NGS analysis for gene regulation and epigenomics

Jacques Serizay, Cyril Matthey-Doret

January 11, 2020

Day 1

Introduction to Hi-C

Regulatory function of chromatin

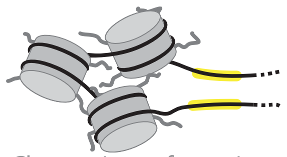
- Chromatin made up of many DNA-bound proteins
- Dynamic interactions
- Chromatin state regulates gene expression

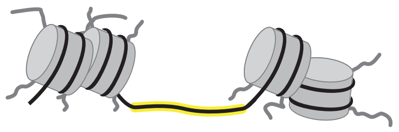
A picture of the regulatory landscape

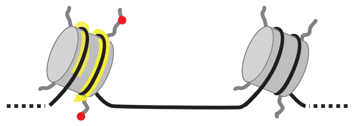
- Gene expression tells us what genes are affected during e.g. differentiation, disease…
- Epigenomics can tell us how / why they are activated

Types of chromatin information

We will look at 3 scales of chromatin organization during this workshop:


Chromatin conformation capture
(3-C, Hi-C and Capture-C)

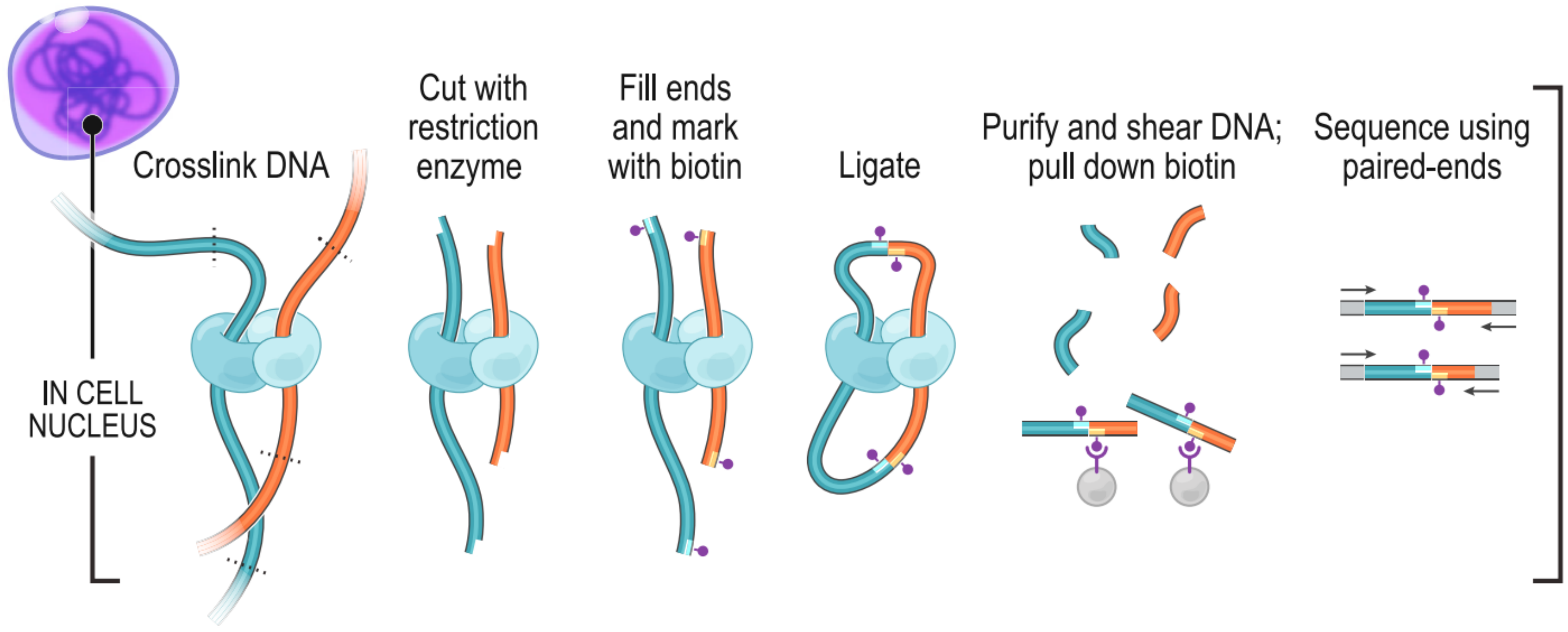

Assay for transposase accessible
chromatin (ATAC-Seq)


ChIP-Seq of methylated histones
(Histone methylation)

Adapted from Illumina

Capturing chromosome conformation

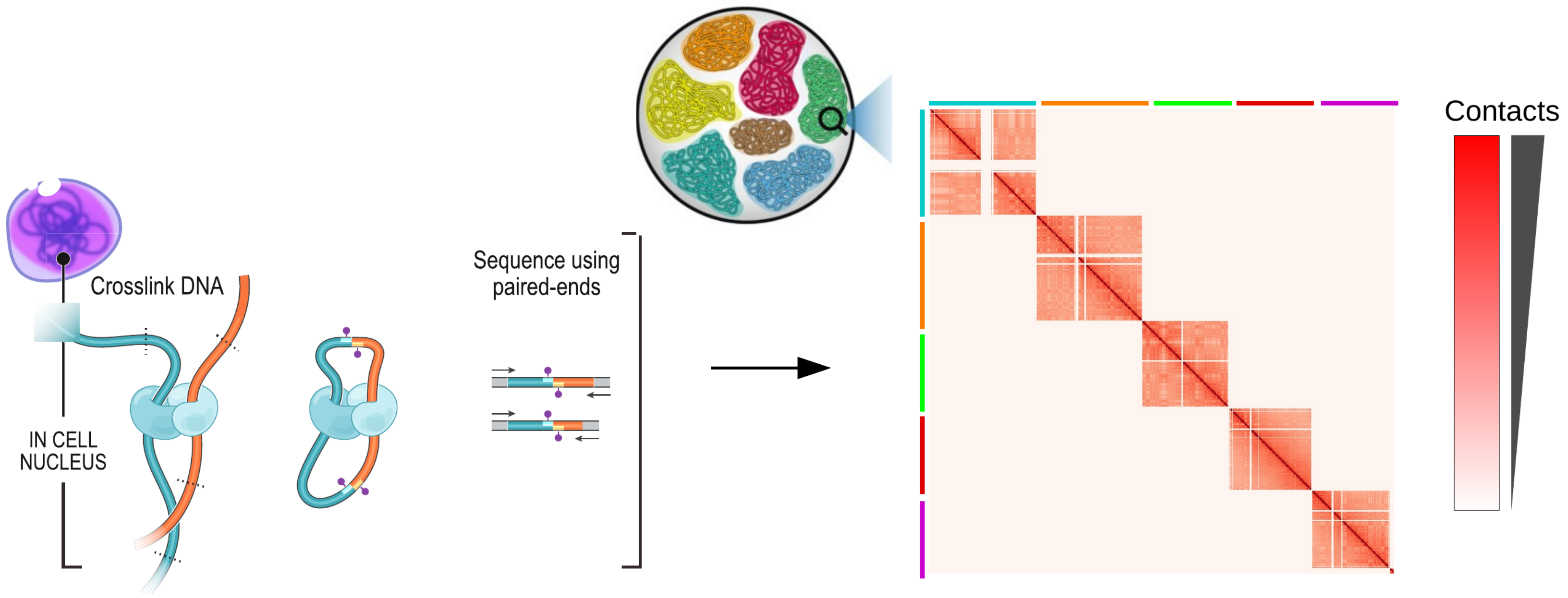
- 3C methods capture the 3D structure of the genome
- Originally relied on qPCR, Hi-C is the NGS variant



IN CELL NUCLEUS
Crosslink DNA
Cut with restriction enzyme
Fill ends and mark with biotin
Ligate
Purify and shear DNA; pull down biotin
Sequence using paired-ends

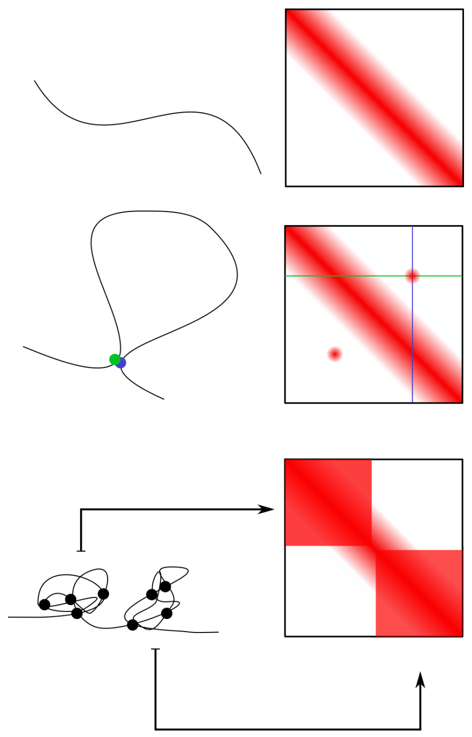
Hi-C experimental process

Interpreting chromosome contact maps



Contact frequency reflects spatial organization

Interpreting chromosome contact maps



- Diagonal gradient due to polymer behavior
- Various patterns correspond to 3D structures

Derivative methods

3C-based methods (digestion + religation):

- MicroC: DNAse instead of restriction enzyme
- Promoter capture: Hi-C with capture probes on promoters

Others:

- GAM: Cryoslicing of nuclei followed by sequencing
- SPRITE: Serial dilution and barcoding

From reads to contact maps

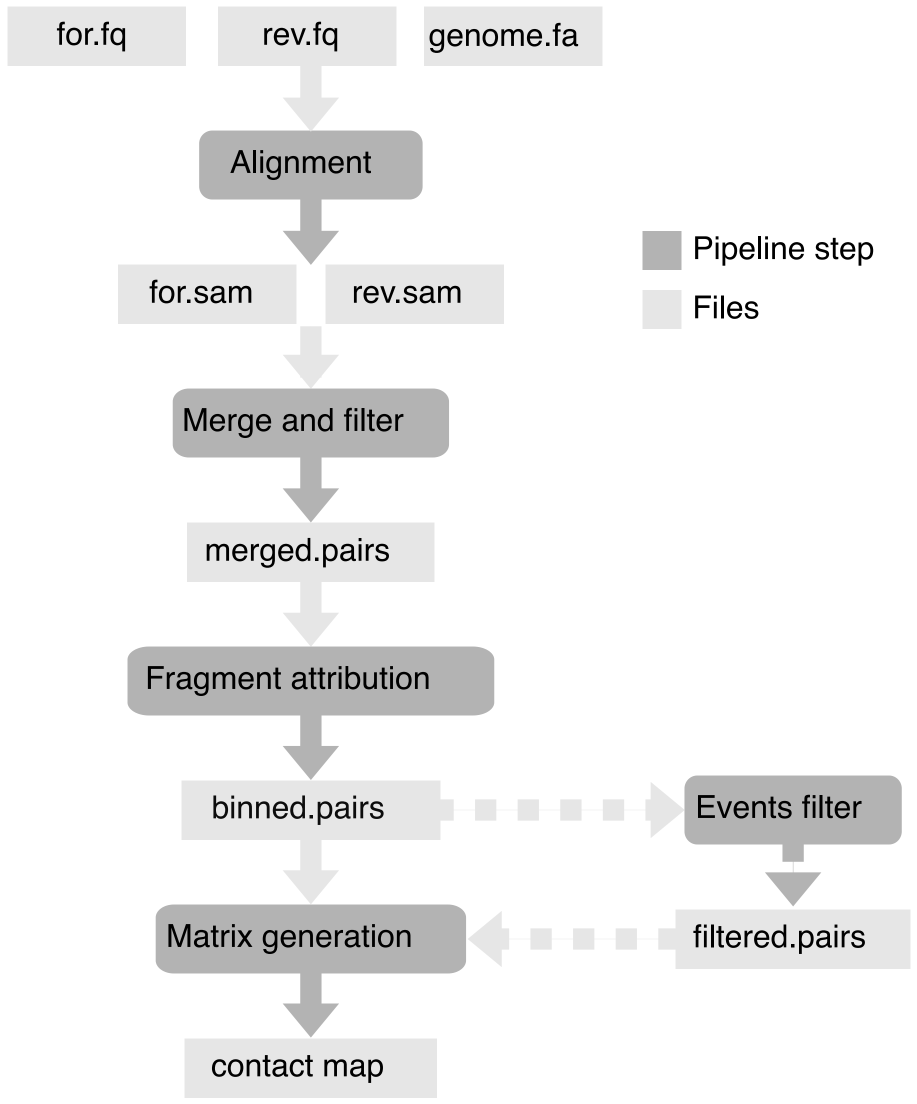
Many “end-to-end” pipelines available:

- **nf-core/hic**
- Hi-C pro
- FanC
- hicstuff
- Juicer
- …

Important to understand individual steps (biases, custom analyses, …)

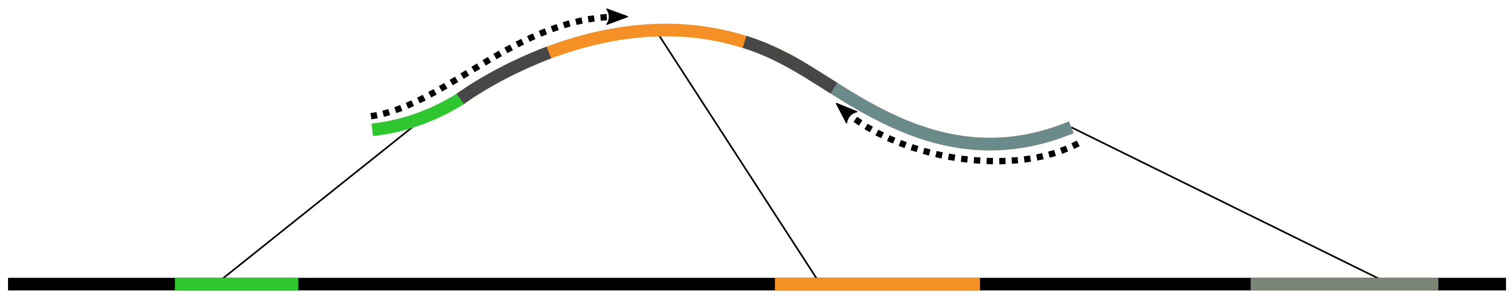
Hi-C processing: Overview

General steps common to all Hi-C pipelines



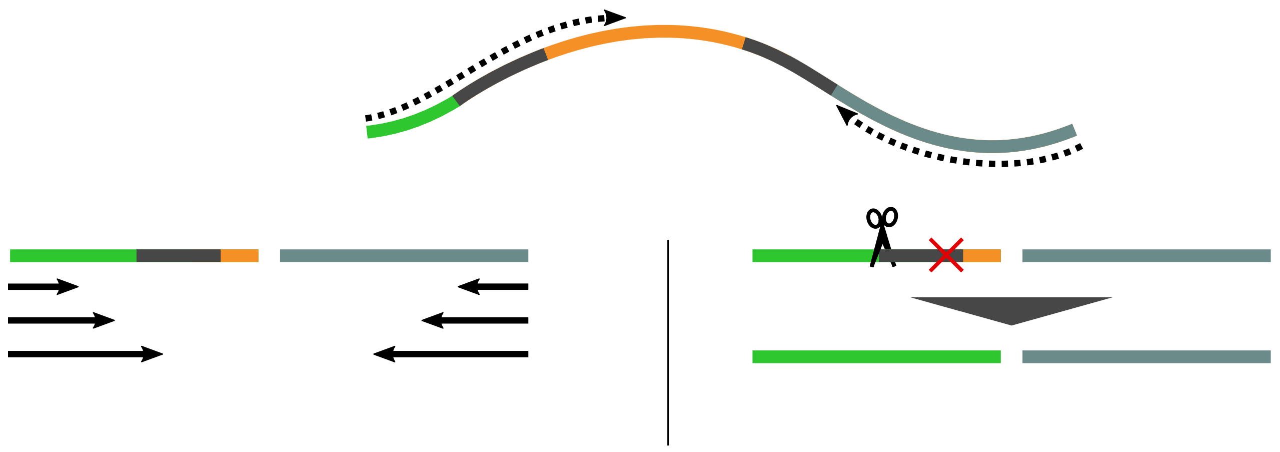
Hi-C processing: Read alignment

- Standard short read mapping
- Separate single-end alignment for forward and reverse
- Longer reads relative to fragment length: More chimeric reads



Hi-C processing: Read alignment

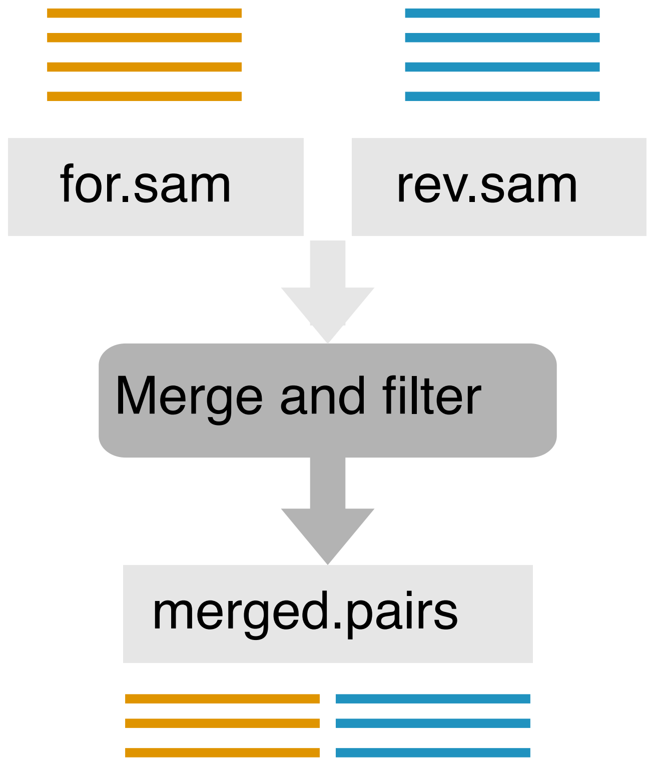
- Solutions generally built in pipelines
  - iterative alignment
  - in-silico fastq digestion
- Mostly important with longer reads



```
truncate read
while not mapped:
  extend read
  mapped = align read
```

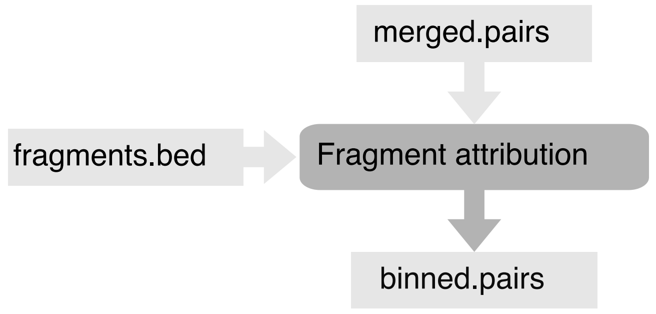
Hi-C processing: Merging BAMs

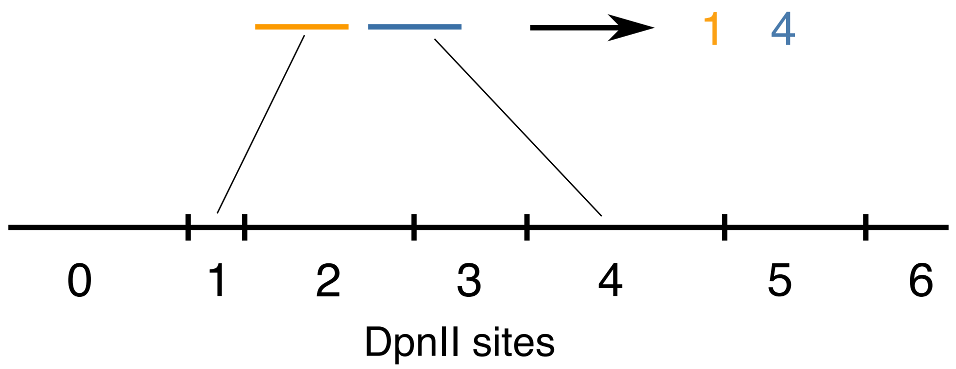
- Merge forward and reverse reads by name
- Filter out ambiguous alignment (low mapQ)
- 1 pair of reads = 1 contact
- The pairs format is commonly used



Hi-C processing: Fragment attribution

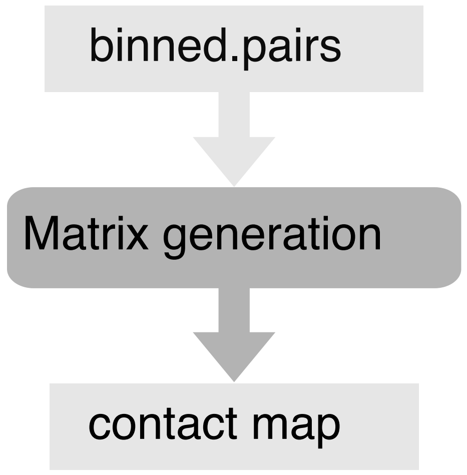
- Assign contacts to discrete segments (bins)
- Usually regular intervals of e.g. 10kb





Hi-C processing: The Matrix

- Sum contact events for each combination of bins
- Most combinations have 0 contacts: “sparse data”
- Storing NxN matrix with mostly 0 is impractical + Use sparse format instead



	0	1	2	3
0				
1		4		
2				
3			2	

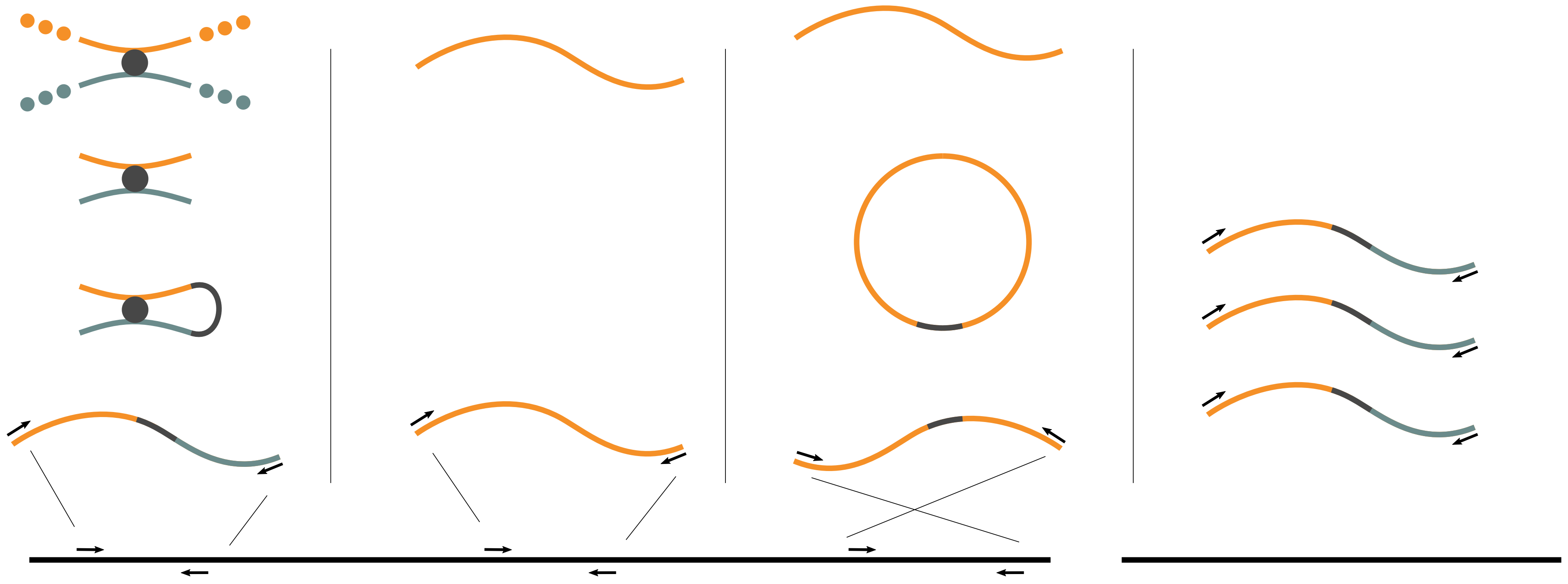
0	0	0	0
0	4	0	0
0	0	0	0
0	0	2	0

x	y	v
1	1	4
3	2	2

Hi-C processing: Additional filters

Hi-C specific filters to remove uninformative events

- Self-religating fragments
- Undigested fragments
- Duplicates filter

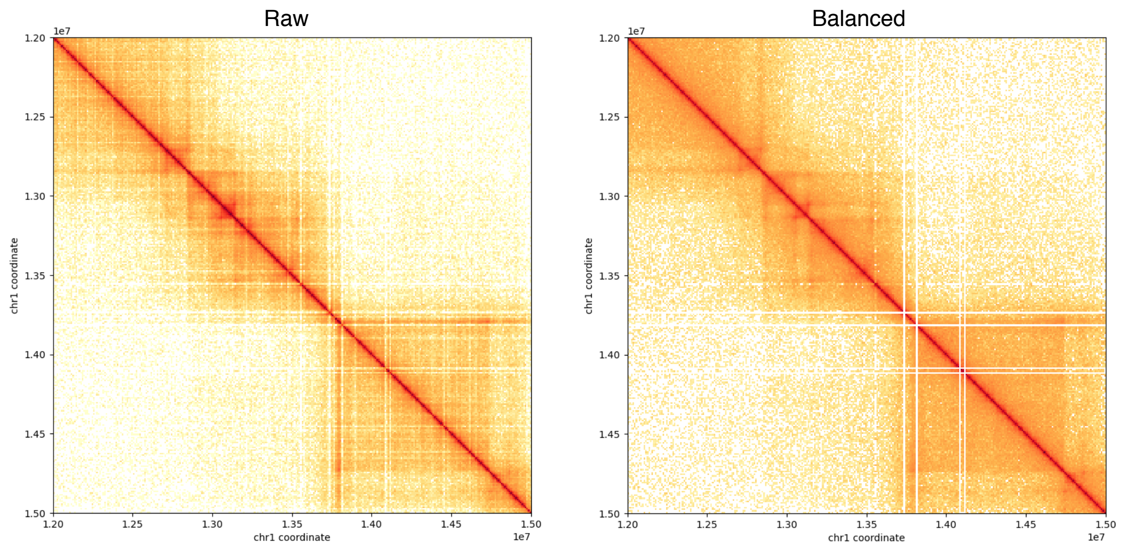


Hi-C processing: Balancing

Hi-C is susceptible to many biases affecting local coverage such as:

- GC content
- Chromatin accessibility
- Restriction site density

Matrix balancing is a normalization to reduce the impact of those biases.



Hi-C processing: Balancing

Assumption: All bins (regions) have the same contact probability.

- Divide each pixel by row x col iteratively.
- Predefined number of iterations or until convergence.

```
# Simplified ICE procedure
m, n = mat.shape
for _ in range(100):
  for x in range(m):
    row_mean = mat[x, :].mean()
    for y in range(n):
      col_mean = mat[:, y].mean()
      mat[x, y] = mat[x, y] / (row_mean * col_mean)
```

- Generate your own contact map from the reads, normalize and visualize it

Exercises