**Pipeline for Identification of A/B Compartments**

This pipeline is based on the protocol outlined by Miura et al. in their book1. It utilizes HiCExplorer commands as detailed in the documentation2 and generates A/B compartments starting from a raw .cool matrix. While input for pipeline is a raw .cool matrix, but you can start from a normalized .cool matrix and bypass the step 1 below.

1. **Normalize .cool file:**
   1. **(optional) Normalize to equal level of read coverage or value ranges:** If you want to compare different Hi-C interaction matrices (samples/replicates), the matrices need to be normalized to equal level of read coverage or value ranges. This can be achieved with ‘hicNormalize’3 command as follows. Please refer to [hicNormalize documentation](https://hicexplorer.readthedocs.io/en/latest/content/tools/hicNormalize.html) for further details.

hicNormalize -m matrix.cool --normalize smallest -o normalized\_matrix.cool

* 1. **Correct HiC matrix using Iterative Correction (ICE):** The Hi-C matrix has to be corrected to remove GC, open chromatin biases and, most importantly, to normalize the number of restriction sites per bin4. However, for this method to work correctly, bins with zero reads assigned to them should be removed as they cannot be corrected. Also, bins with the low number of reads should be removed. Bins with an extremely high number of reads can also be removed from the correction as they may represent copy number variations. To aid in the identification of bins (threshold values) with low and high read coverage, the histogram (diagnostic plot) of the number of reads can be plotted together with the Median Absolute Deviation (MAD) using the ‘hicCorrectMatrix’5 command as shown below.

hicCorrectMatrix diagnostic\_plot –m matrix.cool -o plot\_file.png --chromosomes chr

Once the thresholds have been decided, the matrix can be corrected as follows.

hicCorrectMatrix correct -m matrix.cool --correctionMethod ICE –chromosomes chr\_list --iterNum 5000 -o corrected.cool --filterThreshold -1.5 5

1. **Generate observed-over-expected matrix:**

To compute the A/B compartments the matrix needs to be transformed to an observed-over-expected matrix6 using. ‘hicTransform’7 command.

hicTransform -m corrected.cool --method obs\_exp -o out.cool --chromosomes chr\_list

1. **Calculate Pearson correlation matrix:**

Next, we transform the observed-over-expected matrix into a Pearson correlation matrix by calculating the Pearson correlation values between all pairs of rows and columns as follows.

hicTransform -m obs\_exp.cool --method pearson -o pearson.cool --chromosomes chr\_list

1. **Compute first principal component of the Pearson correlation matrix:**

The first principal component (PC) of the Pearson correlation matrix has been shown to correlate with A/B compartments6,8. The first PC track (in bedgraph format) is computed using ‘hicPCA’9 command from HiCExplorer as shown below.

hicPCA -m pearson.cool -o pca1.bedgraph --format bedgraph -we 1 --chromosomes chr\_list

1. **Calculate gene density:**

Unfortunately, PCA does not provide information about the relationship between the sign of the first principal component (PC) and the A/B compartments. Therefore, it is recommended to use external information, such as histone marks, to identify A/B compartments. One of the straightforward methods to determine A/B compartments is by calculating gene density. The A compartment (active) tends to have a high gene density. It is estimated that regions with high GC content have a higher relative gene density compared to regions with lower GC content10,11. The following Python script calculates the GC content (i.e., gene density) track for given chromosomes. This script requires the TADbit package12. Please provide appropriate input arguments at the beginning of the script (lines 9-11).

1. # conda env = tadbit
3. **import** numpy as np
4. **import** pandas as pd
5. **from** pytadbit.parsers.genome\_parser **import** get\_gc\_content, parse\_fasta
6. **import** os.path
7. **from** subprocess **import** call
9. hic\_pro\_abs\_bed\_file**=**""   # full path to abs.bed file generated by HiC-Pro
10. genome\_fasta\_file**=**""      # full path to genome fasta file
11. out\_bedgraph\_file**=**""      # full path to output bedgraph file
13. df **=** pd.read\_csv(hic\_pro\_abs\_bed\_file,sep**=**"\t",header**=**None)
14. df\_chr**=**df[df[0]**==**CHR]
15. print(df\_chr)
17. rich\_in\_A **=** get\_gc\_content(parse\_fasta(genome\_fasta\_file),chromosomes**=**[CHR],resolution**=**RESOLUTION, n\_cpus**=**4)
18. print(rich\_in\_A)
19. print(len(rich\_in\_A))
21. df\_chr[4]**=**rich\_in\_A
22. df\_chr**=**df\_chr.dropna(subset**=**4)
23. df\_out**=**df\_chr[[0,1,2,4]]
24. print(df\_out)
25. df\_out.to\_csv(out\_bedgraph\_file, header**=**False,index**=**False,sep**=**"\t")
26. **Plot A/B compartments:**

A/B compartments can be visualized using ‘pyGenomeTracks’13 package. To begin, you need to create a track file that includes the first principal component (PC), gene density, and the contact matrix. Below is an example of how to structure your track file, along with the command to plot the A/B compartments using this file.

Example track.ini file:

1. [x-axis]
2. where = top
4. [spacer]
6. [pca1 bedgraph]
7. file = pca1\_Seq021\_MicroC\_merged\_rep\_chr1\_500000.bedgraph
8. color = blue
9. height = 5
10. transform = no
11. negative\_color = red
12. height = 2
13. title = PCA 1
15. [rich\_in\_A.bedgraph]
16. file = rich\_in\_A\_Seq021\_MicroC\_merged\_rep\_chr1\_500000.bedgraph
17. color = orange
18. type = line:2
19. height = 2
20. title = GC content
21. [spacer]
23. [hic matrix]
24. file = obs\_exp\_Seq021\_MicroC\_merged\_rep\_chr1\_500000.cool
25. depth =  250\_000\_000
26. file\_type = hic\_matrix\_square
27. colormap = coolwarm
28. transform = log
29. min\_value = -2
30. max\_value = 2
31. show\_masked\_bins = true
32. title = log(O/E)

Command:

pyGenomeTracks **--**tracks tracks.ini **--**region CHR:star**-**end **--**outFileName out.png

**References:**

1. Miura, H., Poonperm, R., Takahashi, S. & Hiratani, I. Practical Analysis of Hi-C Data: Generating A/B Compartment Profiles. in *X-Chromosome Inactivation: Methods and Protocols* (ed. Sado, T.) 221–245 (Springer, New York, NY, 2018). doi:10.1007/978-1-4939-8766-5\_16.

2. HiCExplorer A / B compartment analysis. https://hicexplorer.readthedocs.io/en/latest/content/example\_usage.html#id12.

3. HiCExplorer hicNormalize. https://hicexplorer.readthedocs.io/en/latest/content/tools/hicNormalize.html.

4. HiCExplorer Correction of Hi-C matrix. https://hicexplorer.readthedocs.io/en/latest/content/example\_usage.html#correction-of-hi-c-matrix.

5. HiCExplorer hicCorrectMatrix. https://hicexplorer.readthedocs.io/en/latest/content/tools/hicCorrectMatrix.html.

6. Lieberman-Aiden, E. *et al.* Comprehensive Mapping of Long-Range Interactions Reveals Folding Principles of the Human Genome. *Science* **326**, 289–293 (2009).

7. HiCExplorer hicTransform. https://hicexplorer.readthedocs.io/en/latest/content/tools/hicTransform.html.

8. Fortin, J.-P. & Hansen, K. D. Reconstructing A/B compartments as revealed by Hi-C using long-range correlations in epigenetic data. *Genome Biol.* **16**, 180 (2015).

9. HiCExplorer hicPCA. https://hicexplorer.readthedocs.io/en/latest/content/tools/hicPCA.html.

10. MacKinnon, L. Characteristics of the Human Genome.

11. GC-content. *Wikipedia* (2024).

12. TADbit. https://github.com/3DGenomes/TADbit.

13. pyGenomeTracks. https://github.com/deeptools/pyGenomeTracks/.