yang, Johannes Nuebler, Guanjue Xiang, Ross C. Hardison, Qunhua Li, and Yu Zhang. OnTAD: hierarchical domain structure reveals the divergence of activity among TADs and boundaries. 20(1):282.

[8] François Serra, Davide Baù, Mike Goodstadt, David Castillo, Guillaume J. Filion, and Marc A. Marti-Renom. Automatic analysis and 3d-modelling of hi-c data using TADbit reveals structural features of the fly chromatin colors. 13(7):e1005665.

[9] Carlos Gómez-Marín, Juan J. Tena, Rafael D. Acemel, Macarena López-Mayorga, Silvia Naranjo, Elisa de la Calle-Mustienes, Ignacio Maeso, Leonardo Beccari, Ivy Aneas, Erika Vielmas, Paola Bovolenta, Marcelo A. Nobrega, Jaime Carvajal, and José Luis Gómez-Skarmeta. Evolutionary comparison reveals that diverging CTCF sites are signatures of ancestral topological associating domains borders. 112(24):7542–7547.

[10] Experiments – ENCODE. Available at https://www.encodeproject.org/experiments/.

A Appendix

A.1 Metrics on test dataset

Table 2: Comparison ArrowHead vs Consensus results on test set

	Mean M_{gt}	**Mean M_{pred}**	**Quantity**	**Mean Length**
25kb	0.751	0.580	60299	356582.6
100kb	0.789	0.569	51146	351321.7

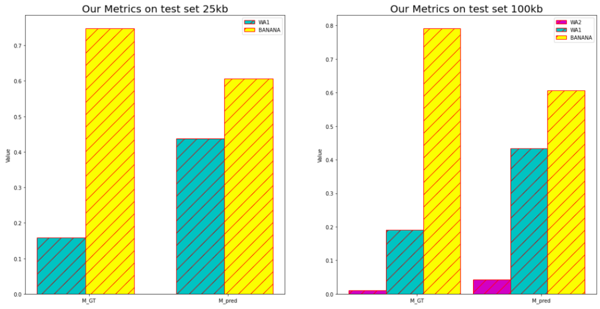


Figure 9: Comparison of our metrics' values between two teams for 25kb and three teams for 100 kb

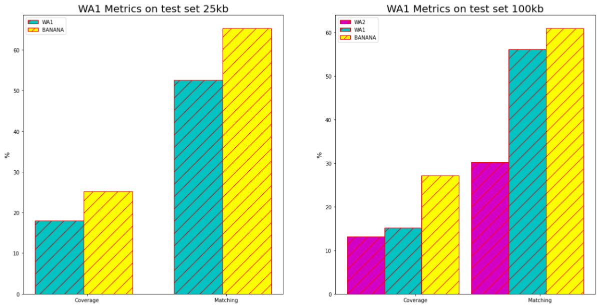


Figure 10: Comparison of WA1 metrics' values between two teams for 25kb and three teams for 100 kb

A.2 Gold standard- Arrowhead TAD size distribution

Consider merging two by two or all 4 in one big graphic, as the display has certain issues otherwise.

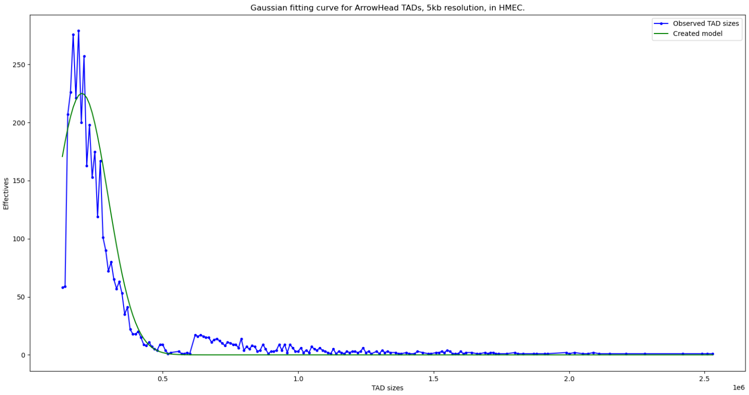


Figure 11: HMEC ArrowHead 5kb fitting

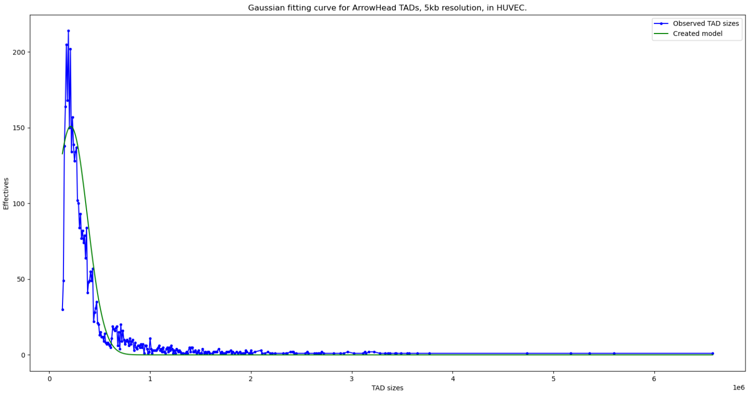


Figure 12: HUVEC$_{A}rrowHead_{5}kb_{f}itting$

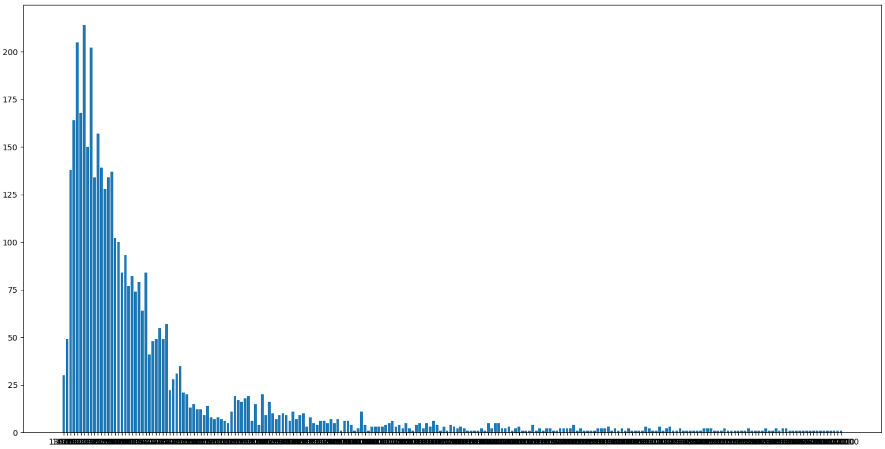


Figure 13: HUVEC ArrowHead 5kb fitting

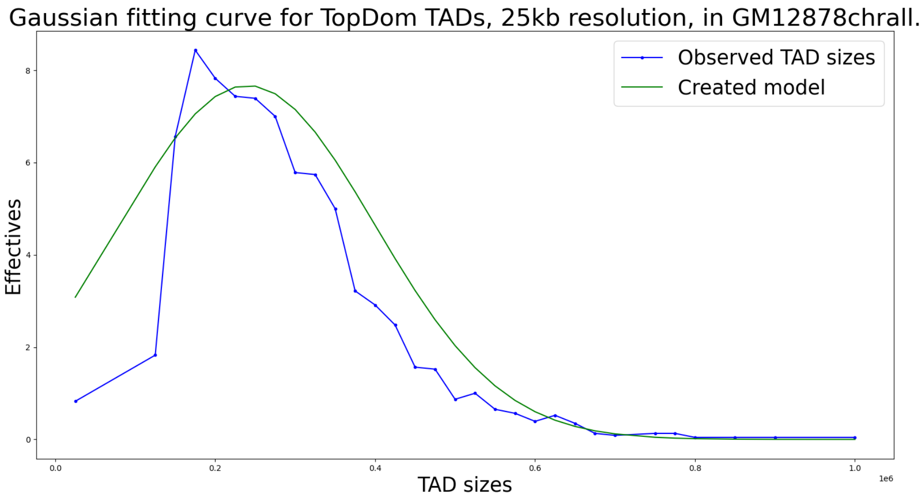


Figure 14: HUVEC ArrowHead 5kb fitting

Results_plots/All_ArrowHead_5kb_fitting.png

Figure 15: HUVEC ArrowHead 5kb fitting

Results_plots/All_ArrowHead_5kb_fitting_with_doubl

Figure 16: ll$_{A}rrowHead_{5}kb_{f}itting_{w}ith_{d}oublepeak$