Thesis

Chromatin Compartments and Selection on X

Søren Jørgensen

2024-11-30

Chromatin Compartments and Selection on X

How Edges of Active Chromatin Align with Selection Regions in Primates

Søren Jørgensen

2024-11-30

Submitted in fulfillment of the requirements of the degree of

“3-dimensional structure of chromatin brings light onto the mystery of selfish genes.

- Søren Jørgensen ”

Table of contents

1 Introduction

1.1 Sexual reproduction (spermatogenesis, meiosis)

The production of gametes in a sexually reproducing organism is a highly complex process that involves numeruous elements. Spermatogenesis, the process of forming male gametes, involves four stages of differentiation from a germ cell through *spermatogonia*, *pachytene spermatocyte*, and *round spermatids* to *spermatozoa*, or *sperm* [8], and it is the very basis of male reproduction. The specialized cell division of meiosis neatly handles the pairing, recombination, and segregation of homologous chromosomes, thereby ensuring proper genetic distribution. Deeply understanding the steps of molecular steps of reproduction and how our genetic material is inherited is essential in biology, bringing insight to areas such as speciation, population diversity, and (male) infertility.

1.2 Selfish genes (and randomness)

The conventional story of meiosis in gametogenesis is one of random segregation of the sex chromosomes. They split into haploid gametes, where each chromosome has an equal chance of being passed on to a gamete. That seems like a fair game, but what if some genes are cheating the system by making others less viable. A meiotic driver is a selfish gene element that modulates meiosis and preferentially transmits its own allele through meiosis, regardless of the downstream fitness effects it may have (good or bad) on the organism it is part of. This phenomenon challenges the traditional understanding of selection, extending its scope beyond the fitness effects on an organism to include selective pressures at the molecular level. For example, if some genes on the X chromosome create a disadvantage for gametes that *do not* contain those genes, making sure the Y chromosome is not as viable as the X, resulting in a sex imbalance and possibly numerous other downstream effects. That is exactly what is coined *sex chromosome meiotic drive* [2], a result of selfish genetic elements.

Motivated by previous results in the Munch Research group [4] on hybrid incompatibility and extended common haplotypes [6, 7] that could be explained by meiotic drive, we wanted to investigate how these patterns correlate with chromatin compartments.

1.3 High-Throughput Chromosome Conformation Capture (Hi-C)

Our DNA can be divided into different orders of structure. *3C* focus on identifying the highest orders of organization inside the nucleus, that is, when the 30 nm thick coil of chromatin fibers folds into loops, Topologically Associating Domains (TADs), and chromatin compartments. Here, we narrow our focus on the largest of the structures, *compartments*, that is known to determine availability to transcription factors, thus making an *A* compartment *active*—and the *B* compartment *inactive*. The introduction of the Hi-C (high-throughput 3C) method [3] opened new possibilities for exploring the three-dimensional organization of the genome.

2 Methods

In this project, we formulate two objectives:

A: Reproduce the Hi-C interaction maps and eigendecomposition from [8], with some modifications. We briefly use *HiCExplorer*, but change the analyses to use the *Open2C Ecosystem* [5] which have a Pyton API as well as command-line functions, which can be paired very well with Jupyter Notebooks. The majority of the data analysis was run with a *gwf* workflow, and the commands that were visually inspected were run in Jupyter Notebooks.

B Compare with regions of selection that are found in *papio anubis*, and maybe in *human* too. Investigate the biological meaning of the results.

All computations were performed on GenomeDK (GDK) [ref], an HPC cluster located on Aarhus Uninversity, and most of the processing of the data was nested into a *gwf* workflow [ref], a workflow manager developed at GDK. I would like to thank GDK and Aarhus University for providing computational resources and support that contributed to these research results.

The whole of this project is carried out with reproducibility in mind, so an effort (and quite a significant amount of time) has been put into documenting code and organizing the project for readbility and transparency through a Quarto project [ref]. Therefore, all code, virtual environments and text is made available as a Quarto book, rendered directly from the GitHub repository with GitHub Pages [ref]. To make this possible, the Quarto documentation has been extensively studied and discussed with *KMT* [ref, aknowledge].

2.1 Downloading Data and Project Structure

To reproduce the results from [8], I chose to use their raw data directly from the SRA portal [ref]. I filtered the data to contain all their paired-end Hi-C reads, and included only macaque samples. The data set also contains RNAseq data, and the same tissues for both macaque and mouse. The meta data for the data set was extracted into a runtable `SRA-runtable.tsv`. To get an overview of the data accessions used in this analysis, we will first summarize the runtable that contains the accession numbers and some metadata for each sample (Table 2.1). It adds up to ~1Tb of compressed `fastq` files, holding ~9.5 billion reads, roughly evenly spread on the 5 tissue types.

	source_name	GB	Bases	Reads
0	fibroblast	211.403275	553,968,406,500	1,846,561,355
1	pachytene spermatocyte	274.835160	715,656,614,700	2,385,522,049
2	round spermatid	243.128044	655,938,457,200	2,186,461,524
3	sperm	164.131640	428,913,635,400	1,429,712,118
4	spermatogonia	192.794420	518,665,980,300	1,728,886,601

Table 2.1: Summary of the data accessions used in this analysis

Table 2.1

2.2 Handling coolers (Or: preparing coolers)

Figure 2.1: A flowchart showing the pipeline from `.fastq` to `.mcool`. The first 6 steps were done with a Probably BioRender or Inkscape.

2.2.1 The *gwf* workflow targets

A *gwf* workflow was created to handle the first part of the data processing, and each accesion number (read pair, mate pair) from the Hi-C sequencing was processed in parallel, so their execution was independent from each other.

2.2.1.1 Downloading the reads

The reads were downloaded from NCBI SRA portal [ref] directly to GDK using `sra-downloader` [ref] through docker [ref] as `.fastq.gz` files.

2.2.1.2 Handling the reference

Needs:

☒ *rheMac10*, `bwa` indexing
☒ Wang et al. used *rheMac2*, the first assembly of rhesus
☒ Bowtie2 indexing

The latest reference genome for rhesus macaque (*macaca mulata*), *rheMac10* (or *Mmul_10*, UCSC or NCBI naming conventions, respectively) was downloaded to GDK from UCSC web servers with `wget` [ref]. To use `bwa` (Burrow Wheeler's Aligner) [ref] for mapping, rheMac10 needs to be indexed with both `bwa index` with the `--bwtsw` option and `samtools faidx`, which results in six indexing files for `bwa mem` to use.

Since [2019], the reference genome for rhesus macaque has changed several times from *rheMac2* to *rheMac10*, each time resulting in a much less fragmented reference assembly. Part of the reasoning for reproducing their results was doing so on the latest assembly of the Macaca mulata genome, which arguably will result in a more accurate mapping of the reads, and a better inference of the chromatin compartments as well.

Several mappers were used in different configurations (described in below), and `bowtie2` requires its own indexing of the reference, using `bowtie2-build --large-index`, which creates six index files for `bowtie2` to use.

2.2.1.3 Mapping Hi-C reads

Needs:

☐ Genereal problems from standard PE reads to Hi-C reads (rescue chimeric fragments)
☒ HiCExplorer, bwa, mapping reads seperately
☐ Bowtie2 (local and end-to-end)
☐ Open2C formats and `bwa`

Reproduction It was not feasible to follow the same approach as Wang et al. [8] with both *HiCExplorer* and *Open2C*, as they use a third software, *HiC-Pro*. Hic-Pro uses bowtie2 in end-to-end mode, followed by remapping of 5'-ends of the unmapped reads to rescue chimeric fragments along with another approach. I argue that even though we are trying to reproduce results, it is nonsensical to use methods that are not state-of-the-art. That said, either widely used method should produce similar results.

- Hi-C is a rapidly evolving concept and a lot has changed in 5 years [since 2019]

HiCExplorer Initially, recommendations from HiCExplorer were used. According to their documentation [ref] it is crucial to 1) align reads locally, as Hi-C has a higher fraction of reads that are chimeric, and 2) mapping mates separately to mitigate some of the heuristics made by aligners for standard paired-end libraries. However, the resulting files were incompatible with the *open2C* ecosystem, and I therefore followed *HiCExplorer* pipeline to plot and explore the matrices created from this mapping. However, the work was laborious for experimentation, as the provided functions all write plots to files. I did not manage to make an efficient implementation

for plotting the `.h5` files produced by the pipeline, and I relatively quickly shifted to *Open2C* for their promises of the greener grass.

Open2C Suspiciously, Open Chromosome Collective [5] never mentions the

2.2.1.4 Pair and sort the reads

Needs:

- ☐ mapping mates separately vs. as paired-end reads
- ☐ `pairtools parse` and `pairtools sort`
- ☐ discuss the use of default parameters: docs

2.2.1.5 Filter (deduplicate) pairs

At this point we will remove all reads that are mapped to an unplaced scaffold. Even though the publication of *rhemac10* assembly states they have closed 99% of the gaps since *rhemac8*, *rheMac10* still contain more than 2,500 unplaced scaffolds, which are all uninformative when calculating the chromatin compartments as is the goal of this analysis. Therefore, we simply only include the list of conventional chromosomes (1..22, X, Y) when doing the deduplication. Initially, the default values were used to remove duplicates, where pairs with both sides mapped within 3 base pairs from each other are considered duplicates.

`cooler` recommend to store the most comprehensive and unfilteres list of pairs, and then applying a filter on it on the fly by piping from `pairtools select` I have missed this step, so I have not filtered for mapping quality. I will make a histogram showing the distribution of mapq scores to see the significance of this. Or just rerun that part of the analysis.

2.2.1.6 Create interaction matrices (coolers)

The final part of the *gwf* workflow takes `.pairs` as input and outputs a `.cool` file (*cooler*). Initially, we read directly from the newly generated deduplicated pairs without additional filtering, but here, the official recommendation is to filter out everything below $mapq = 30$ by piping the pairs through `pairtools select "(mapq1>=30) and (mapq2>=30)"` to `cooler cload pairs`.

We should have plenty of data to do the filtering, but I argue it is not strictly necessary. I will show a histogram of the *mapq* scores to convince you [ref]. Otherwise, I will have fixed this issue.

2.2.2 Notebook edits

As `cooler` and `cooltools` have a Python API, the more experimental parts of the analysis were moved to Jupyter Notebooks (still running on GenomeDK). `cooltools` comes with a helper library for operations on genomic intervals called `bioframe`.

2.2.2.1 Pooling samples (Merging coolers)

The samples are grouped into *replicates* with a unique **BioSample** ID, but we chose to pool all the interaction matrices for each cell type. We argue that when Wang et al. [8] determine compartments to be highly reproducible between replicates, by merging the replicates we can get a more robust signal.

`cooler merge` was used to merge all samples in each sub-folder (cell type) to just one interaction matrix for each cell type. The function merges matrices of the same dimensions by simply adding the interaction frequencies of each genomic position together, resulting in less empty positions by chance.

2.2.2.2 Create multi-resolution coolers (zoomify)

A feature of working inside the ecosystem of *Open2C* [ref] is that it natively provides support for storing sparse interaction matrices in multiple resolutions in the same file by adding groups to the cooler [ref]. We can then efficiently store resolutions (i.e., different bin sizes) that is multiples of the smallest bin size. We chose to use 10kb, 50kb, 100kb, and 500kb bins, and the resolutions are made by recursively binning the base resolution. We call this process zoomifying.

2.2.2.3 Matrix balancing (Iterative correction)

Finally, we balance the matrices using the cooler CLI. We use `cooler balance` with the default options which iteratively balances the matrix (Iterative Correction). It is first described as a method for bias correction of Hi-C matrices in [1], where it is paired with eigenvector decomposition, coining the combined analysis **ICE**. Here, the eigenvector decomposition of the obtained maps is experimentally validated to provide insights into local chromatin states.

[According to `cooler` documentation] We have to balance the matrices on each resolution, and thus it cannot be done prior to zoomifying. They state that the balancing weights are resolution-specific and will no longer retain its biological meaning when binned with other weights. Therefore, we apply `cooler balance` to each resolution separately. `cooler balance` will create a new column in the `bins` group of each cooler , `weight`, which can then be included or not in the downstream analysis. This means we will have access to both the balanced and the unbalanced matrix.

The default mode uses genome-wide data to calculate the weights for each bin. It would maybe be more suitable to calculate the weights for *cis* contacts only, and that is possible through the `--cis-only` flag, and that can be added to another column, so that we can compare the difference between the two methods easily. However, we will only use the default mode for now.

2.2.2.4 Eigendecomposition

The eigendecomposition of a Hi-C interaction matrix is performed in multiple steps. As value of the eigenvector is only *significant* up to a sign, it is convention [ref] to use GC content as a phasing track to orient the vector. E1 is arbitrarily defined to be positively correlated with GC content, meaning a positive E1 value signifies an active chromatin state, which we denote a A-type compartment (or simply A-compartment). We performed eigendecomposition of two resolutions, 100 Kbp and 500 Kbp.

First, we calculate the GC content of each bin of the reference genome, *rheMac10*, which is binned to the resolution of the Hi-C matrix we are handling. It is done with `bioframe.frac_gc` (*Open2C*). To calculate the E1 compartments, we use only within-chromosome contacts (*cis*), as we are not interested in the genome-wide contacts. `cooltools.eigs_cis` will decorrelate the contact-frequency by distance before performing the eigendecomposition. `eigs_cis` needs a *viewframe* (view) to calculate E1 values, the simplest view being the full chromosome. However, when there is more variance between chromosome arms than within arms, the sign of the first eigenvector will be determined largely by the chromosome arm it sits on, and not by the chromatin compartments. To mitigate this, we apply a chromosome-arm-partitioned view of the chromosome (as a bedlike format, described in `bioframe` docs [ref]).

2.2.2.5 Plotting

Needs:

- ☐ HiCExplorer: plot with CLI, writes to pngs
- ☐ cooler: plot through python API (fetch matrix from cooler)

3 Results

Here are the glorious results

4 Discussion

Here is the discussion

Bibliography

[1] Maxim Imakaev, Geoffrey Fudenberg, Rachel Patton McCord, Natalia Naumova, Anton Goloborodko, Bryan R Lajoie, Job Dekker, and Leonid A Mirny. Iterative correction of Hi-C data reveals hallmarks of chromosome organization. *Nature Methods*, 9(10):999–1003, October 2012. ISSN 1548-7091, 1548-7105. doi: 10.1038/nmeth.2148. URL https://www.nature.com/articles/nmeth.2148.

[2] John Jaenike. Sex Chromosome Meiotic Drive. *Annual Review of Ecology and Systematics*, 32 (1):25–49, November 2001. ISSN 0066-4162. doi: 10.1146/annurev.ecolsys.32.081501.113958. URL https://www.annualreviews.org/doi/10.1146/annurev.ecolsys.32.081501.113958.

[3] Erez Lieberman-Aiden, Nynke L. Van Berkum, Louise Williams, Maxim Imakaev, Tobias Ragoczy, Agnes Telling, Ido Amit, Bryan R. Lajoie, Peter J. Sabo, Michael O. Dorschner, Richard Sandstrom, Bradley Bernstein, M. A. Bender, Mark Groudine, Andreas Gnirke, John Stamatoyannopoulos, Leonid A. Mirny, Eric S. Lander, and Job Dekker. Comprehensive Mapping of Long-Range Interactions Reveals Folding Principles of the Human Genome. *Science*, 326(5950):289–293, October 2009. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1181369. URL https://www.science.org/doi/10.1126/science.1181369.

[4] Kasper Munch. Munch-group, November 2024. URL https://munch-group.org/research.html.

[5] Open Chromosome Collective. Open Chromosome Collective (Open2C), 2024. URL https://open2c.github.io/.

[6] Laurits Skov, Moisès Coll Macià, Elise Anne Lucotte, Maria Izabel Alves Cavassim, David Castellano, Mikkel Heide Schierup, and Kasper Munch. Extraordinary selection on the human X chromosome associated with archaic admixture. *Cell Genomics*, 3(3):100274, March 2023. ISSN 2666979X. doi: 10.1016/j.xgen.2023.100274. URL https://linkinghub.elsevier.com/retrieve/pii/S2666979X23000344.

[7] Erik F. Sørensen, R. Alan Harris, Liye Zhang, Muthuswamy Raveendran, Lukas F. K. Kuderna, Jerilyn A. Walker, Jessica M. Storer, Martin Kuhlwilm, Claudia Fontsere, Lakshmi Seshadri, Christina M. Bergey, Andrew S. Burrell, Juraj Bergman, Jane E. Phillips-Conroy, Fekadu Shiferaw, Kenneth L. Chiou, Idrissa S. Chuma, Julius D. Keyyu, Julia Fischer, Marie-Claude Gingras, Sejal Salvi, Harshavardhan Doddapaneni, Mikkel H. Schierup, Mark A. Batzer, Clifford J. Jolly, Sascha Knauf, Dietmar Zinner, Kyle K.-H. Farh, Tomas Marques-Bonet, Kasper Munch, Christian Roos, and Jeffrey Rogers. Genome-wide coancestry reveals details of ancient and recent male-driven reticulation in baboons. *Science*,

380(6648):eabn8153, June 2023. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.abn8153. URL https://www.science.org/doi/10.1126/science.abn8153.

[8] Yao Wang, Hanben Wang, Yu Zhang, Zhenhai Du, Wei Si, Suixing Fan, Dongdong Qin, Mei Wang, Yanchao Duan, Lufan Li, Yuying Jiao, Yuanyuan Li, Qiujun Wang, Qinghua Shi, Xin Wu, and Wei Xie. Reprogramming of Meiotic Chromatin Architecture during Spermatogenesis. *Molecular Cell*, 73(3):547–561.e6, February 2019. ISSN 10972765. doi: 10.1016/j.molcel.2018.11.019. URL https://linkinghub.elsevier.com/retrieve/pii/S1097276518309894.

Bioinformatics Research Centre
Department of Molecular Biology and Genetics
Aarhus University
Universitetsbyen 81
8000 Aarhus C
Denmark