**Pipeline for generating contact matrix from FASTQ files using HiC-Pro**1,2

1. **MERGE TECHNICAL REPLICATES OF INDIVIDUAL BIOLOGICAL REPLICATES**
2. Arrange the FASTQ files according to the following directory structure. HiC-Pro considers all readsr within one input folder as one sample3.

+ PATH\_TO\_MY\_DATA

+ sample1 (bio\_rep1)

++ file1\_R1.fastq.gz

(tech\_rep\_1)

(tech\_rep\_1)

++ file1\_R2.fastq.gz

++ file2\_R1.fastq.gz

(tech\_rep\_2)

++ file2\_R2.fastq.gz

++ ...

+ sample2 (bio\_rep2)

++ file1\_R1.fastq.gz

++ file1\_R2.fastq.gz

...

1. Generate annotation files.
   * **chromosomes sizes file:** Two-column tab-separated text file containing chromosome names and sizes.

e.g.: <https://hgdownload.soe.ucsc.edu/goldenPath/hg38/bigZips/hg38.chrom.sizes>

* + **The bowtie2 indexes:** See the [bowtie2 manual page](https://bowtie-bio.sourceforge.net/bowtie2/index.shtml) for details about how to create such indexes.
  + **(HiC only) A BED file of the restriction fragments after digestion:** This file depends on both the restriction enzyme and the reference genome. See the [HiC-Pro annotation](https://github.com/nservant/HiC-Pro/tree/master/annotation) for details about how to generate this file.

1. Setup the configuration file.

Copy and edit the 'config-hicpro.txt' file in your local directory. Modify the options in the configuration file as required. Below are some important options to consider; for more detailed explanations of these options, refer to the [HiC-Pro manual](https://github.com/nservant/HiC-Pro/blob/master/doc/MANUAL.md).

* PAIR1\_EXT = Keyword for first mate detection. Default: \_R1
* PAIR2\_EXT = Keyword for second mate detection. Default: \_R2
* REFERENCE\_GENOME = Reference genome prefix used for genome indexes. Default: hg19
* GENOME\_SIZE = Full path to chromosome size file.
* MIN\_CIS\_DIST4 = [Important option for Micro-C data] Filter short range contact below the specified distance. Example: 1000
* BIN\_SIZE = Resolution of contact maps to generate (space separated). Default: 20000 40000 150000 500000 1000000
* (Hi-C only) GENOME\_FRAGMENT = Full path to BED file with restriction fragments.
* (Hi-C only) LIGATION\_SITE = Ligation site sequence(s) used for reads trimming. Depends on the fill in strategy. Note that multiple ligation sites can be specified (comma separated). Example: AAGCTAGCTT
* (Hi-C only) MIN\_FRAG\_SIZE = Maximum size of restriction fragments to consider for the Hi-C processing. Example: 100
* (Hi-C only) MAX\_FRAG\_SIZE = Maximum size of restriction fragments to consider for the Hi-C processing. Example: 100000
* (Hi-C only) MIN\_INSERT\_SIZE = Minimum sequenced insert size. Shorter 3C products are discarded. Example: 100
* (Hi-C only) MAX\_INSERT\_SIZE = Maximum sequenced insert size. Larger 3C products are discarded. Example: 600

1. Run HiC-pro.
2. MY\_INSTALL\_PATH/bin/HiC-Pro -i FULL\_PATH\_TO\_DATA\_FOLDER -o FULL\_PATH\_TO\_OUTPUTS -c MY\_LOCAL\_CONFIG\_FILE
3. **MERGE BIOLOGICAL REPLICATES**

To merge biological replicates after merging technical replicates, we can rerun HiC-Pro starting from the validPairs files generated during the previous analysis. There's no need to start from the FASTQ files again3.

1. Arrange validPairs files according to the following directory structure.

++ data

++++ sample\_1

(bio\_rep\_1)

(bio\_rep\_2)

++++++++ validPairs tech\_replicate1

++++++++ validPairs tech\_replicate2

++++++++ validPairs tech\_replicate1

++++++++ validPairs tech\_replicate2

….

1. Rerun Hic-Pro in stepwise mode.
2. MY\_INSTALL\_PATH/bin/HiC-Pro -i FULL\_PATH\_TO\_VALIDPAIRS\_DATA -o FULL\_PATH\_TO\_OUTPUTS -c MY\_LOCAL\_CONFIG\_FILE -s merge\_persample -s build\_contact\_maps -s ice\_norm
3. **GENERATE .COOL FILES**

HiC-Pro outputs contact matrices in .matrix format. However, .cool files are generally accepted by most HiC analysis software and are also easier to handle than other formats. Therefore, I convert .matrix files to .cool files using the following 'hicpro2higlass.sh' script that comes with HiC-Pro.

1. HICPRO\_PATH/bin/utils/hicpro2higlass.sh -i MATRIX\_FILE -r RESOLUTION -c CHROMSIZES\_FILE -o OUTPUT\_PATH -p NUM\_PROC

For more details on input arguments of ‘hicpro2higlass.sh’ script please refer - <https://github.com/nservant/HiC-Pro/blob/master/bin/utils/hicpro2higlass.sh>

1. **NORMALIZE COOL FILES**

Although HiC-Pro can generate normalized .cool files, I prefer to normalize the raw .cool files separately. I use the following Python script for normalizing the .cool files. This script uses Iterative correction5,6 matrix balancing method from cooler package7. The full path to the .cool file should be provided at line number 11 before running.

|  |
| --- |
| 1. **import** numpy as np 2. **import** matplotlib.pyplot as plt 3. **import** pandas as pd 4. **import** cooler 5. **import** multiprocessing as mp 6. **import** cooltools 7. **from** tqdm **import** tqdm 8. **from** subprocess **import** call 9. **import** sys 10. cool\_file\_path**=”FULL\_PATH\_TO\_COOL\_FILE**” 11. print(cool\_file\_path)  14. clr **=** cooler.Cooler(cool\_file\_path) 15. resolution **=** clr.binsize 16. print(f"Resolution = {resolution}") 17. print(clr.chromnames) 18. print(clr.bins()[:10]) 19. print(clr.pixels()[:10]) 21. clr\_columns **=** clr.bins()[:10].columns 23. #from https://www.biostars.org/p/9561371/ 24. **if** "weight" **not** **in** clr\_columns: 25. weights **=** cooler.balance\_cooler(clr) 26. print(weights) 27. print(weights[0].shape) 28. with clr.open('r+') as f: 29. f["bins"].create\_dataset("weight", data**=**weights[0], compression**=**"gzip", compression\_opts**=**6) 31. print(clr.bins()[:10]) 32. print(clr.pixels()[:10]) |

**References and useful resources:**

1. Servant, N. *et al.* HiC-Pro: an optimized and flexible pipeline for Hi-C data processing. *Genome Biol.* **16**, 259 (2015).

2. Servant, N. HiC-Pro Github. https://github.com/nservant/HiC-Pro (2024).

3. HiC-Pro github issue #121 How to pool several datasets. https://github.com/nservant/HiC-Pro/issues/121.

4. HiC-Pro github issue #267 Identify unligated internal reads in Micro-C dataset. https://github.com/nservant/HiC-Pro/issues/267.

5. Imakaev, M. *et al.* Iterative correction of Hi-C data reveals hallmarks of chromosome organization. *Nat. Methods* **9**, 999–1003 (2012).

6. Hsieh, T.-H. S. *et al.* Enhancer–promoter interactions and transcription are largely maintained upon acute loss of CTCF, cohesin, WAPL or YY1. *Nat. Genet.* **54**, 1919–1932 (2022).

7. API Reference — cooler.balance\_cooler. https://cooler.readthedocs.io/en/latest/api.html#cooler.balance\_cooler.

HiC-Pro manual: <https://github.com/nservant/HiC-Pro/blob/master/doc/MANUAL.md>