Hi-C Data Analysis

Leila Taher

WS 2022/2023

Contents

1 Introduction

Three-dimensional (3D) genome architecture allows linearly distal genomic loci to interact with one another (eg enhancers with promoters), making it a key player in gene regulation. High-throughput chromosome conformation capture (Hi-C), have enabled us to catalog genomic interactions and features of 3D genome architecture in various cell types.

Hi-C data are typically summarized in a Hi-C matrix that estimates the probability of interaction between any two loci in the genome. Early Hi-C studies, observed that the genome segregates into so called "A" and "B" compartments based on transciptional states. They constitute for the most variation in HiC interaction data and are therefore defined by a PCA.

Short range intra-domain interactions visible on the diagonal of a HiC contact map and were originally identified by quantifying the directionality of interactions on data in the range of 0.2 - 1Mbp and were

termed "Topologically Associating Domains" (TADs). Many such domains are flanked by CTCF binding sites in convergent orientation corresponding to CTCF/cohesin mediated loop extrusion (CTCF loops).
Whereas A and B compartments correspond to domains that interact preferentially with sequences in other A or B compartment regions, respectively, TADs correspond to sequences that interact preferentially with themselves, rather than with other regions of the genome.
The aim of this tutorial is to generate and visualize a Hi-C contact matrix to depict the 3D organization of the Drosophilia melanogaster genome inside the nucelus. Specifically, you will:

1. Map Hi-C reads to the reference genome
2. Construct a Hi-C matrix
3. Visualize the Hi-C matrix
4. Correct and normalize the Hi-C matrix
5. Determine A and B compartments
6. Call topologically associating domains (TADs)

1.1 Tools

- FastQC
- Bowtie2
- Samtools
- HiCExplorer
  - `conda install hicexplorer -c bioconda -c conda-forge`
- IGV

`HiCExplorer` is a powerful and easy to use set of tools to process, normalize and visualize Hi-C data. Install the tool(s) if necessary.

1.2 Data

You will be working with Hi-C data obtained from the fruit fly *Drosophila melanogaster* Schneider 2 (S2) cells. S2 cells are one of the most commonly used *Drosophila melanogaster* cell lines. The data was generated by Ramirez et al. You can find more information here.

Sample	FASTQ file
S2 cells (forward reads)	HiC_S2_1p_10min_lowU_R1.fastq.gz
S2 cells (reverse reads)	HiC_S2_1p_10min_lowU_R2.fastq.gz

The files contain a subset of the original sequencing reads. You can find the original data in the NCBI Gene Expression Omnibus (GEO) database under accession GSE97965.

2 Quality control

We will use FastQC to generate a report about the quality of our sequencing reads. FastQC is a java application designed to spot potential problems in high throughput sequencing data sets.

1. Create a folder to store the output and run FastQC for each `.fastq` file. FastQC will generate two files for each input file, one compressed, and one not.
2. Examine the `.html` files.

3 Mapping the Hi-C reads to the reference genome

Although there are more appropriate tools for Hi-C data, in this tutorial we will map the reads using Bowtie2.

3.1 Indexing the genome

First you need to download the sequence of the *Drosophila melanogaster* reference genome assembly.
The genome of *Drosophila melanogaster* was first sequenced in 2000 and contains four pairs of chromosomes: an X/Y pair, and three autosomes labeled 2, 3, and 4. Females have two X chromosomes and males one X and one Y chromosome. Both sexes have two sets of the autosomal chromosomes 2, 3, and 4. The X is divided into two arms by the position of centromere, a large left arm (XL) and a much smaller right arm (XR), and is thus acrocentric. The Y is also acrocentric with a slightly longer long arm (YL) and a short arm (YS). Chromosomes 2 and 3 are metacentric with the centromere residing in the center of two roughly equal *left* (L) and *right* (R) arms. Chromosomes X, Y, and 4 are acrocentric.
The latest BDGP6.32 of the *Drosophila melanogaster* genome assembly includes a total of 143 Mbp on 1870 scaffolds comprising 2442 contigs. The vast majority, 137.6 Mb, of this sequence resides on seven chromosomes/chromosome arms: X, 2L, 2R, 3L, 3R, 4 and Y.

3. Download the soft-masked top-level sequence file of the BDGP6 assembly of the *Drosophila melangaster* genome from the Ensembl website (Ensembl release 105, `http://ftp.ensembl.org/pub/release-105/fasta/drosophila_melanogaster/dna/`). Note the “README” file explaining the directory structure.

4. Create a folder called “BDGP6” and store the resulting (uncompressed) file in this folder.

5. index the genome for mapping:

```
bowtie2-build BDGP6/Drosophila_melanogaster.BDGP6.32.dna_sm.toplevel.fa \
./BDGP6/Drosophila_melanogaster.BDGP6.32.dna_sm.toplevel \
--threads 2
```

   - Note that you may need to include the complete paths to the tool and input files.
   - Bowtie/Bowtie2 indexes the reference genome using a scheme based on the Burrows-Wheeler transform (BWT) and the FM index. This index permits fast substring queries while keeping a relatively small memory footprint.
   - For more details, please refer to the Bowtie2 manual.

3.2 Running Bowtie2

The forward and reverse reads of each pair must be mapped individually to avoid mapper specific heuristics designed for standard paired-end libraries.

6. generate a directory `Mappings` to save the mapped reads

7. to map the forward reads:

```
bowtie2 -x BDGP6/Drosophila_melanogaster.BDGP6.32.dna_sm.toplevel \
        -U data/HiC_S2_1p_10min_lowU_R1.fastq.gz --threads 2 \
        --reorder --very-sensitive-local | samtools view -Shb \
        > Mappings/HiC_S2_1p_10min_lowU_R1.bam
```

   - `-x` precedes the basename of the index for the reference genome.
   - `./BDGP6/Drosophila_melanogaster.BDGP6.32.dna_sm.toplevel` specifies the path to the index for the reference genome.
   - `-U` precedes a comma-separated list of files containing unpaired reads to be mapped.

- `HiC_S2_1p_10min_lowU_R1.fastq.gz` specifies the path to the `.fastq` file(s) with the reads.

8. Do the same for the reverse reads

9. What is the purpose of the `--reorder` parameter?

10. Why do we use local instead of end-to-end alignment?

11. What is the difference between `--local` and `--very-sensitive-local` and why would we want to use the latter option?

    For more details, please refer to the Bowtie2 manual.

12. What is the purpose of Samtools in this command?

13. What percent of the reads have been successfully mapped to the reference genome?

4 Constructing the Hi-C contact matrix

To construct the Hi-C contact matrix we will use the `hicBuildMatrix` tool.

14. The study used DpnII as the restriction enzyme to cleave the DNA. Search for the sequence of the restriction site (restriction sequence) as well as the sequence left by the restriction enzyme after cutting for this specific enzyme (dangling sequence). We will need these as input parameters for HiCExplorer's `hicBuildMatrix` function.

15. Next, use the `hicFindRestSite` to find restriction sites in our reference sequence and save their locations to a `bed` file:

```
hicFindRestSite -fasta BDGP6/Drosophila_melanogaster.BDGP6.32.dna_sm.toplevel.fa \
        -searchPattern <restriction_sequence_DpnII> -o BDGP6/restriction_sites.bed
```

16. Create a folder called "hicMatrix" to store the output files and use `hicBuildMatrix` to create a matrix of interactions.

```
hicBuildMatrix --samFiles Mappings/HiC_S2_1p_10min_lowU_R1.bam \
        Mappings/HiC_S2_1p_10min_lowU_R2.bam \
        --binSize 10000 \
        --restrictionSequence <restriction_sequence_DpnII> \
        --restrictionCutFile BDGP6/restriction_sites.bed \
        --danglingSequence <dangling_sequence_DpnII> \
        --outBam hicMatrix/HiC_S2_1p_10min_lowU_valid.bam \
        --outFileName hicMatrix/HiC_S2_1p_10min_lowU_10kbp.h5 \
        --QCfolder hicMatrix/ \
        --threads 2
```

- `--samFiles` specifies the two `.sam` or `.bam` files with the mapped reads to be processed.
- `--binSize 10000` indicates that the size for the genomic bins should be 10kbp, so that one value in the matrix represents the contact between two genomic loci of this length. The bin size depends on the depth of sequencing. If not given, `hicBuildMatrix` builds a matrix of restriction site resolution.
- `--restrictionSequence` specifies the sequence of the restriction site. Multiple restriction sites can be used, separated by a space.
- `--restrictionCutFile` specifies the BED file(s) with all restriction cut sites (output of "hicFindRestSite" command).

- `--danglingSequence` specifies the sequence left by the restriction enzyme after cutting, if multiple are used, please list them space seperated and preserve the order. Each restriction enzyme recognizes a different DNA sequence and, after cutting, they leave behind a specific "sticky" end or dangling end sequence.
- `--outBam ./hicMatrix/HiC_S2_1p_10min_lowU_valid.bam` instructs HiCExplorer to create a `.bam` file containing only valid Hi-C reads. This `.bam` file can be useful to inspect the distribution of valid Hi-C reads pairs or for other downstream analyses, but is not used by HiCExplorer.
- `--outFileName ./hicMatrix/HiC_S2_1p_10min_lowU_10kb.h5` is the name for the output Hi-C matrix file. Please note that the file type extension for the output matrix must be specified.
- `--QCfolder ./hicMatrix/` is the path to the folder where the quality control data for the matrix will be saved.
- note: don't worry if you get the warning message
  `Could not retrieve index file for 'Mappings/HiC_S2_1p_10min_lowU_R1.bam'`
  this will have no effect on the output

17. Examine the `.html` file in the QC folder. How many read pairs are considered valid?
18. Use IGV to verify that valid reads are found in the neighborhood of the appropriate restriction sites.

4.1 Visualizing the Hi-C matrix

A 10kb bin-based matrix is quite large to plot and is better to reduce the resolution. We usually run out of memory for a 1 kb or a 10 kb matrix and the time to plot is very long (minutes instead of seconds). If you want to know the size of a Hi-C contact matrix you can use the `hicInfo` tool.

The `hicMergeMatrixBins` tool merges the bins of a Hi-C contact matrix into larger bins. The option `numBins` allows the user to specify the number of bins in the new matrix.

19. If we want to go from a 10kbp resolution matrix to a 100kbp resolution matrix, how many bins N must be merged from the original matrix?

```
hicMergeMatrixBins \
        --matrix hicMatrix/HiC_S2_1p_10min_lowU_10kbp.h5 \
        --numBins <N> \
        --outFileName hicMatrix/HiC_S2_1p_10min_lowU_100kbp.h5
```

20. Generate a directory `Plots` and use the `hicPlotMatrix` tool to visualize the merged matrix:

```
hicPlotMatrix \
        --matrix hicMatrix/HiC_S2_1p_10min_lowU_100kbp.h5 \
        --log \
        --dpi 300 \
        --clearMaskedBins \
        --chromosomeOrder 2L 2R 3L 3R 4 X Y \
        --colorMap jet \
        --title "Hi-C contact map 100kbp" \
        --outFileName Plots/HiC_S2_1p_10min_lowU_100kbp.png
```

- `--matrix` specifies the matrix to plot.
- `--log` plots minus log of the matrix values.
- `--dpi 300` sets the resolution for the image to 300.
- `--clearMaskedBins` removes masked bins from the matrix and extends the nearest bins to cover the empty space instead of plotting black lines.

- `--chromosomeOrder` enumerates the chromosomes in the order in which they should be plotted.
- `--colorMap` is the color map to use for the heatmap. You can see the available values here.
- `--title` specifies the title of the plot.
- `--outFileName` determines the name of the file in which to save the image.
- `--perChromosome` plots chromosomes individually

21. Take a look at the generated contact maps.
    - what is plotted on the x an y axis?
    - what does a high value far off from the diagonal imply?

22. Now plot for each chromosome individually. Decide if you want to exclude any chromosomes from further analyses.

5 Correcting the Hi-C matrix

Correcting HiC contact maps is an important step in order to remove technical and experimental biases. The `hicCorrectMatrix` function provides 2 balancing methods which can be applied on a raw matrix.

i. KR: balances a matrix using a fast balancing algorithm introduced by Knight and Ruiz (2013).

ii. ICE: Iterative correction of a Hi-C matrix (Imakaev et al. (2012)). For this method to work correctly, bins with zero reads assigned to them should be removed as they cannot be corrected. Also, bins with low number of reads should be removed, otherwise, during the correction step, the counts associated with those bins will be amplified (usually, zero and low coverage bins tend to contain repetitive regions). Bins with extremely high number of reads can also be removed from the correction as they may represent copy number variations.

Matrix correction involves two steps:

i. Generating a histogram of the sum of contacts per bin (i.e., row sums); this histogram is used to decide on the threshold for removing bins with small numbers of contacts.

ii. Removing the bins with small numbers of contacts and performing the normalization.

23. First, generate the histogram:

    ```
    hicCorrectMatrix diagnostic_plot \
            --chromosomes 2L 2R 3L 3R 4 X Y \
            --matrix hicMatrix/HiC_S2_1p_10min_lowU_100kbp.h5 \
            --plotName Plots/HiC_S2_1p_10min_lowU_100kbp.h5_diagnostic_plot.png
    ```

    The program prints a threshold suggestion that is usually accurate, but is better to examine the histogram. The threshold is visualized in the histogram as a blue vertical line.

24. Use `hicCorrectMatrix correct` to remove the bins that do not satisfy your chosen threshold and perform the correction:

    ```
    hicCorrectMatrix correct \
            --matrix hicMatrix/HiC_S2_1p_10min_lowU_100kbp.h5 \
            --chromosomes 2L 2R 3L 3R 4 X Y \
            --perchr \
            --correctionMethod KR \
            --filterThreshold <min> <max> \
            --outFileName hicMatrix/HiC_S2_1p_10min_lowU_100kbp_corrected.h5
    ```

Hint Think about excluding specific chromosomes.

- `--perchr` instructs HiCExplorer to correct each chromosome separately.
- `--chromosomes` should be followed by the list of chromosomes to be included in the iterative correction.
- `--filterThreshold` removes bins of low or large coverage. Usually these bins do not contain valid Hi-C data or represent regions that accumulate reads and thus must be discarded. A lower and upper threshold are required separated by space and they should correspond to z-scores.

The z-score is the number of standard deviations by which the raw value is above or below the mean value of what is being observed or measured:

$$z\text{-score} = \frac{x - \mu}{\sigma}$$

- Examine the histogram and decide on the two thresholds.

If the correction stops with the following error:

```
‘ERROR:iterative correction:*Error* matrix correction produced extremely large values.
This is often caused by bins of low counts. Use a more stringent filtering of bins.’
```

use a more stringent z-score (eventually exclude entire chromosomes).

25. Visualize the corrected matrix per chromosome.

6 Determining A and B compartments

When scientists first came up with experiments which enabled them to visualize these genome wide contact maps, they saw many large blocks of enriched and depleted interactions generating a "plaid" pattern. If two loci (here 1 Mb regions) are nearby in space, they reasoned that they will share neighbors and have correlated interaction profiles and therefore defined a correlation matrix $\hat{C}$. Each entry in $\hat{C}$, $c_{i,j}$ is the Pearson correlation between the ith row and jth column of the original contact matrix. This process dramatically sharpens the plaid pattern.

PCA is then performed on the HiC matrix and the sign of (usually the first) principal component has been shown to correspond to A and B compartments.

26. Perform this process of 1) generating the pearson correlation matrix and then 2) extracting the first principle component with the function `hicPCA` to identify A and B compartments in our HiC data set.

27. Run `hicPCA` adjusting the parameters:

```
hicPCA --matrix hicMatrix/HiC_S2_1p_10min_lowU_100kbp_corrected.h5 \
        --whichEigenvectors "1"\
        --pearsonMatrix hicMatrix/HiC_S2_1p_10min_lowU_100kbp_corrected_pearson.h5 \
        --format bigwig \
        --ignoreMaskedBins \
        --outputFileName hicMatrix/HiC_S2_1p_10min_lowU_100kbp_corrected_pc1.bw
```

- `--whichEigenvectors` instructs HiCExplorer to compute the first principal component (PC1).
- `--pearsonMatrix` writes the internally computed Pearson matrix to a `h5` file.

- `--outputFileName` specifies the file name(s) for the result of the PCA. The number of output files must match the number of computed PCs.
- `--ignoreMaskedBins` Mask bins are usually set to 0. This option removes the masked bins before the PCA is computed.

28. Now run `hicPlotMatrix` to visualize the Pearson matrix together with the PC scores:

```
hicPlotMatrix --matrix hicMatrix/HiC_S2_1p_10min_lowU_100kbp_corrected_pearson.h5 \
        --perChr \
        --chromosomes 2L 2R 3L 3R X \
        --colorMap hot \
        --bigwig hicMatrix/HiC_S2_1p_10min_lowU_100kbp_corrected_pc1.bw \
        --title "Pearson matrix and PC1" \
        --outFileName Plots/HiC_S2_1p_10min_lowU_100kbp_corrected_pca1.png
```

29. What does PC1 represent?

7 Calling topologically associating domains (TADs)

TAD calling works in several steps:

i. The `hicFindTADs` tool computes a *TAD-separation score* for each genomic bin.

ii. The bins for which the TAD-separation score has a local minimum are compared to the neighboring bins and assigned a P-value.

iii. A cutoff is applied to select the bins more likely to constitute TAD boundaries.

30. We will call TADs with `hicFindTADs` for the higher resolution (10kpb) matrix we started with.

31. Generate the histogram for the 10kbp contact map and correct as before in 23 and 24. Name the corrected matrix `HiC_S2_1p_10min_lowU_10kbp_corrected.h5`

32. Call TADs with `hicFindTADs` for the higher resolution (10kpb) matrix we started with

```
hicFindTADs --matrix ./hicMatrix/HiC_S2_1p_10min_lowU_10kbp_corrected.h5 \
        --minDepth 30000 \
        --maxDepth 100000 \
        --step 10000 \
        --correctForMultipleTesting fdr \
        --thresholdComparisons 0.05
        --delta 0.001 \
        --outPrefix ./hicMatrix/HiC_S2_1p_10min_lowU_10kbp_corrected_TADs \
        --numberOfProcessors 2
```

- `--minDepth 30000` sets the minimum window length (in bp) considered to the left and right of each Hi-C bin. This number should be at least 3 times as large as the bin size of the Hi-C matrix.
- `--maxDepth 100000` sets the maximum window length considered to the left and right of the cut point in bp. This number should around 6-10 times as large as the bin size of the Hi-C matrix.
- `--step 10000` is the step size when moving from `minDepth` to `maxDepth`.
- `--thresholdComparisons 0.05` specifies the P-value threshold for the Bonferroni correction/q-value for the FDR. The probability of a local minima to be a boundary is estimated by comparing the distribution (Wilcoxon rank-sum) of the z-scores between the left and right regions (*diamond*) at the local minimum with the matrix z-scores for a diamond at `minDepth` to the left and a diamond `minDepth` to the right.

- `--delta 0.001` determines the minimum threshold for the difference between the TAD-separation score of a putative boundary and the mean TAD-separation score of the neighboring bins.
- `--outPrefix` is the prefix of the output files.

`hicFindTADs` produces several output files, including the predicted TAD boundaries, domains, and scores:

- `<prefix>_tad_separation.bm` is a "bedgraph matrix file", which can serve as input for `hicPlotTADs`. The format of the file is chrom start end TAD-sep1 TAD-sep2 TAD-sep3, ... Each of the TAD-separation scores in the file corresponds to a different window length, from `minDepth` to `maxDepth`.
- `<prefix>_zscore_matrix.h5` contains the the z-score matrix used for the computation of the TAD-separation score.
- `<prefix>_boundaries.bed` contains the positions of boundaries. The genomic coordinates in this file correspond to the resolution used. Thus, for Hi-C bins of $10,000$ bp the boundary position is $10,000$ bp long. For restriction fragment matrices the boundary position varies depending on the fragment length at the boundary.
- `<prefix>_domains.bed` contains the TADs positions. This is a non-overlapping set of genomic positions.
- `<prefix>_boundaries.gff` is similar to the boundaries .bed file, but includes additional information (P-value, `delta`).
- `<prefix>_score.bedgraph` file contains the TAD-separation score measured at each Hi-C bin coordinate. Is useful to visualize in a genome browser. The P-value and `delta` settings are saved as part of the name.

Lastly we will visualize the resulting TADs. Look again at the plots we generated in 28 and decide on a region about 3 to 5 Mbp in length to plot.

33. Have a look at the `hicPlotTADs` manual page. You can edit the included `tracks.ini` file to you specific paths and file names. Visualize the TADs of a region of your choice.

```
hicPlotTADs --tracks tracks.ini -o <output_file_name> \
        --region <chr_region> --height 6
```

The 3D architecture of genomes of all species results in fascinating and beautiful patterns at various scales. Fig.7 depicts the HiC contact map of a human K562 cell line on chromosome 15. The data used to generate this map comprised over a billion reads.

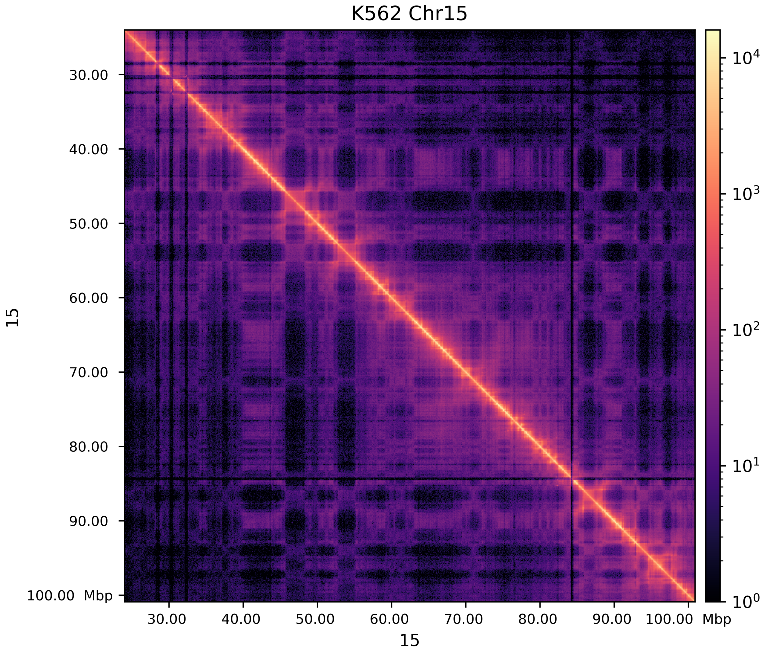

Figure 1: HiC contact map of human Chr15 derived from K562 cell line at 100kbp resolution

8 Version Information

8.1 FastQC

```
FastQC v0.11.5
```

8.2 Botwie2

```
/usr/local/bin/bowtie2-align-s version 2.4.1
64-bit
Built on
Fri 03 Jul 2020 09:59:17 AM UTC
Compiler: gcc version 9.3.0 (Ubuntu 9.3.0-10ubuntu2)
Options: -O3 -msse2 -funroll-loops -g3 -DPOPCNT_CAPABILITY -DWITH_TBB -std=c++11 -DNO_SPINLOCK -
Sizeof {int, long, long long, void*, size_t, off_t}: {4, 8, 8, 8, 8, 8}
```

8.3 samtools

```
Program: samtools (Tools for alignments in the SAM format)
Version: 1.13 (using htslib 1.13)
```

8.4 HiCExplorer

```
hicBuildMatrix 3.7.2
hicPlotMatrix 3.7.2
```

8.5 IGV

```
IGV 2.3
```

9 Acknowledgements

This tutorial is based on the Galaxy Training Materials prepared by Joachim Wolff, Fidel Ramirez, Vivek Bhardwaj, Ekaterina Polkh, 2020 Hi-C analysis of Drosophila melanogaster cells using HiCExplorer (`https://training.galaxyproject.org/archive/2019-02-07/topics/epigenetics/tutorials/hicexplorer/tutorial.html`, accessed on December 17, 2021).