2022-01-28

Prediction of TADs

Summary Report

Creator Team (TAD1@WA):

Sebastian Kot

Ignacy Makowski

Krzysztof Zdąbłasz

Leszek Troc

Partner Team (TAD1@SU):

Yann Zhong

Liam Le Goffic

Alexis Trang

Simon Crouzet

Marina Abakarova

Partner Team (TAD2@WA):

Anna Semik

Konrad Łukasz

Ewa Kizling

Julia Rymuza

Table of contents

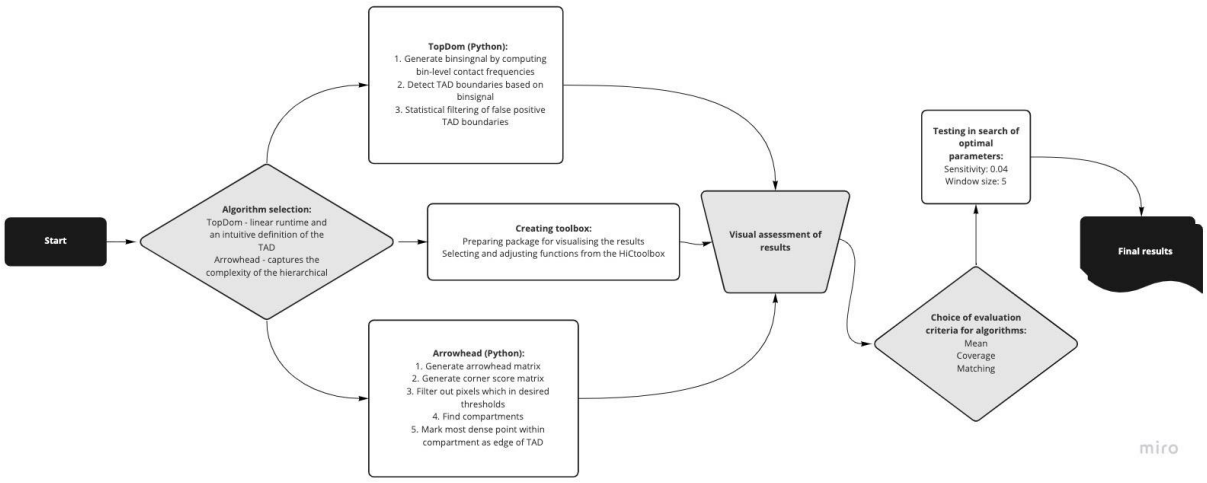
1. Introduction

Due to its size, the DNA is folded to form a chromatin. In order to achieve this, most of the eukaryotic organisms use histones with chaperone proteins[1]. Chromatin can be divided into euchromatin (loose, active, e.g., being expressed as protein) and heterochromatin (compacted, inactive form) and is organized in the 3D space into the Topologically Associating Domains (TADs), which are identified by the Hi-C experiments. Regions within the TADs are much more likely to interact with each other, than regions outside of the TADs do [2]. There are several tools that use Hi-C contact matrices to find the TADs. The matrices produced by the Hi-C experiment demonstrate intrachromosomal interactions. Each pixel represents all the interactions between two selected loci [3]. Our goal was to create an algorithm capable of detecting TADs from these data.

We started by looking into the available solutions. They vary i.a. in size and number of the TADs detected, time complexity and memory usage. Some of them also offer the possibility of detecting nested TADs. One of the state-of-the-art methods is a TopDom algorithm. It counts the sum of the interactions within windows along the Hi-C matrix diagonal, and then selects local minima with a specified p-value [4].

Our initial goal was simple. We decided to analyze two different algorithms in more depth, possibly modify them and for the final step, merge their results. The algorithms we chose are: Arrowhead, which captures the complexity of the hierarchical structure of TADs, and TopDom, which is characterized by a linear runtime and intuitive definition of the TAD boundary [5].

[1] (Biotech Research and Innovation Centre (BRIC) and Centre for Epigenetics, Faculty of Health and Medical Sciences, University of Copenhagen, Copenhagen DK-2200, Denmark, 2017)
[2] (McArthur & Capra, 2021)
[3] (Broad Institute of Harvard and MIT, 2009)
[4] (Molecular and Computational Biology, University of Southern California, Los Angeles, CA 90033, USA, 2015)
[5] (School of Computer Science and McGill Centre for Bioinformatics, McGill University, Montreal, Canada, 2017)


Start
Algorithm selection:
TopDom - linear runtime and an intuitive definition of the TAD
Arrowhead - captures the complexity of the hierarchical
TopDom (Python):
1. Generate binsingnal by computing bin-level contact frequencies
2. Detect TAD boundaries based on binsignal
3. Statistical filtering of false positive TAD boundaries
Creating toolbox:
Preparing package for visualising the results
Selecting and adjusting functions from the HiCtoolbox
Arrowhead (Python):
1. Generate arrowhead matrix
2. Generate corner score matrix
3. Filter out pixels which in desired thresholds
4. Find compartments
5. Mark most dense point within compartment as edge of TAD
Visual assessment of results
Testing in search of optimal parameters:
Sensitivity: 0.04
Window size: 5
Choice of evaluation criteria for algorithms:
Mean
Coverage
Matching
Final results
miro

2. Materials & Methods

Strategy and data

At the beginning of the project, as mentioned earlier, we decided to take the strategy of reimplementing two popular algorithms. We planned to compare and possibly merge their results, so that we could find the most reliable TADs. We evaluated their quality on data from the GM12878 cell line compared to the reference results of the original Arrowhead. Due to the availability of auxiliary tools, both algorithms were written by us in Python.

Modified TopDom

The original TopDom[6] consists of 3 stages:

1. Generate binsingnal by computing bin-level contact frequencies
2. Detect TAD boundaries based on binsignal
3. Statistical filtering of false positive TAD boundaries

Our version is as follows:

1. As a first step, we decided to use the original definition of binsignal as an average contact frequency among pairs of chromatin regions in a window surrounding the bin[7]. To achieve this, we used built-in functions and made use of numpy matrices to improve performance.
2. For this step, we developed an entirely new approach compared to the original algorithm. Our goal was to give users the ability to decide how precisely TADs are detected. Thus, we introduced the sensitivity parameter. It is simply a value between 0 and 1, that can be described as a threshold, below which fluctuations in binsignal are recognized as noise in data and are replaced with flat lines. For example, sensitivity=0.04 means that fluctuations below 4% will be ignored. This step can be divided into four substeps:
   2.1. Replace binsignal with local extremes curve

[6] (Molecular and Computational Biology, University of Southern California, Los Angeles, CA 90033, USA, 2015)
[7] (Molecular and Computational Biology, University of Southern California, Los Angeles, CA 90033, USA, 2015)

2.2. Calculate global maximum and minimum and multiply their difference by sensitivity parameter

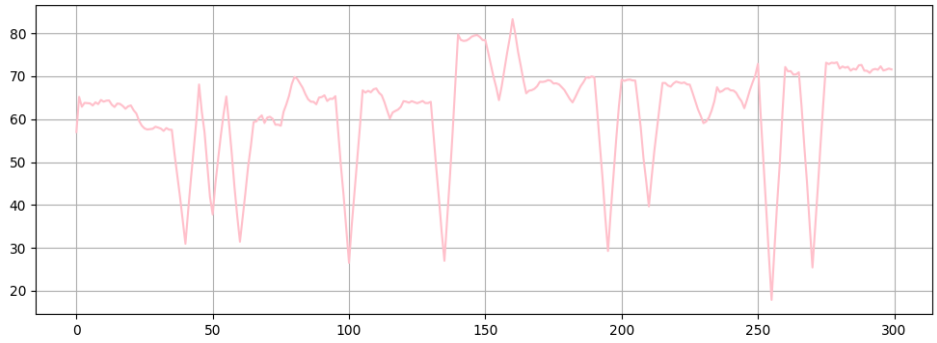


Figure 1. Original binsignal

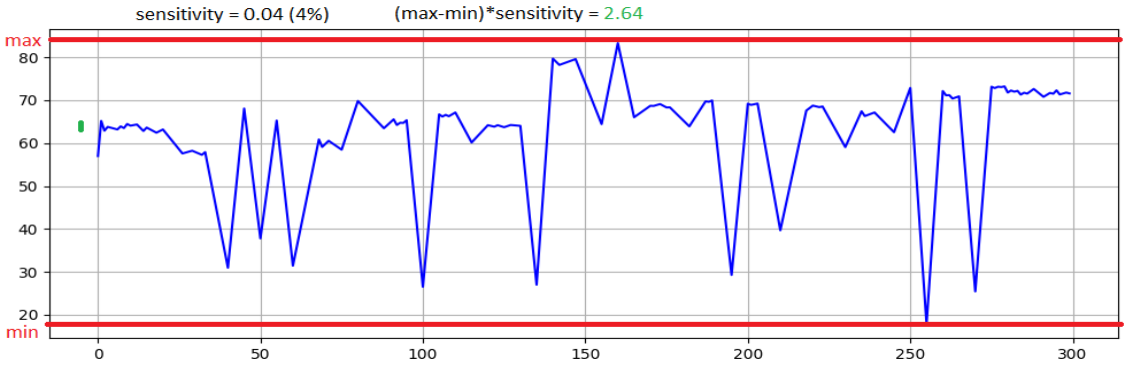


Figure 2. Local extremes curve

2.3. Filter out noise in the data by replacing regions with fluctuations below calculated number with flat lines

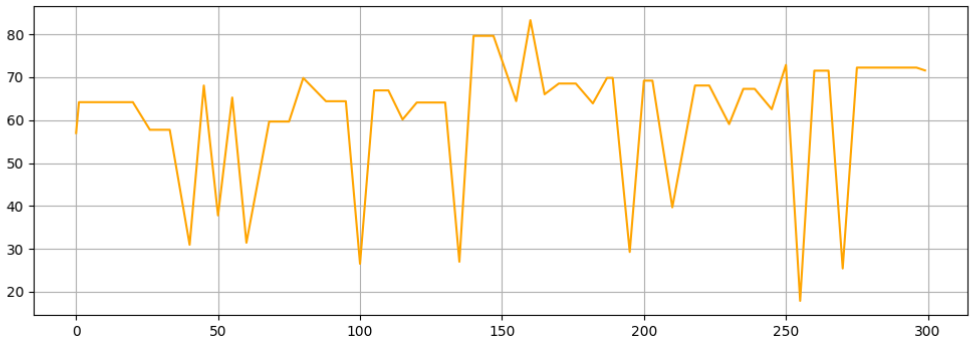


Figure 3. Flattened binsignal

2.4. Find minima (TAD boundaries)

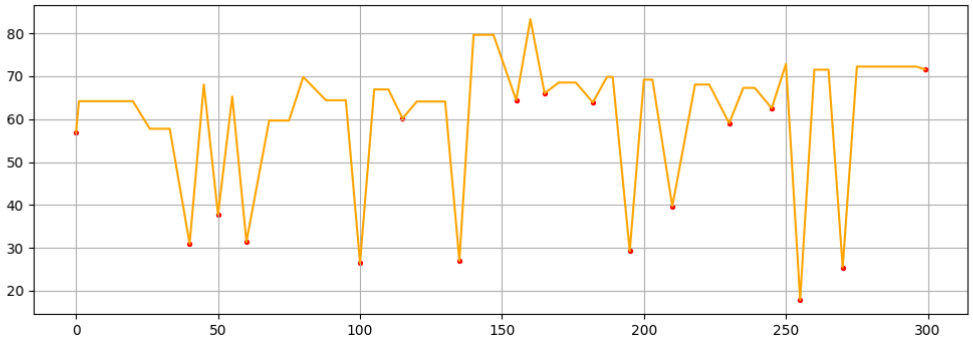


Figure 4. Flattened binsignal with marked minima (red dots)

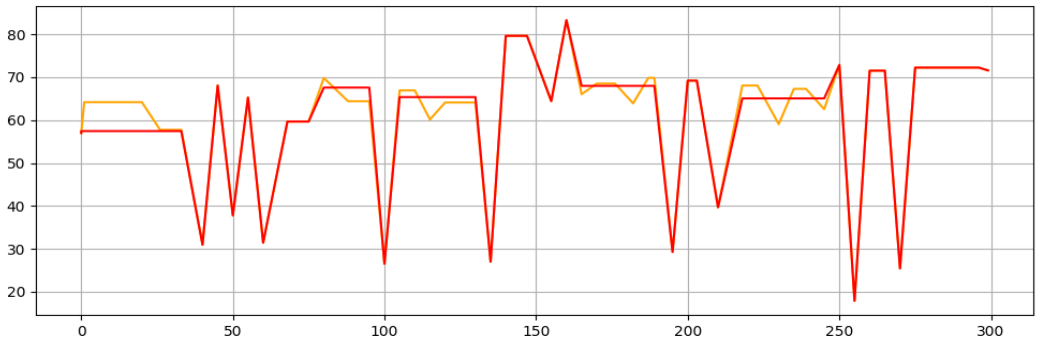


Figure 5. This figure exemplifies the same binsignal flattened with sensitivity=0.04 (orange line) and sensitivity=0.15 (red line)

3. In the final step we filter out the minima using the Wilcoxon test. It calculates the probability of interactions in [i - window size: i, i: i + window size] area coming from the same distribution as [i - window size: i, i - window size: i] and [i: i + window size, i: i + window size] areas. The function is implemented exactly as in the original TopDom.

Arrowhead

The Arrowhead implementation is strictly based on its description in paragraph IV.a of a document located at cell site[8].

We worked with 100kb intrachromosomal Hi-C data taken from official repository [9].

[8] https://www.cell.com/cms/10.1016/j.cell.2014.11.021/attachment/d3c6dcd8-c799-4f68-bbe4-201be54960b5/mmc1.pdf?fbclid=IwAR0T6nBsrpK74wKDMrcQ4qyEMBForKzJV2Y8zdqKacRkcPeSiZqK8fNcmms

[9] http://www.lcqb.upmc.fr/meetu/dataforstudent/HiC/GM12878/100kb_resolution_intrachromosomal

Steps of the Arrowhead algorithm:

1 Generate the arrowhead matrix
2 Generate the corner score matrix based on aggregating data in L and U triangles for each pixel of the arrowhead matrix
3 From the corner score matrix, filter out only the pixels which are within desired thresholds
4 Find compartments (groups of adjacent pixels which passed the filter step)
5 Mark most dense point within compartment as edge of TAD

When implemented correctly, this algorithm has $O(n^2)$ time and memory complexity, where n is the number of loci contained in the HiC data.

The goal was implementing this algorithm without modifications and potentially tweaking it later in order to produce better results. However, this appeared to be more difficult than it seemed at first, because many important implementation details are omitted in the article mentioned above. One example of this is generating the arrowhead matrix: the article provides a simple equation for calculating each value of this matrix, but completely ignores the fact that it uses elements with indices that can be negative (and therefore don't exist). Our implementation uses the (quite logical) assumption, that the value of these undefined elements is 0, but perhaps a different assumption could achieve better results.

Another moving part in the algorithm are the thresholds used to filter pixels which are potential TAD corners. Our experiments with these thresholds didn't yield any improvements, therefore the final implementation uses the numbers mentioned in the article.

Despite much tweaking and experimenting, the results achieved by our implementation of the arrowhead algorithm were highly unsatisfactory regardless of the metric used for assessment. Therefore, these results are disregarded in the rest of the report and our application defaults to only using the TopDom algorithm.

3. Results

Intro

We have been testing our algorithms in comparison to exemplary Arrowhead data provided at the beginning of the course. Unfortunately, the Arrowhead part performed very poorly, even after many adjustments to parameters and the algorithm itself. That is why we have resigned from joining the results of Arrowhead and TopDom and focused mainly on the TopDom part.

TopDom part was tested under each chromosome 1-22, X in 25kb and 100kb resolution and passed to TAD1@SU and TAD2@WA via repository. We have received results only from TAD2@WA therefore we compared our results only to theirs.

We have discussed integrating with TadKB database, but assuming that provided exemplary data can be (by the rules of the course) considered as the golden standard, this idea was dropped.

For the sake of the evaluation, we proposed 4 metrics:

- The number of TADs found and expected
- Mean size - mean length of a TAD (in bins)
- Coverage - if we project TADs to two dimensions how the surface of TADs behaves in comparison to exemplary data
- Matching - we try to match found TADs boundaries to the exemplary data, looking for the best fit for each TAD, and then we calculate how good is the match on average of all matches

Parameter testing

We have been adjusting parameters within given ranges:

- Sensitivity - 0.04 and 0.1
- Window Size - 5, 7, 15
- P-value threshold - 0.05 (this is the default value from the TopDom specifications, but our application gives the option of specifying any desired value)

Chromosome (25 kb res)	Parameters		Metrics					
			Number of TADs		Mean TAD size		Coverage	Matching
	Parameter	Value	found	expected	found	expected		
1	sensitivity	0.04	628	889	15.87	9.61	7.64%	72.14%
	window size	5						
22	sensitivity	0.04	118	177	17.36	9.94	2.46%	75.14%
	window size	5						
X	sensitivity	0.04	255	226	24.35	11.96	10.05%	47.84%
	window size	5						
1	sensitivity	0.04	516	889	19.32	9.61	8.18%	69.36%
	window size	7						
22	sensitivity	0.04	99	177	20.70	9.94	2.76%	74.31%
	window size	7						
X	sensitivity	0.04	218	226	28.49	11.96	9.90%	46.17%
	window size	7						
1	sensitivity	0.04	381	889	26.17	9.61	9.13%	65.26%
	window size	15						
22	sensitivity	0.04	75	177	27.32	9.94	3.36%	70.89%
	window size	15						
X	sensitivity	0.04	137	226	45.33	11.96	8.44%	42.49%

	window size	15						
1	sensitivity	0.1	350	889	28.48	9.61	6.23%	65.09%
	window size	5						
22	sensitivity	0.1	80	177	25.61	9.94	2.75%	73.21%
	window size	5						
X	sensitivity	0.1	114	226	54.47	11.96	5.48%	47.43%
	window size	5						
1	sensitivity	0.1	315	889	31.65	9.61	6.34%	64.48%
	window size	7						
22	sensitivity	0.1	71	177	28.86	9.94	3.03%	69.09%
	window size	7						
X	sensitivity	0.1	108	226	57.5	11.96	5.94%	39.71%
	window size	7						
1	sensitivity	0.1	224	889	44.50	9.61	6.76%	57.36%
	window size	15						
22	sensitivity	0.1	54	177	37.94	9.94	3.26%	68.05%
	window size	15						
X	sensitivity	0.1	47	226	31.72	11.96	4.55%	25.25%
	window size	15						

Chromosome (100 kb res)	Parameters		Metrics					
			Number of TADs		Mean TAD size		Coverage	Matching
	Parameter	Value	found	expected	found	expected		
1	sensitivity	0.04	311	889	8.01	2.40	8.78%	59.22%
	window size	5						
22	sensitivity	0.04	61	177	8.39	2.48	3.90%	61.72%
	window size	5						
X	sensitivity	0.04	74	226	10.59	2.99	11.07%	43.64%
	window size	5						
1	sensitivity	0.04	298	889	8.36	2.40	8.66%	60.17%
	window size	7						
22	sensitivity	0.04	58	177	8.83	2.48	3.97%	64.94%
	window size	7						
X	sensitivity	0.04	66	226	11.88	2.99	11.53%	38.18%
	window size	7						
1	sensitivity	0.04	244	889	10.21	2.40	8.29%	57.31%
	window size	15						
22	sensitivity	0.04	34	177	15.06	2.48	3.84%	57.65%
	window size	15						
X	sensitivity	0.04	43	226	18.23	2.99	7.88%	33.26%

	window size	15						
1	sensitivity	0.1	213	889	11.70	2.40	7.70%	53.01%
	window size	5						
22	sensitivity	0.1	42	177	12.19	2.48	3.92%	55.25%
	window size	5						
X	sensitivity	0.1	45	226	17.42	2.99	7.62%	38.39%
	window size	5						
1	sensitivity	0.1	179	889	13.92	2.40	7.21%	49.52%
	window size	7						
22	sensitivity	0.1	34	177	15.06	2.48	3.62%	55.64%
	window size	7						
X	sensitivity	0.1	38	226	20.63	2.99	6.41%	36.01%
	window size	7						
1	sensitivity	0.1	108	889	23.07	2.40	5.92%	36.32%
	window size	15						
22	sensitivity	0.1	18	177	28.44	2.48	3.31%	42.74%
	window size	15						
X	sensitivity	0.1	22	226	35.37	2.99	4.18%	22.11%
	window size	15						

Within the presented results, we can observe that increasing sensitivity and window size decreases results of our metrics significantly. We have discovered that the best fit for window size is the actual one from paper - 5, and for sensitivity - values around 0.04. As mentioned in the original TopDom publication, window size = 5 achieves the highest average PCC/wPCC scores, which can be considered a general guideline to determine the value of this parameter. However, the ideal value may differ for different genomes studied.

Regarding the sensitivity parameter, as mentioned earlier, it can be described as a threshold, below which fluctuations are recognized as noise in data. That being so, the value of 4% seems like a reasonable choice. Nevertheless, selection of this parameter should be well thought out and correlated with data resolution, not to get inaccurate or false positive results.

It is interesting to observe how low the overall assessment is against the coverage metric. The reason for this has less to do with the data and more with the metric itself. In its calculation, it takes into consideration how many contact points contained in the found TADs are also a part of the reference TADs. This number is divided by the number of pixels which are part of a TAD in our or reference data (or both). Even a few undetected larger TADs, as well as faulty TADs, can have a significant effect on the assessment value of this metric.

Coming back to the data, it is quite clear that the number of TADs we found is generally smaller than expected. This may be caused by the inherent properties of the TopDom: because of its relative simplicity (and therefore linear time complexity) it is not able to detect nested TADs as some other algorithms. However, one more important factor needs to be considered when assessing the results. The number of found domains is highly correlated with the data resolution, regardless of the algorithm [10]. We could not determine the resolution at which the golden data were generated. Hence, a certain amount of distrust needs to be maintained.

Contrary to the number of TADs, the mean length of found TADs is consistently higher than the expected mean length. The scaling factor ranges from about 2 to as high as 10. It increases when either of the parameters increases. This can also be related to the data resolution.

[10] (Zuffereye, Tavernari, Oricchio, & Ciriello, 2018)

Time of execution

TopDom is a linear algorithm ($O(w^2n)$, where w is the window size and n - number of bins). We were able to process all 100kb data in about 8 minutes, and all 25kb in about 30 minutes on a regular modern desktop machine without parallelism.

Comparison to partner team

When we compare our results to Team-WA2, we can see that TADs found by the other team are matching ours at an average of 81.85%. We have managed to find approximately two times more TADs on chromosomes 1 and 21. However, the other team detected 13 TADs more than we did on chromosome X. Despite the differences in those numbers, the mean size appears to stay the same with theirs/ours ratio of 1.5.

Chromosome (100 kb res)	**Parameters**		**Metrics**					
			Number of TADs		**Mean TAD size**		**Coverage**	**Matching**
	Parameter	**Value**	**ours**	**theirs**	**ours**	**theirs**		
1	sensitivity	0.04	311	165	8.01	12.78	17.86%	73.48%
	window size	5						
22	sensitivity	0.04	61	29	8.39	12.07	6.39%	89.58%
	window size	5						
X	sensitivity	0.04	74	87	10.59	15.63	21.30%	82.50%
	window size	5						

Flexibility

Our program is written in Python and supports various parameters to make it more modular, versatile, and portable. Python can run on many types of machines. We support the following arguments:

Parameter	Values	Description
--run-arrowhead	any -> True not provided -> False	We have implemented arrowhead but poorly. Algorithm can still be run with this parameter.
--run-topdom	any -> True not provided -> False	Main solution is TopDom. This parameter triggers it's processing.
--results-path	../results /var/opt/results	Where to put results, directories should be created before. Can be relative or absolute
--chromosomes	1 1,2,X	Comma separated list of chromosomes we would like to process
--with-metrics-results	any -> True not provided -> False	Should attach metrics results to files?
--with-results-coordinates	any -> True not provided -> False	Should attach results coordinates to files?
--with-expected-results-coordinates	any -> True not provided -> False	Should attach expected results to files?
--with-images-results	any -> True not provided -> False	Should attach images of results to files?
--topdom-normalization-alpha	0.12 1 not provided -> no normalisation	Optionally normalise matrix with alpha.
--resolution	25k 100k	Resolution of data used.
--topdom-sensitivity	(0,1)	Ignores fluctuations of the binsignal up to a selected fraction of its value
--topdom-window-size	Recommended [5,20]	Size to expand from bin while calculating binsignal
--topdom-pval-limit	Recommended 0.05	Completing the statistical significance of the boundaries found
--should-show	any -> True not provided -> False	Should additionally display results?
--should-use-other-results-as-golden	any -> True not provided -> False	Should switch comparison to other team results?
--provided-other-teams-results-path	../results/team-wa2 /mnt/team-wa2	Where to get results of teamwa2

--provided-arrowhead-results-path	../result/golden-standard/	Where to get results of exemplary Arrowhead.
--data-path	../data	Where to get HiC data.

4. Discussion

Working on this project allowed us to familiarize ourselves with a relatively new biological concept, known as the Topologically Associating Domains. We understand that detecting them is crucial for further development of research on the complex chromatin structure. Knowledge of its spatial organization will allow researchers to explain complicated and not yet fully understood dependencies between DNA regions and how this affect gene regulation.

The problem is that it is not possible to unambiguously define TAD boundaries. This concept is slightly abstract and different algorithms will return different results for the same data. While defining TADs, one can only take some clues into consideration (like the presence of CTCF or specific gene types nearby). Nevertheless, strong conservation between different cell types and even species, shows the important role that TADs play in cells.

During our work, we have encountered many difficulties. At the very beginning, we were slowed down by the fact that only half of our team has biological education on the academic level. Spatial organization of chromatin is not a basic knowledge, and it requires some time and effort to be well understood. The next obstacle was implementing the Arrowhead, as it is a rather complex algorithm. Our implementation returned results, but these were poor and did not overlap with reference data. For that reason, we were forced to abandon the initial idea, which was to modify both Arrowhead and TopDom and merge their results.

Our modified version of TopDom seems to perform well and for certain combinations of parameters can return reliable results. The sensitivity parameter is intuitive and easy to understand. For bigger values, only firmly marked boundaries will be detected. When it comes to lower values, smaller TADs can be found with better accuracy. However, the choice of this parameter should be well thought out. Numbers that are too low can lead to false positive results and for numbers way too big, huge areas will be returned. That will provide no valuable information for the analysis.

Our algorithm has vast potential for future development. One of its biggest advantages is linear execution time. Thanks to this, the algorithm can be easily modified without negatively influencing the computational costs. A possible way of development may be to implement multiple binsignal flattening with different values of the sensitivity parameter, which would allow recognizing nested TADs in the analyzed data. Another example of possible further modifications is to merge the results with

Arrowhead to benefit from the strong sides of both algorithms. Unfortunately, due to the lack of time, we were unable to implement these ideas.

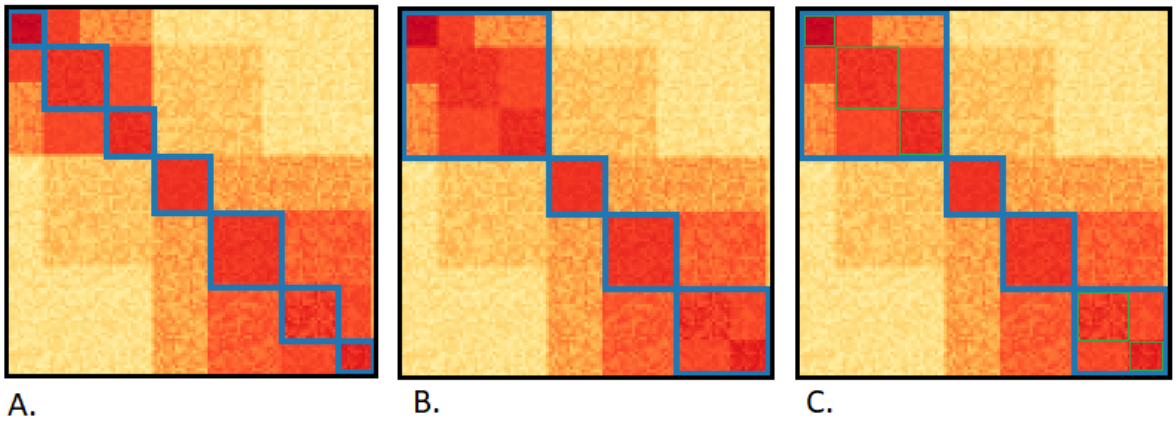


Figure 6. Visualization of possible future algorithm improvements. Illustration A shows TADs found in the toy data, with sensitivity=0.04. Illustration B shows the same data with sensitivity=0.1. Illustration C shows how these two outcomes could be used to recognize TAD complexes with nested TADs.

To summarise, thanks to this project, we were able to improve our interdisciplinary knowledge and dive into the most recent bioinformatics problems. It allowed us to practice our abilities to work as a team, as well as our programming skills. Despite not fulfilling all of our initial goals, we are satisfied with the results, and see many possible ways to improve our work in the future.

5. Bibliography

Biotech Research and Innovation Centre (BRIC) and Centre for Epigenetics, Faculty of Health and Medical Sciences, University of Copenhagen, Copenhagen DK-2200, Denmark. (2017). *Histone chaperone networks shaping chromatin function.* Retrieved January 28, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5319910/

Broad Institute of Harvard and MIT. (2009). *Comprehensive mapping of long range interactions reveals folding principles of the human genome.* Science. Retrieved January 28, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2858594

McArthur, E., & Capra, J. A. (2021). *Topologically associating domain boundaries.* The American Journal of Human Genetics. Retrieved January 28, 2022, from https://www.sciencedirect.com/science/article/pii/S000292972100001X

Molecular and Computational Biology, University of Southern California, Los Angeles, CA 90033, USA. (2015). *TopDom: an efficient and deterministic method for identifying topological domains in genomes.* Retrieved January 28, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4838359/

School of Computer Science and McGill Centre for Bioinformatics, McGill University, Montreal, Canada. (2017). *A critical assessment of topologically associating domain prediction tools.* Retrieved January 28, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5389712

Zuffereye, M., Tavernari, D., Oricchio, E., & Ciriello, G. (2018). *Comparison of computational methods for the identification of topologically associating domains.* Genome Biology. Retrieved January 28, 2022, from https://genomebiology.biomedcentral.com/articles/10.1186/s13059-018-1596-9