UNIVERSITÀ
DI TRENTO
Department of
Cellular, Computational and Integrative Biology - CIBIO

Master's Thesis Dissertation
October 18th 2023

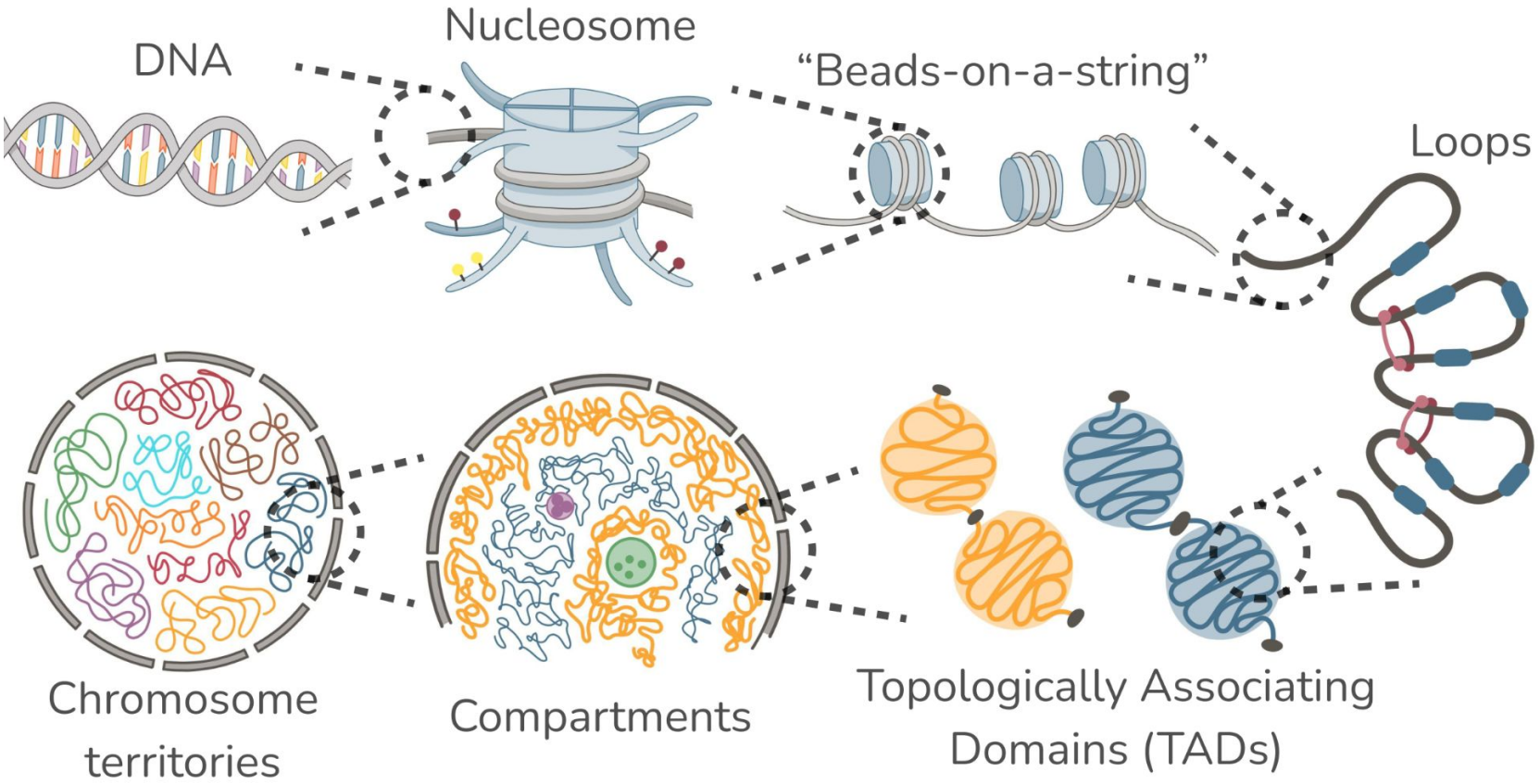
Identification of Chromatin Organization Backbone Through Network Sparsification

Supervisor:
Alessio Zippo

Co-Supervisor:
Leonardo Morelli

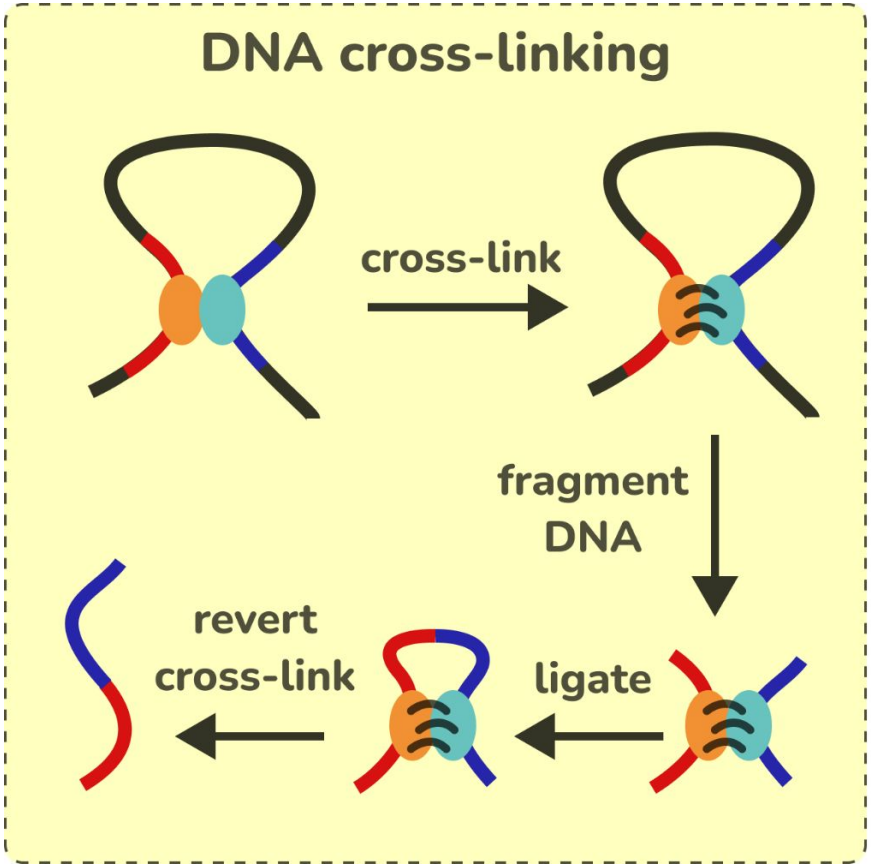
Graduand:
Stefano Cretti

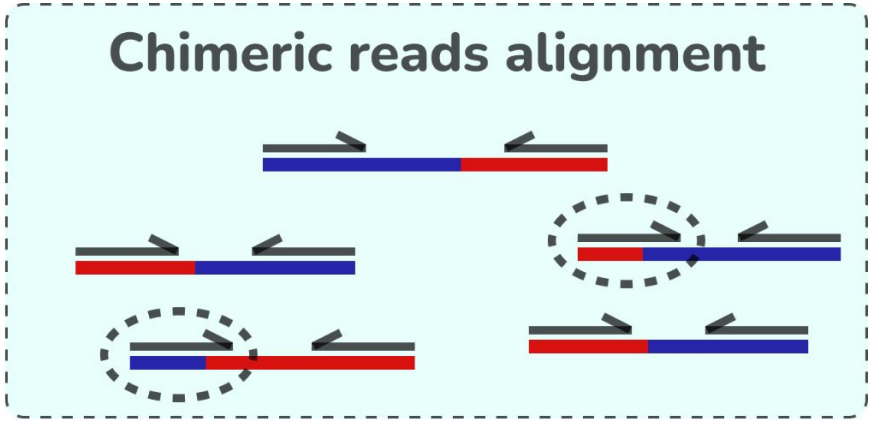
Chromatin Organization

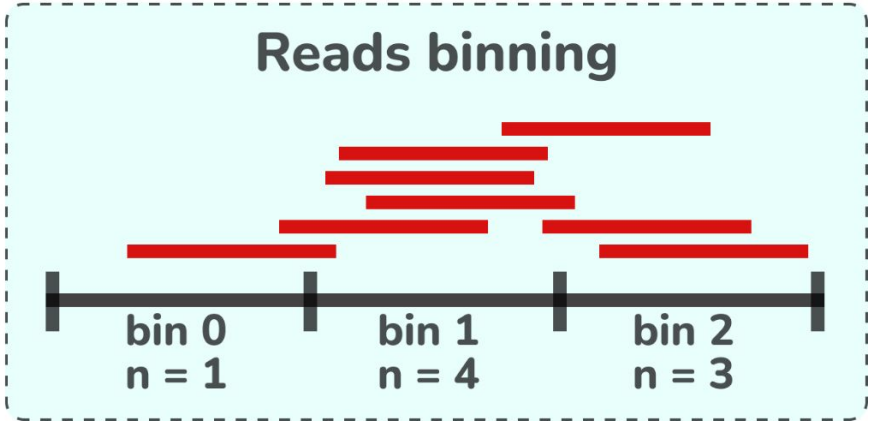


Structures drawn by Annarita Zanon

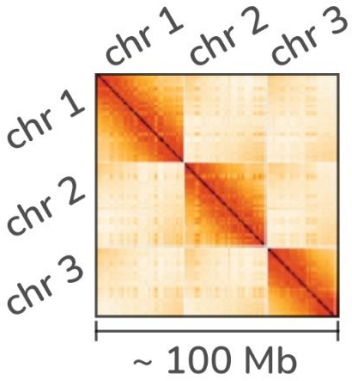
Hi-C Protocol



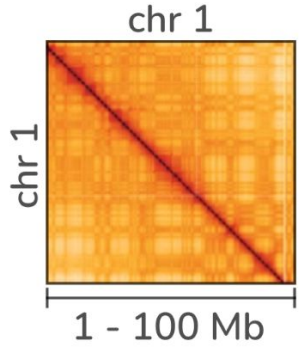




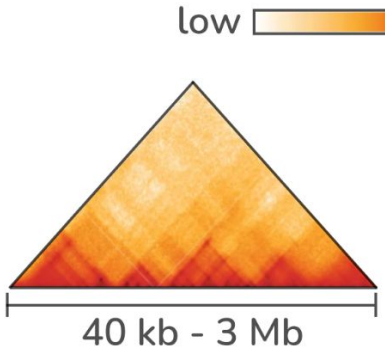
Contact Matrices



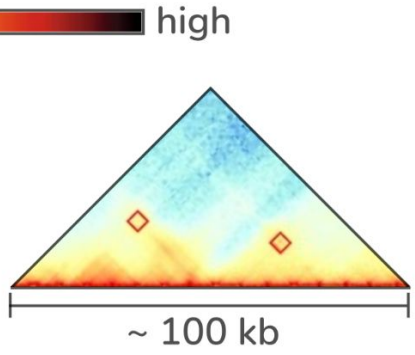
Chrom. Territories



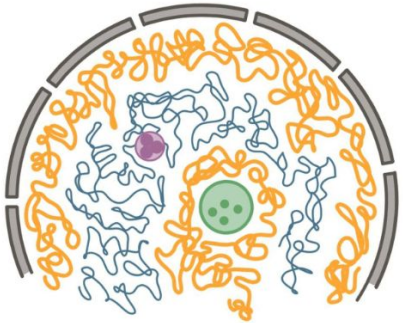
Compartments

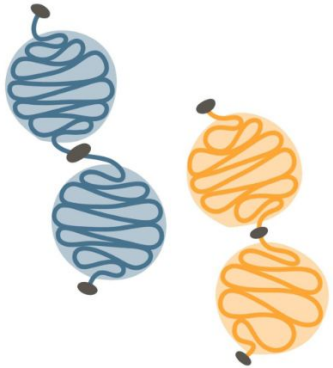


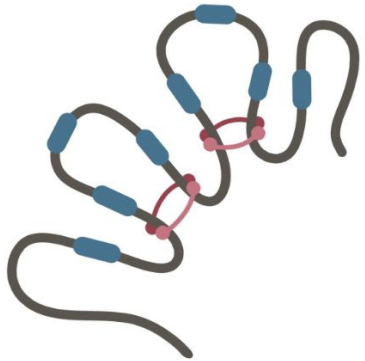
TADs



Loops

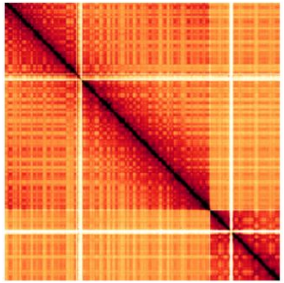






Structures drawn by Annarita Zanon, matrices from Wolff et al., 2022 and from Kempfer and Pombo, 2019

Aims



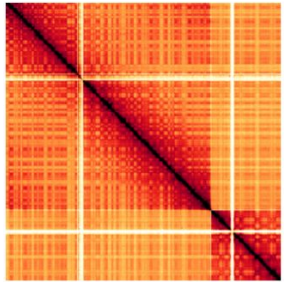
✓

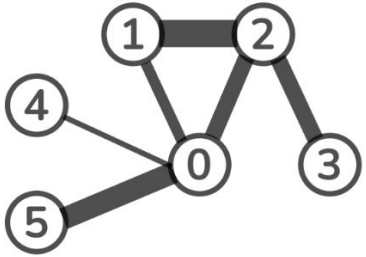
- visualization
- row/column operations

✗

- very big and sparse
- few tools

Aims





✓

- visualization
- row/column operations

+

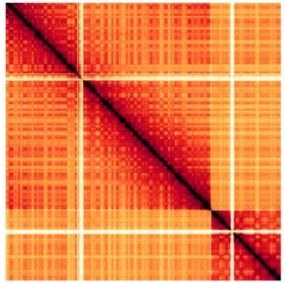
- complex relationships
- can integrate information

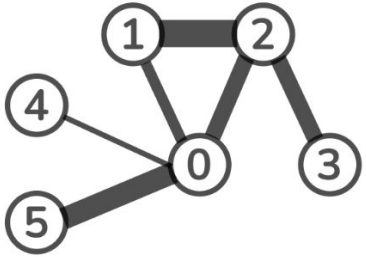
✗

- very big and sparse
- few tools

- difficult to apply to Hi-C data

Aims





Problem to solve

✓
- visualization
- row/column operations

+

- complex relationships
- can integrate information

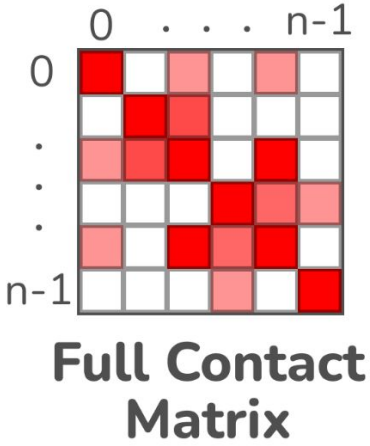
=

Need for easy and accessible Hi-C network analysis to obtain new insights

✗
- very big and sparse
- few tools

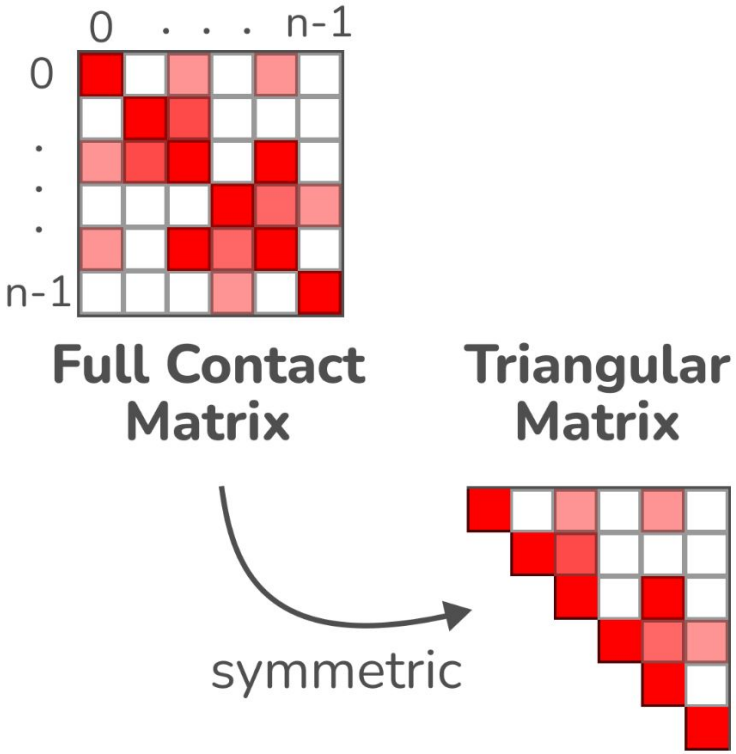
- difficult to apply to Hi-C data

From Matrix To Network

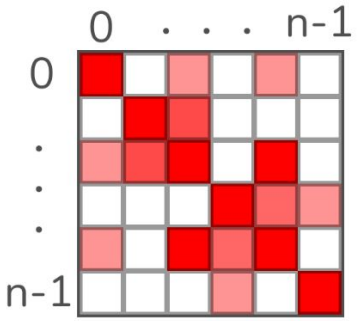


Full Contact Matrix

From Matrix To Network



From Matrix To Network



Full Contact Matrix

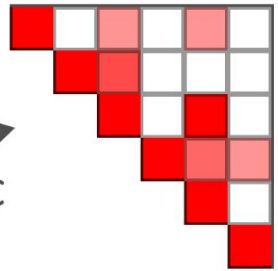
sparse

Triangular Matrix

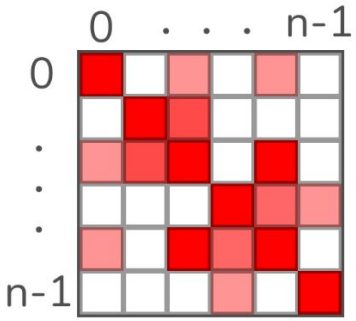
row id	col id	count
0	0	12
0	2	6
0	4	7
1	1	16
...	...	...

ijv table

symmetric



From Matrix To Network



Full Contact Matrix

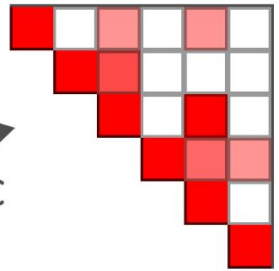
sparse

row id	col id	count
0	0	12
0	2	6
0	4	7
1	1	16
...	...	...

ijv table

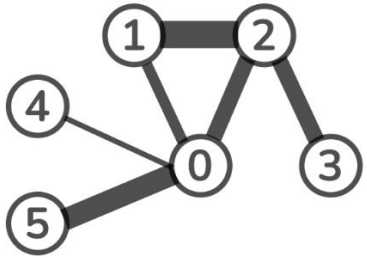
Triangular Matrix

symmetric

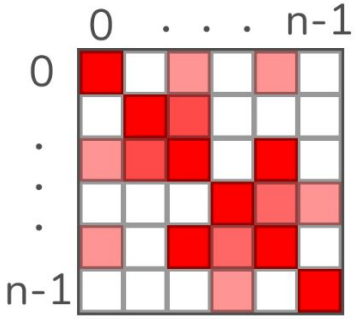


edge list

Weighted Undirected Graph



From Matrix To Network



Full Contact Matrix

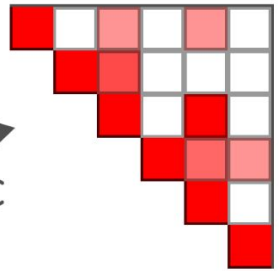
sparse

row id	col id	count
0	0	12
0	2	6
0	4	7
1	1	16
...	...	...

ijv table

Triangular Matrix

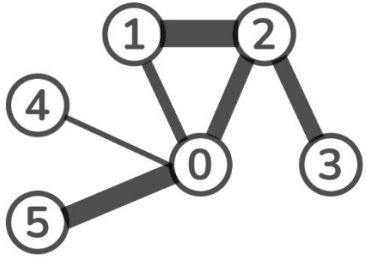
symmetric



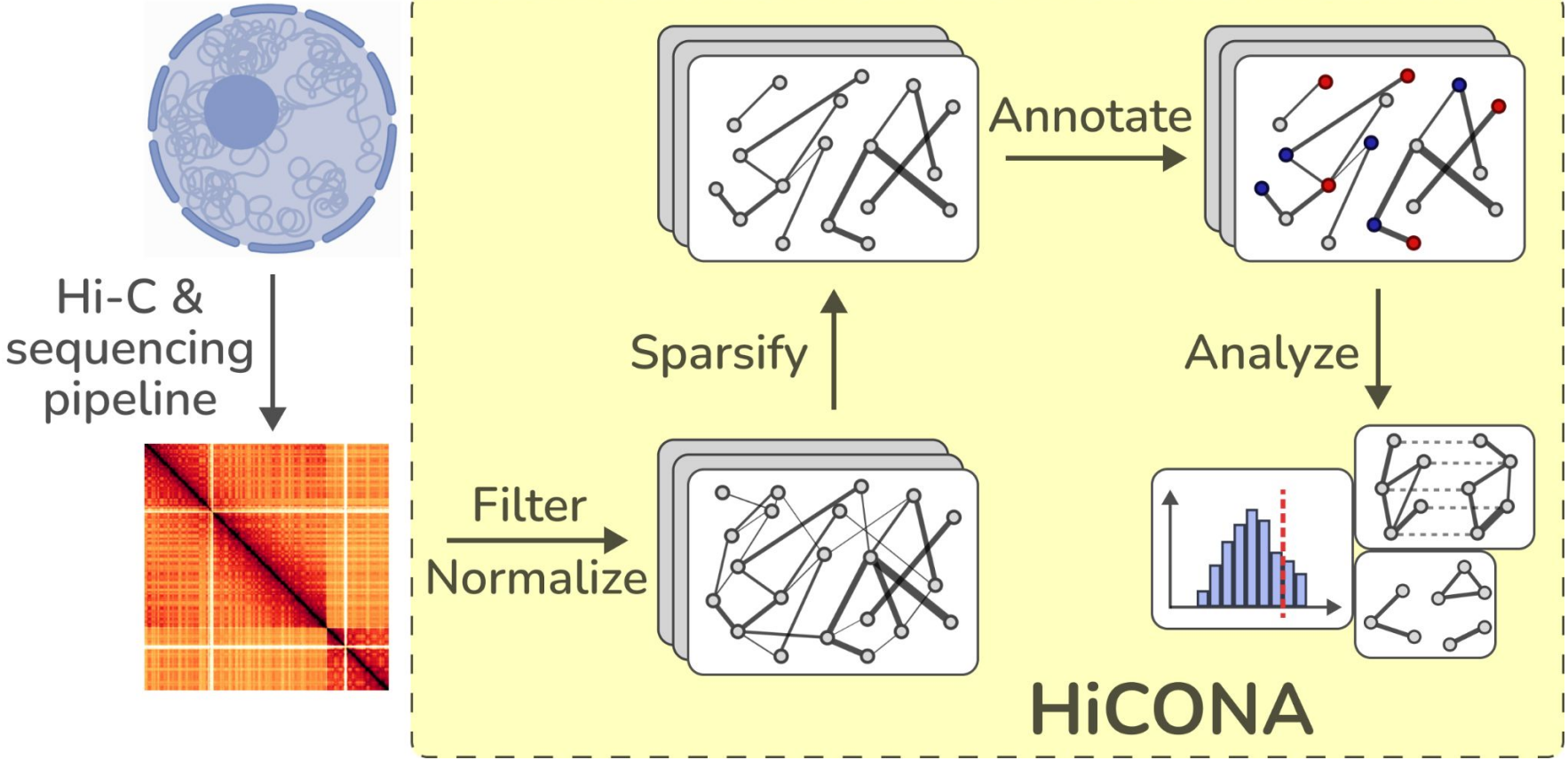
!!! ~10^9 pixels

edge list

Dense Weighted Undirected Graph

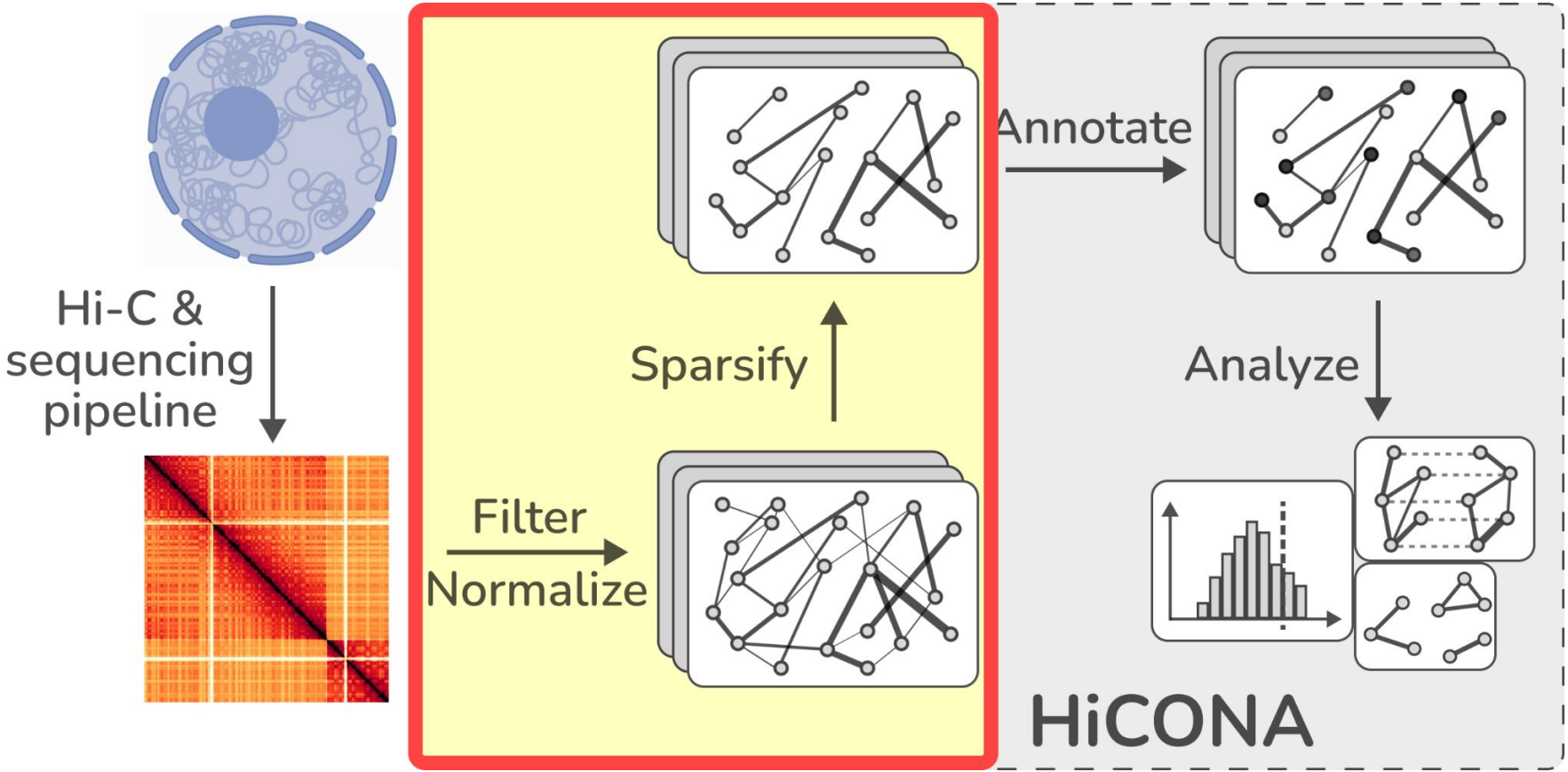


HiCONA Pipeline



Assets from cooltools documentation and BioRender were used

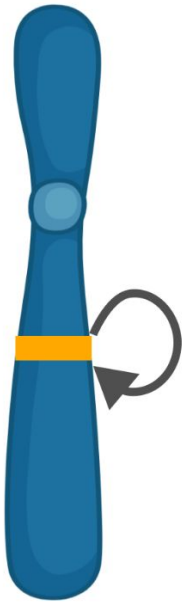
HiCONA Pipeline



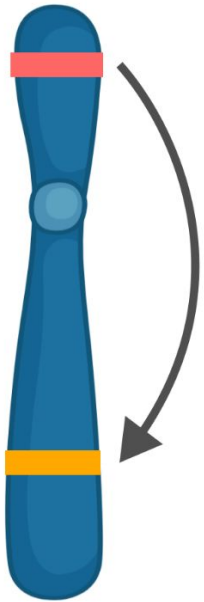
Assets from cooltools documentation and BioRender were used

Raw Pixels Filtering

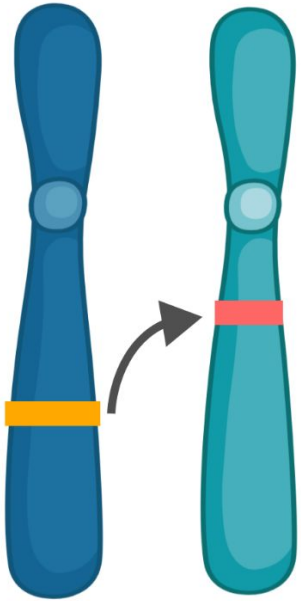
bin 1	bin 2	count
5	5	2
5	6	3
10	180	2
200	230	5
230	233	4



Self-looping



Above genomic distance



Inter chromosomal

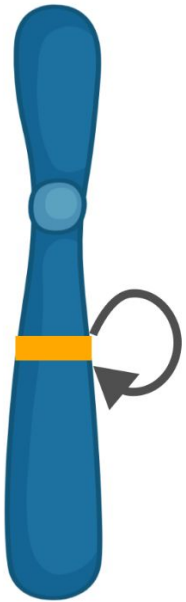
Assets from BioRender were used

Raw Pixels Filtering

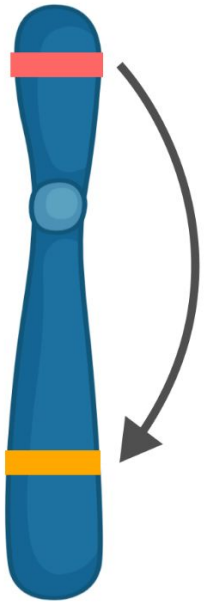
bin 1	bin 2	count	
5	5	2	*
5	6	3	
10	180	2	*
200	230	5	*
230	233	4	

↓

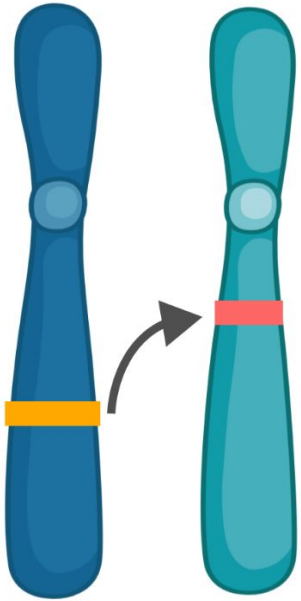
bin 1	bin 2	count
5	6	3
230	233	4



Self-looping



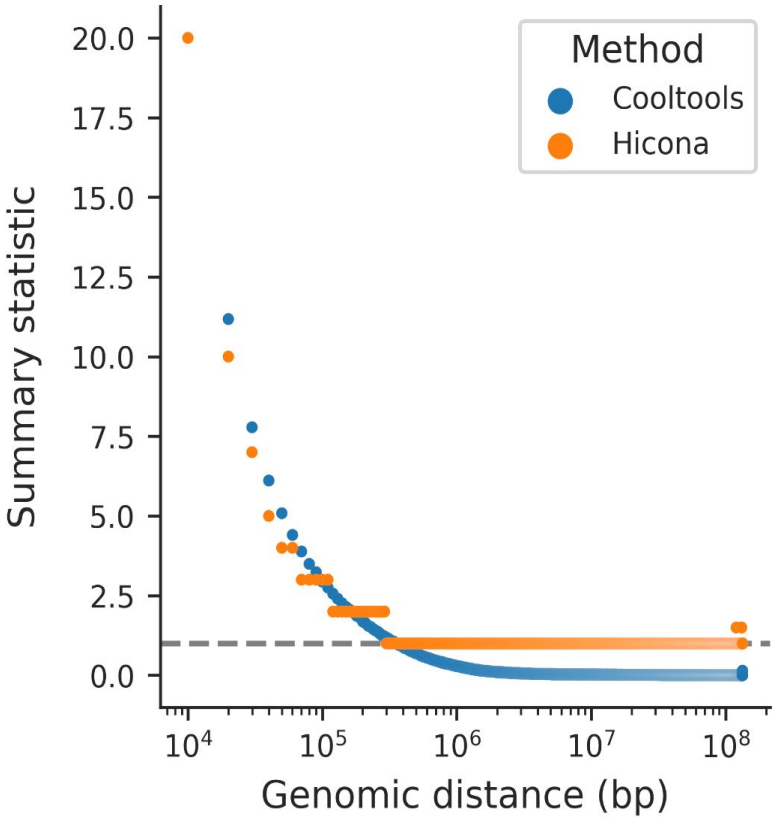
Above genomic distance



Inter chromosomal

Assets from BioRender were used

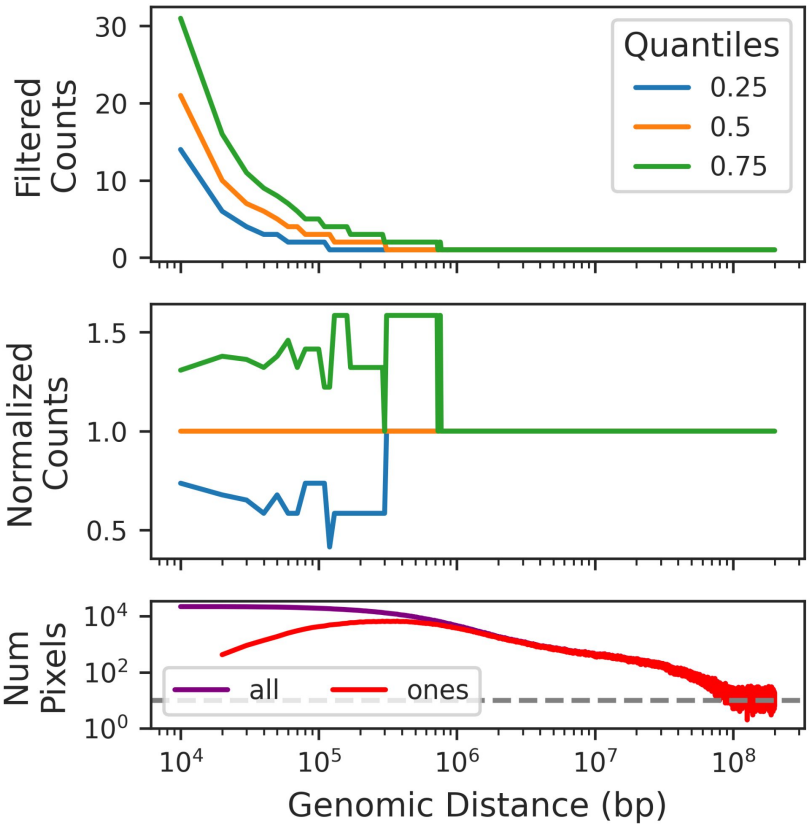
Genomic Distance Normalization



P(x) = median of pixels at distance *x*

Similar to more biologically accurate method but better suited for performance

Genomic Distance Normalization

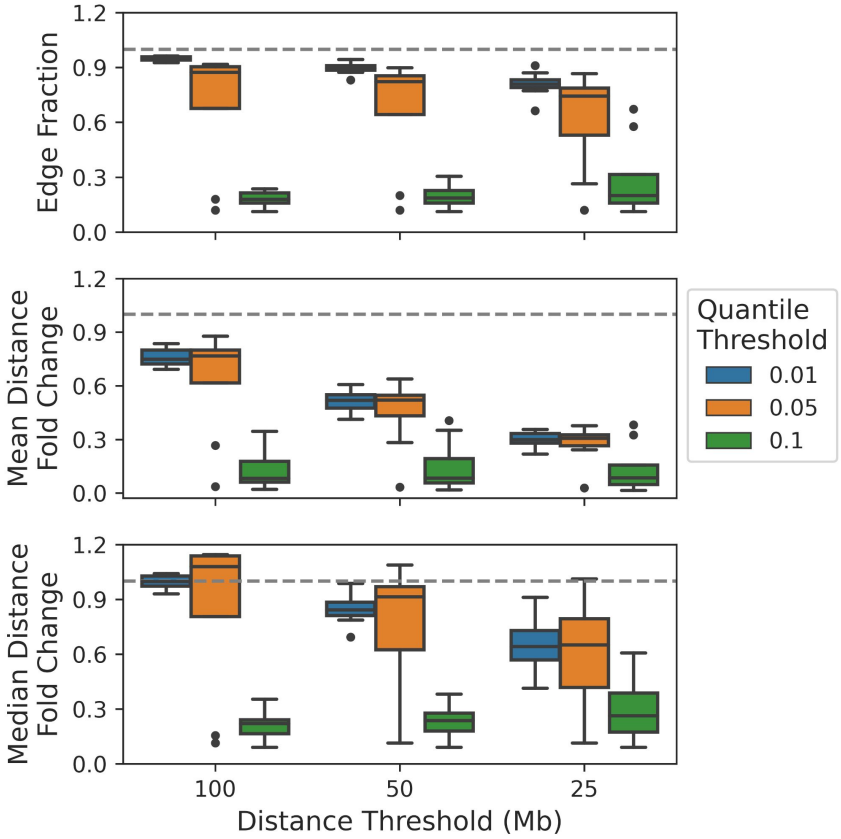


$$norm\_counts = \log_2\left(\frac{obs\_counts}{norm\_factor} + 1\right)$$

Consistent range of values, usually in [0.5, 1.5]

Instability might arise at very high distances (> 80 Mb)

Normalized Pixels Filtering

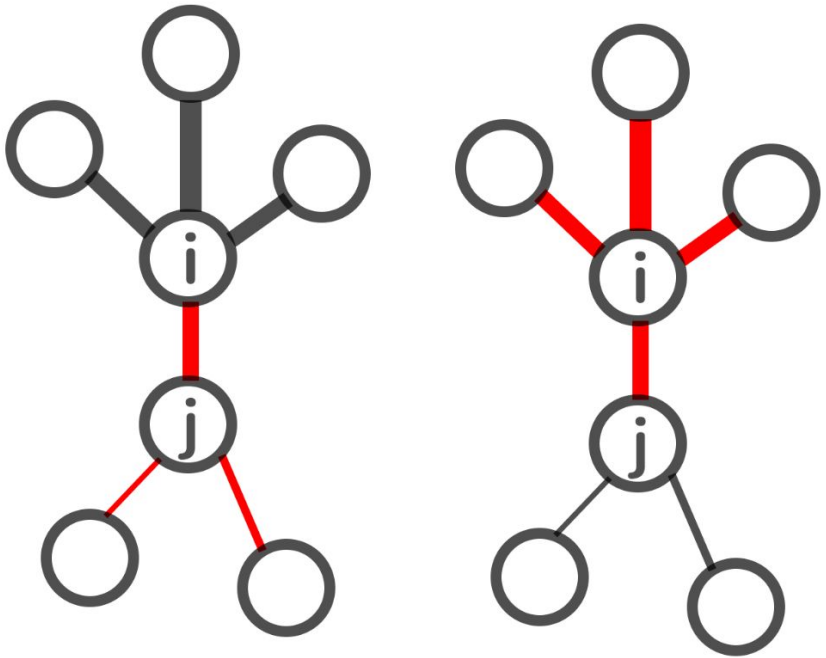


Strict thresholds introduce variability and huge distance reductions
(risk of bias)

Distance: 100 Mb / 50 Mb
Quantile: 0.00 / 0.01

Network Sparsification

Reduce graph density, retain topological properties



Original algorithm from Serrano et al., 2009

Network Sparsification

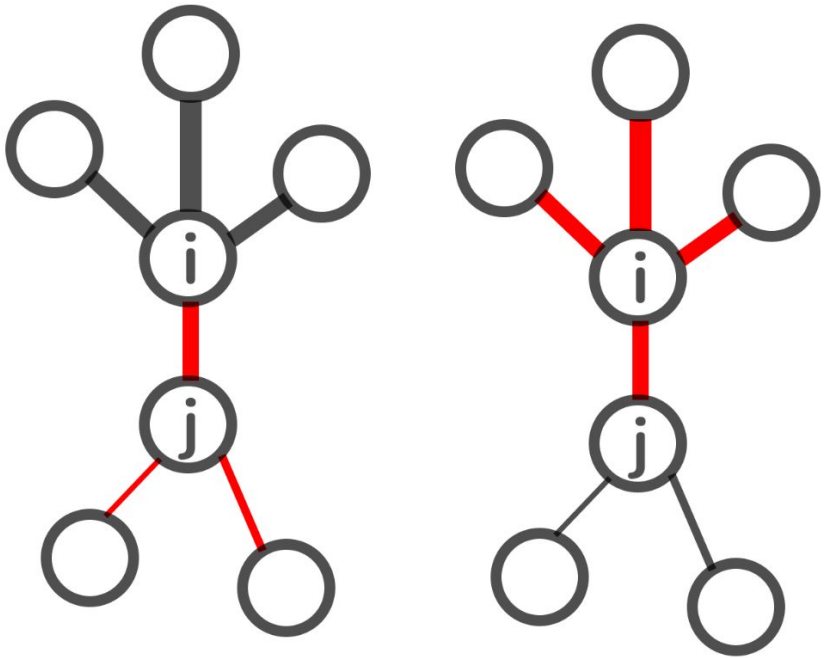
Reduce graph density, retain topological properties

For both *i* and *j* compute

$$\alpha_{ij} = 1 - (k-1)\int_0^{p_{ij}} (1-x)^{k-2} dx$$

k = number of neighbors

p_{ij} = normalized edge weight



Original algorithm from Serrano et al., 2009

Network Sparsification

Reduce graph density, retain topological properties

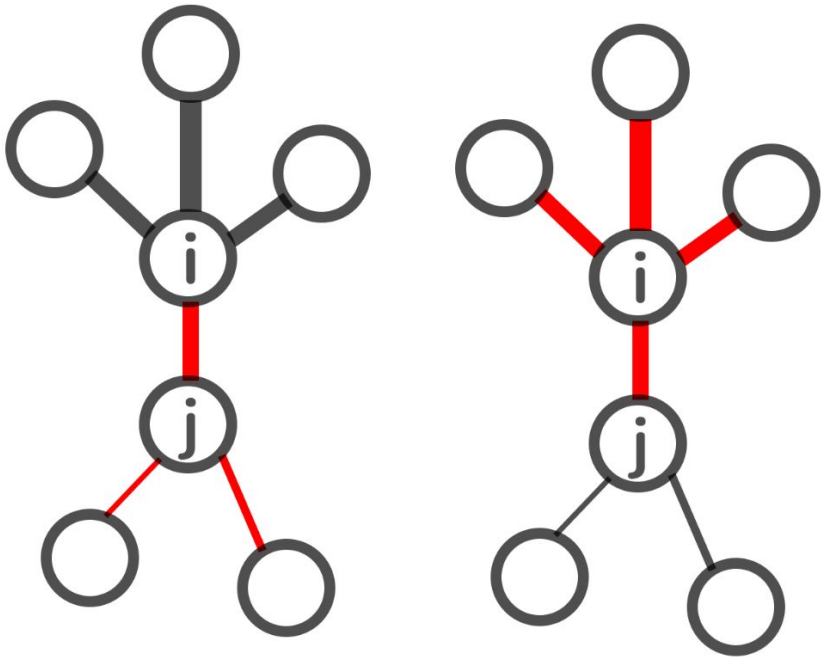
For both *i* and *j* compute

$$\alpha_{ij} = 1 - (k-1)\int_0^{p_{ij}} (1-x)^{k-2} dx$$

k = number of neighbors

p_{ij} = normalized edge weight

set $\alpha_{ij} = \min(\alpha_{ij,i}, \alpha_{ij,j})$ (or max)



Original algorithm from Serrano et al., 2009

Network Sparsification

Reduce graph density, retain topological properties

For both *i* and *j* compute

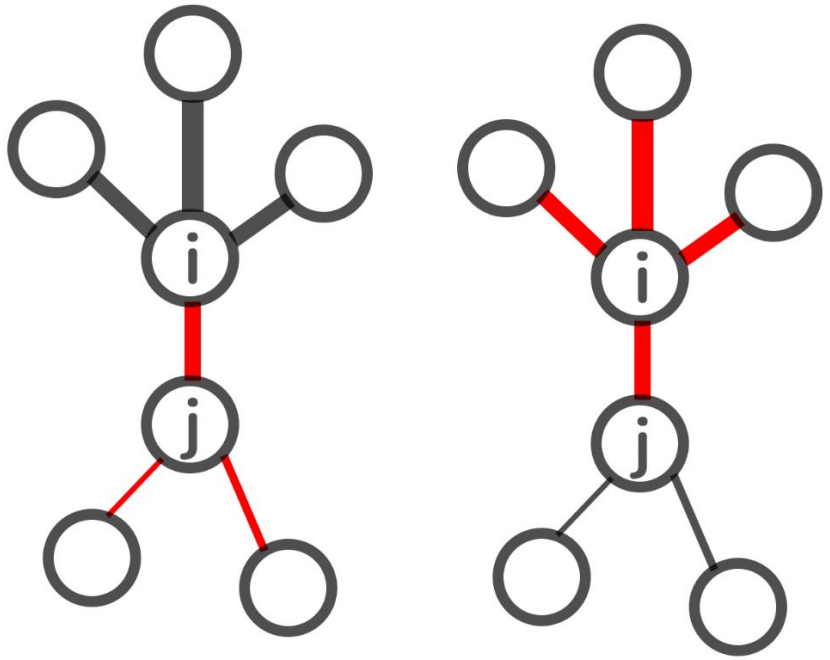
$$\alpha_{ij} = 1 - (k-1)\int_0^{p_{ij}} (1-x)^{k-2} dx$$

k = number of neighbors

p_{ij} = normalized edge weight

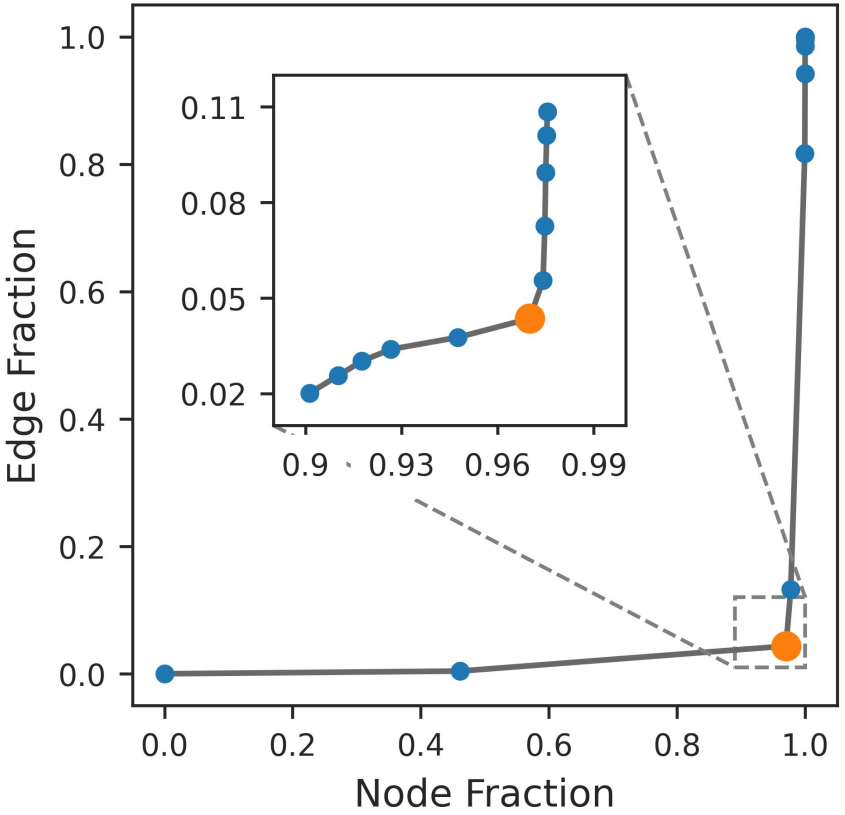
set $\alpha_{ij} = \min(\alpha_{ij,i}, \alpha_{ij,j})$ (or max)

$$\textbf{keep } \alpha_{ij} \iff \alpha_{ij} < \alpha_{threshold}$$



Original algorithm from Serrano et al., 2009

Network Sparsification

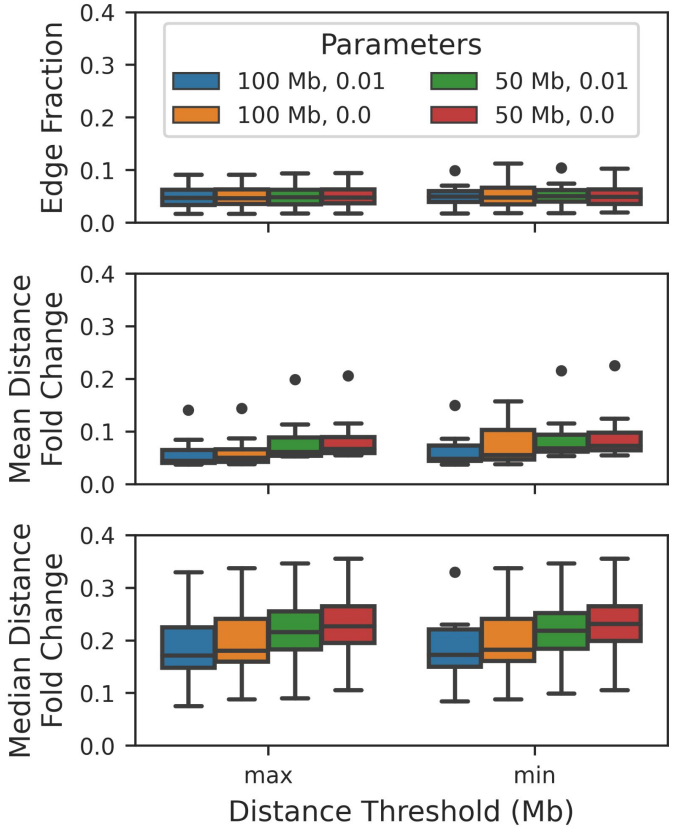


optimal α value

- maximizes node fraction
- minimizes edge fraction

α is chosen via
iterative local search

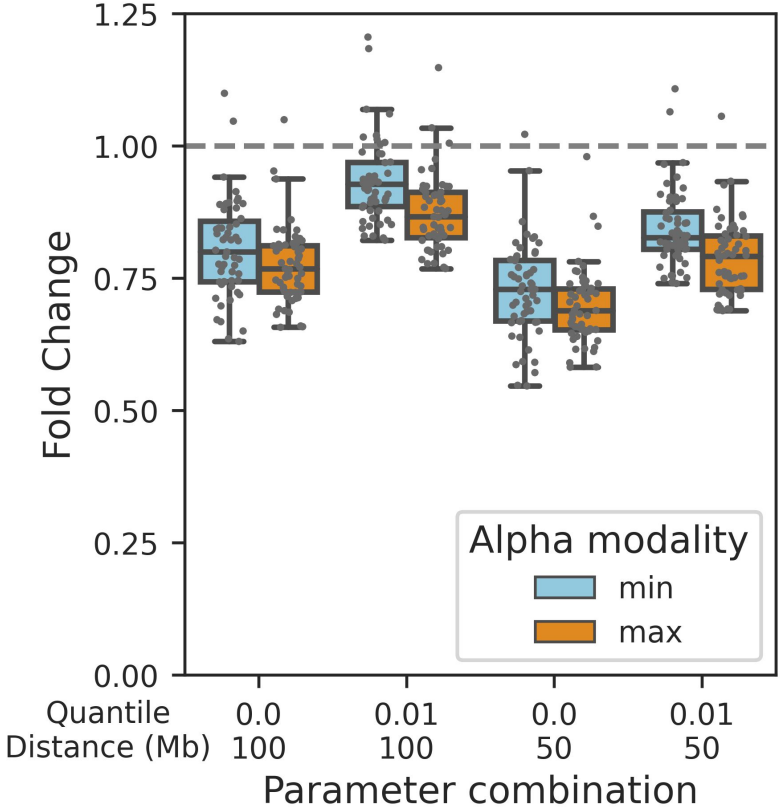
Network Sparsification

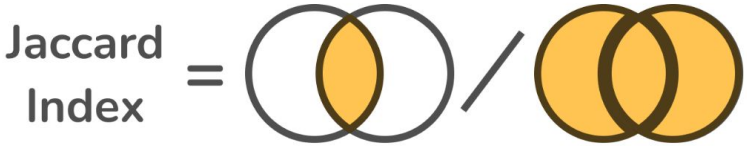


Edge number reduction of 95% on average

Average median distance of 400 kb, compatible with promoter enhancer interactions

Consistency On Replicates



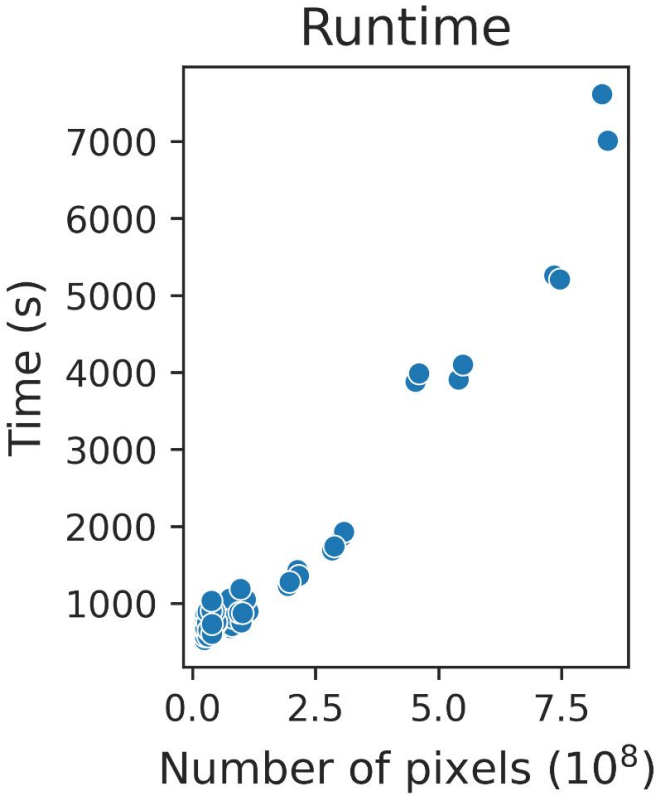


Low Jaccard index (~ 0.1)

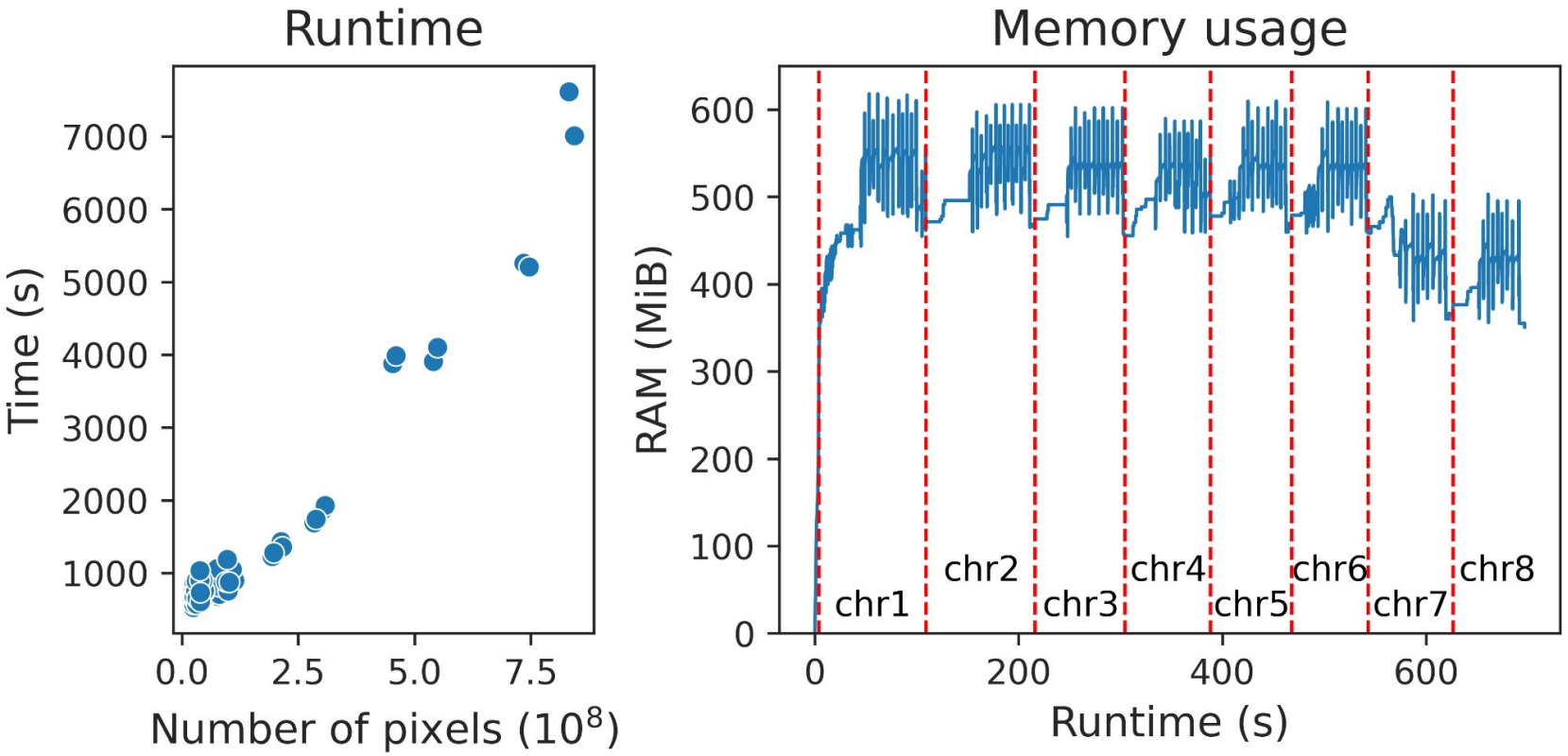
Sparsification retains most correlation

Filtering parameters lead to statistically significant differences

Performance Benchmarking



Performance Benchmarking



Future Of HiCONA

Implementation

Language: Python ≥ 3.8

Unit testing: pytest

Styling: Black

Linting: pyflakes

Documentation

Manual: Sphinx (NumPy)

Tutorials: jupyter notebook

Read the Docs

Functionalities

Network Generation:
pixel filtering
distance normalization
network sparsification

Network Annotation:
adding annotations
removing annotation
one hot encoding

Network Analysis
node-labels permutation
contrast subgraphs
others

Deployment

Package Indexes:

Github

PyPI

Conda

Acknowledgements

I would like to thank

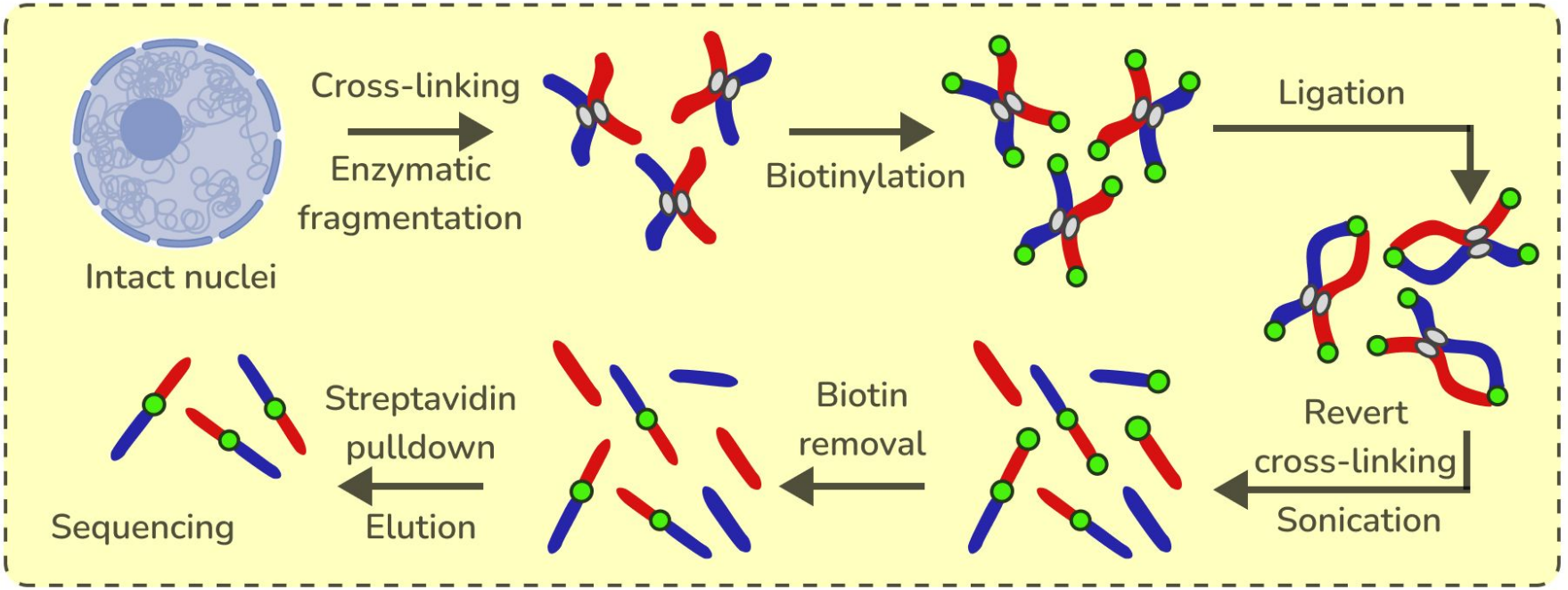
my tutor, **Leonardo Morelli**
my supervisor, **Professor Alessio Zippo**

all the people from the **Chromatin Biology & Epigenetics Lab**
all the people from **Ufficio Bioinformatici Nord**

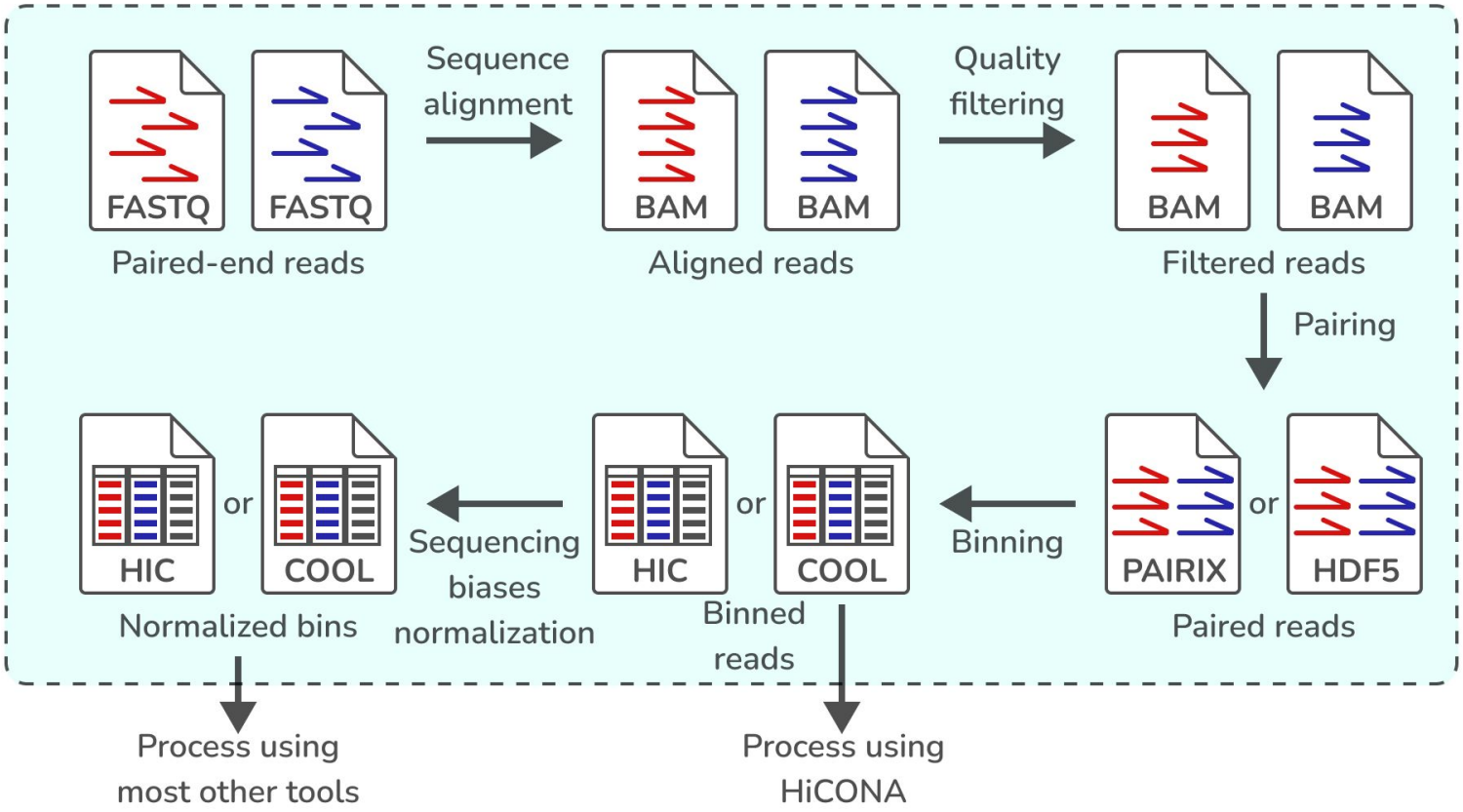
all the people who supported me while writing my thesis

Supplementary Slides

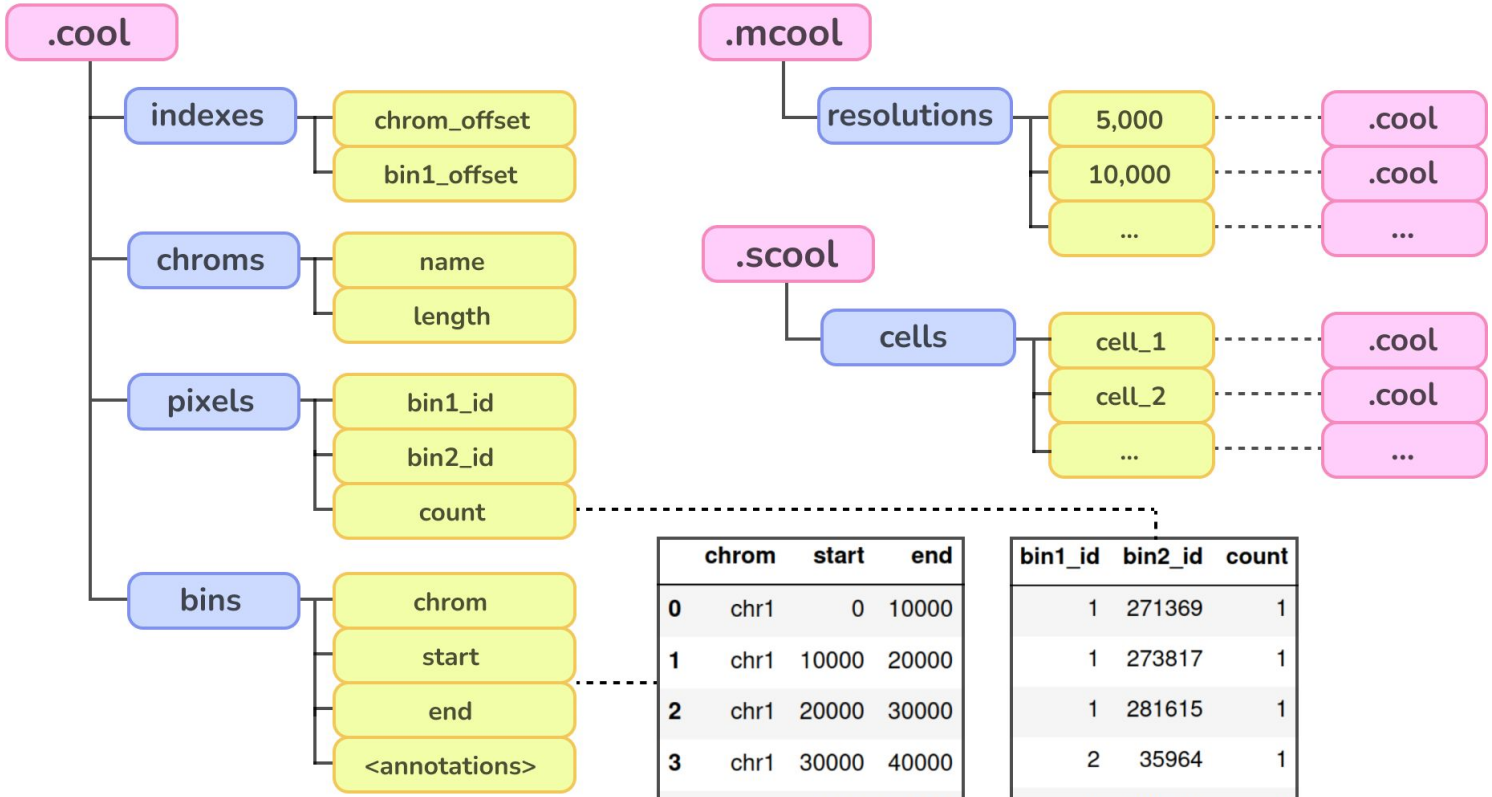
Hi-C in vitro


Intact nuclei
Cross-linking
Enzymatic fragmentation
Biotinylation
Ligation
Revert cross-linking
Sonication
Biotin removal
Streptavidin pulldown
Elution
Sequencing

Hi-C in silico



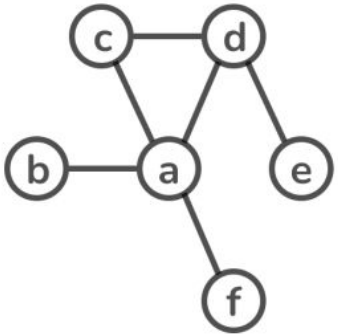
.cool formats



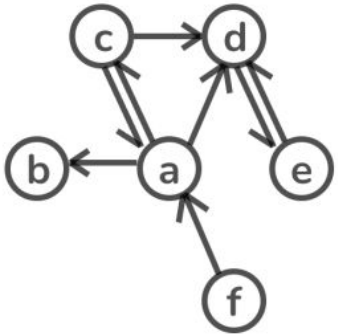
	chrom	start	end
0	chr1	0	10000
1	chr1	10000	20000
2	chr1	20000	30000
3	chr1	30000	40000
4	chr1	40000	50000

bin1_id	bin2_id	count
1	271369	1
1	273817	1
1	281615	1
2	35964	1
2	235740	1

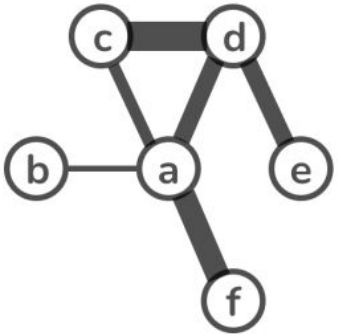
Graph Theory



Unweighted
undirected



Unweighted
directed



Weighted
undirected

{(a,b), (a,c), (a,d),
(a,f), (c,d), (d,e)}

Edge list

	a	b	c	d	e	f
a	0	1	1	1	0	1
b	1	0	0	0	0	0
c	1	0	0	1	0	0
d	1	0	1	0	1	0
e	0	0	0	1	0	0
f	1	0	0	0	0	0

Adjacency matrix

a → b → c → d → f
b → a
c → a → d
d → a → c → e
e → d
f → a

Adjacency list

Sparsification Algorithm

Algorithm 1 Network sparsification algorithm, Serrano et. al, 2009

Input: G = a weighted, undirected graph defined over vertices V and edges E, α = sparsification cutoff

Output: G' = the sparsified graph

Step 1: sparsification score computation
Set $\alpha_{ij,curr} = 1, \forall (i,j) \in V, j \in$ neighbors of i (Note: $\alpha_{ij} = \alpha_{ji}$ since G is undirected)
for $i \in V$ **do**
    w_{sum} = sum of the weights of all edges incident to i
    k_i = degree of node i
    for $j \in$ neighbors of i **do**
        $w_{norm} = w_{ij}/w_{sum}$
        $\alpha_{ij,new} = 1 - (k_i - 1)\int_0^{w_{norm}} (1-x)^{k_i - 2} dx$
        $\alpha_{ij,curr} = \min(\alpha_{ij,curr}, \alpha_{ij,new})$
    end for
end for

Step 2: network filtering
G' = empty graph
for $e_{ij} \in E$ **do**
    if $\alpha_{ij,curr} < \alpha$ **then**
        Add edge e_{ij} to G'
    end if
end for

Node-labels permutation algorithm

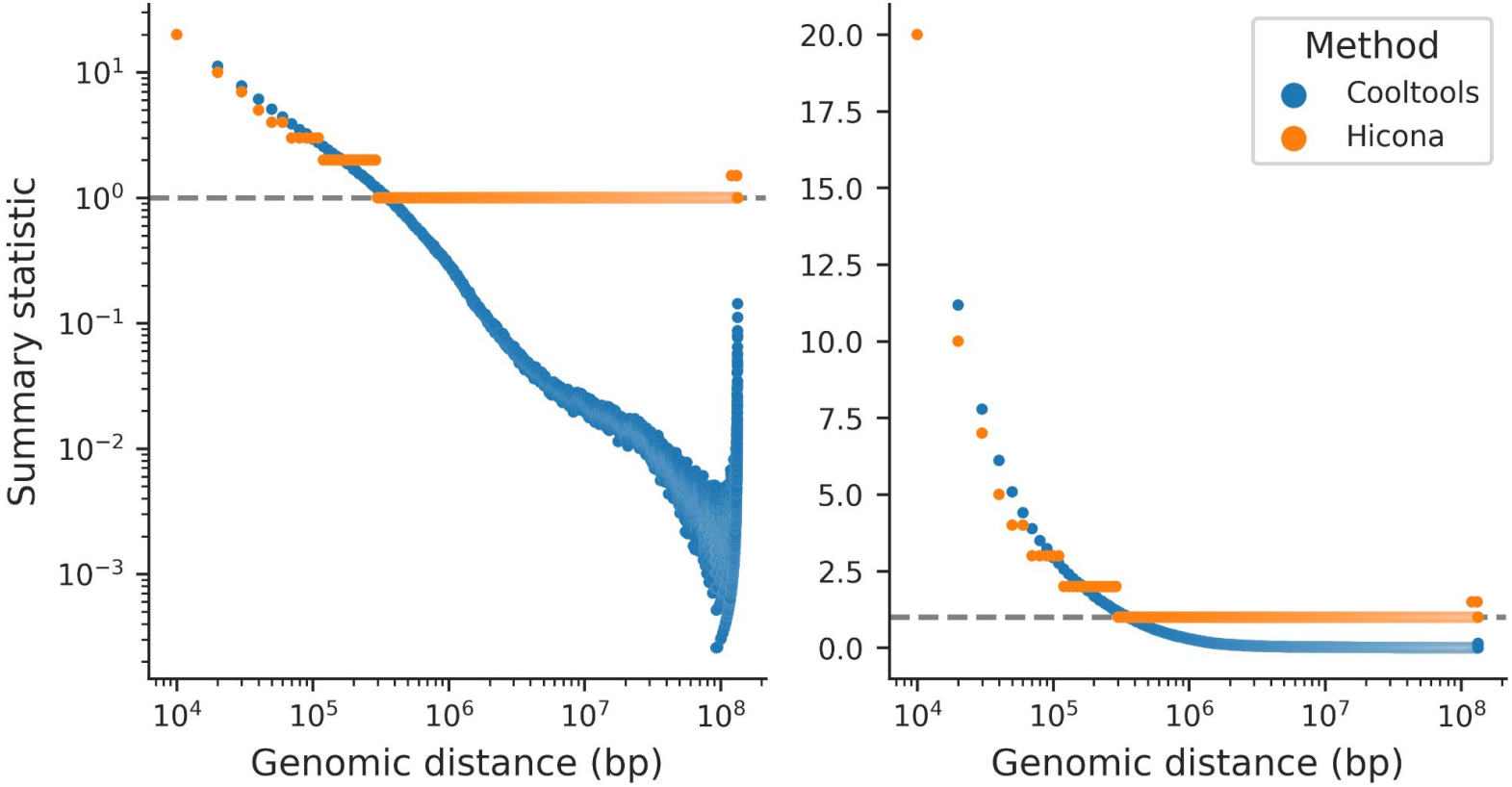
Algorithm 2 Node-labels permutation algorithm

Input: G = a graph defined over vertices V and edges E; each vertex can be annotated as A, B, both or neither. S = the average of a node-level statistic over a set of nodes. N = the number of permutations to perform.

Output: The p-value for the test $\begin{cases} H_0 : S(A) \leq S(B) \\ H_1 : S(A) > S(B) \end{cases}$

1: Define a vector of node labels $L = [l_1, l_2, \ldots, l_{|V|}]$, such that each element is a tuple $l_i = (l_{Ai}, l_{Bi})$ with $l_{Ai} = 1$ if v_i is annotated as A, 0 otherwise (define l_{Bi} analogously)
2: Define the sets $A = \{v_i | v_i \in V, l_{Ai} = 1\}$, $B = \{v_i | v_i \in V, l_{Bi} = 1\}$
3: Compute $s_{original} = \log_2(S(A)/S(B))$
4: Initialize $counter = 0$
5: **for** $n \in [1, N]$ **do**
6: $L_n = [l_{\pi_n(1)}, l_{\pi_n(2)}, \ldots, l_{\pi_n(|V|)}]$, where π_n is some permutation function
7: $A_n = \{v_i | v_i \in V, l_{n,Ai} = 1\}$, $B_n = \{v_i | v_i \in V, l_{n,Bi} = 1\}$
8: $s_{permutation} = \log_2(S(A_n)/S(B_n))$
9: **if** $s_{permutation} > s_{original}$ **then**
10: $counter = counter + 1$
11: **end if**
12: **end for**
13: $p_{val} = (counter + 1)/(N + 1)$

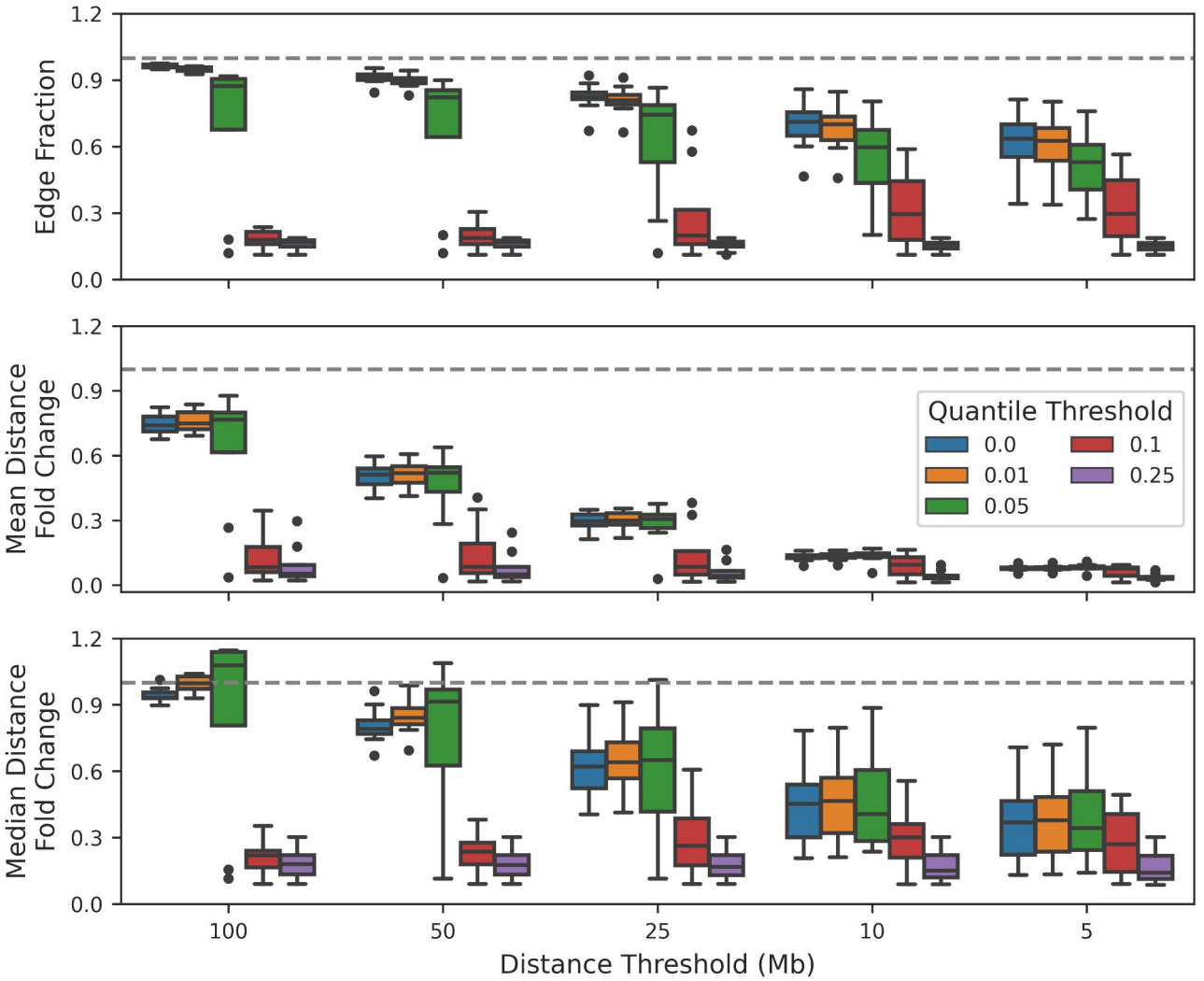
Normalization factors



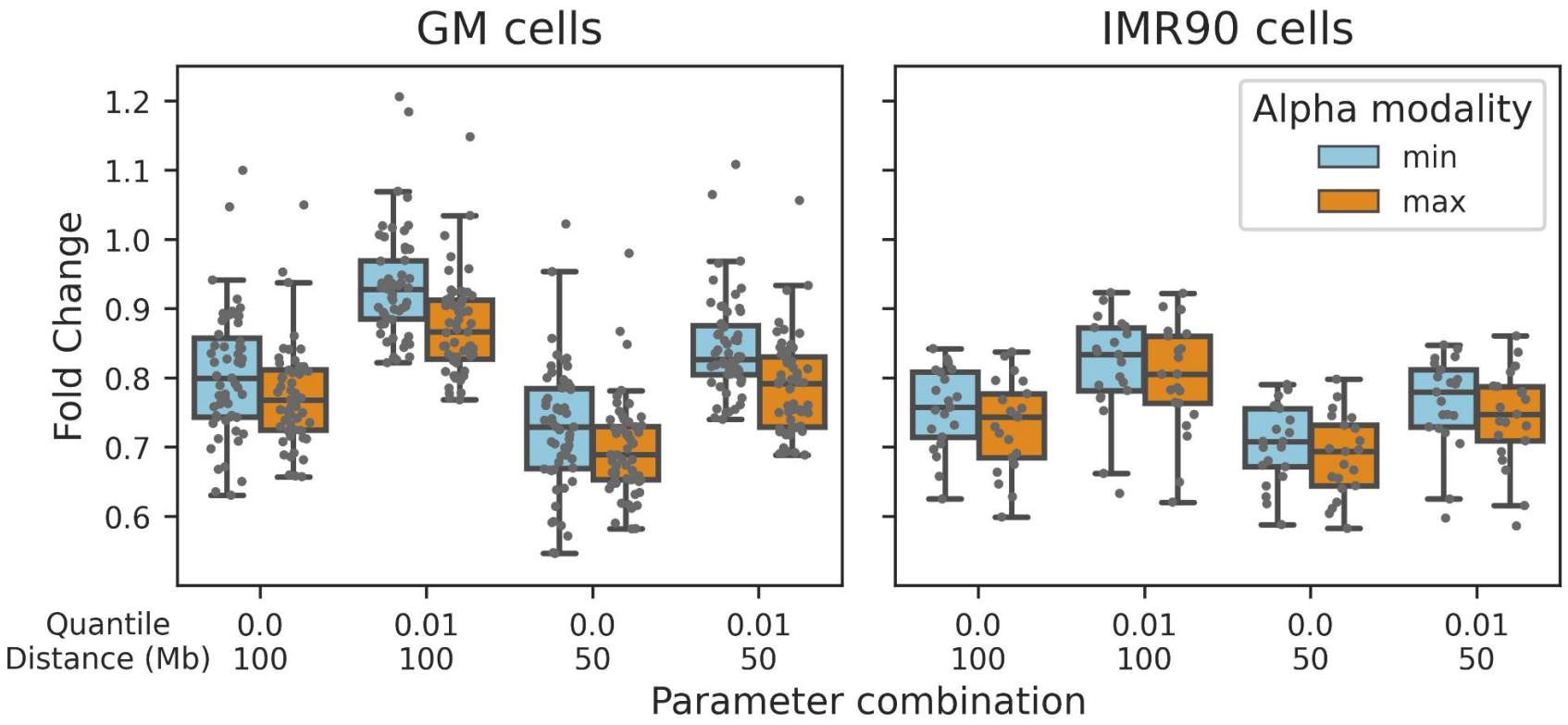
Test Files

4DN ID	Publication	Experiment	Enzyme	Cell Type	Res (kb)	Pixels
4DNFIYECESRC	Rao, 2014	In situ Hi-C	MboI	GM12878	5	$1.89 \cdot 10^9$
					10	$1.57 \cdot 10^9$
4DNFIIG4IWKW	Rao, 2014	In situ Hi-C	MboI	IMR90	5	$4.95 \cdot 10^8$
					10	$4.03 \cdot 10^8$
4DNFIIFAUT24	Rao, 2014	In situ Hi-C	MboI	HMEC	5	$2.15 \cdot 10^8$
					10	$1.83 \cdot 10^8$
4DNFIYL35EHL	Rao, 2014	In situ Hi-C	MboI	HUVEC	5	$2.00 \cdot 10^8$
					10	$1.77 \cdot 10^8$
4DNFIUTE4F4B	Oksuz, 2021	In situ Hi-C	HindIII	H1-hESC	5	$1.41 \cdot 10^8$
					10	$1.26 \cdot 10^8$
4DNFIUPGJLFO	Oksuz, 2021	In situ Hi-C	DpnII	H1-hESC	5	$8.91 \cdot 10^7$
					10	$7.41 \cdot 10^7$
4DNFIPVA6VYB	Oksuz, 2021	In situ Hi-C	DdeI	H1-hESC	5	$8.16 \cdot 10^7$
					10	$7.41 \cdot 10^7$
4DNFIR1FK55F	Oksuz, 2021	Micro-c	MNase	H1-hESC	5	$5.73 \cdot 10^7$
					10	$4.39 \cdot 10^7$

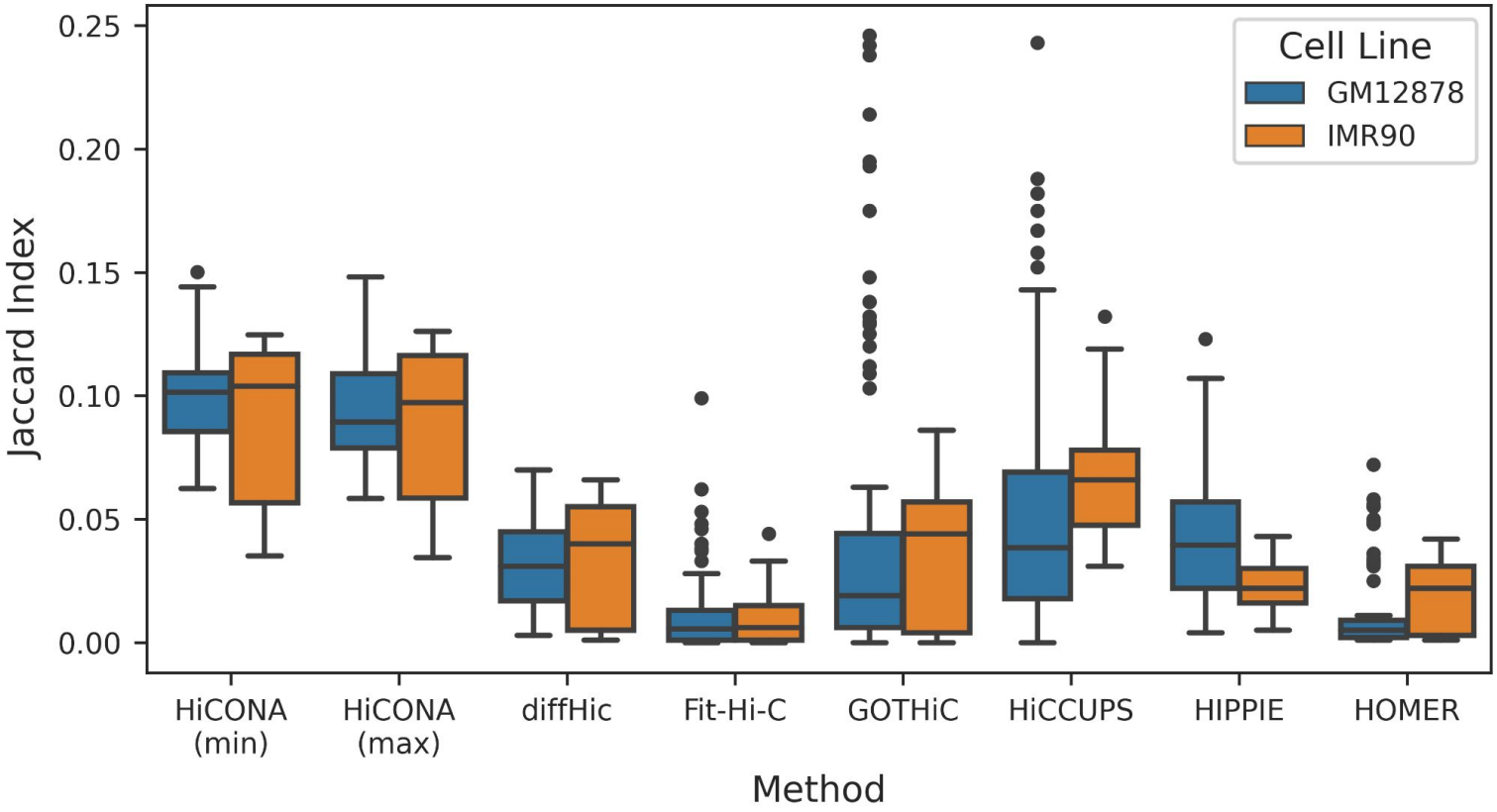
Full threshold grid



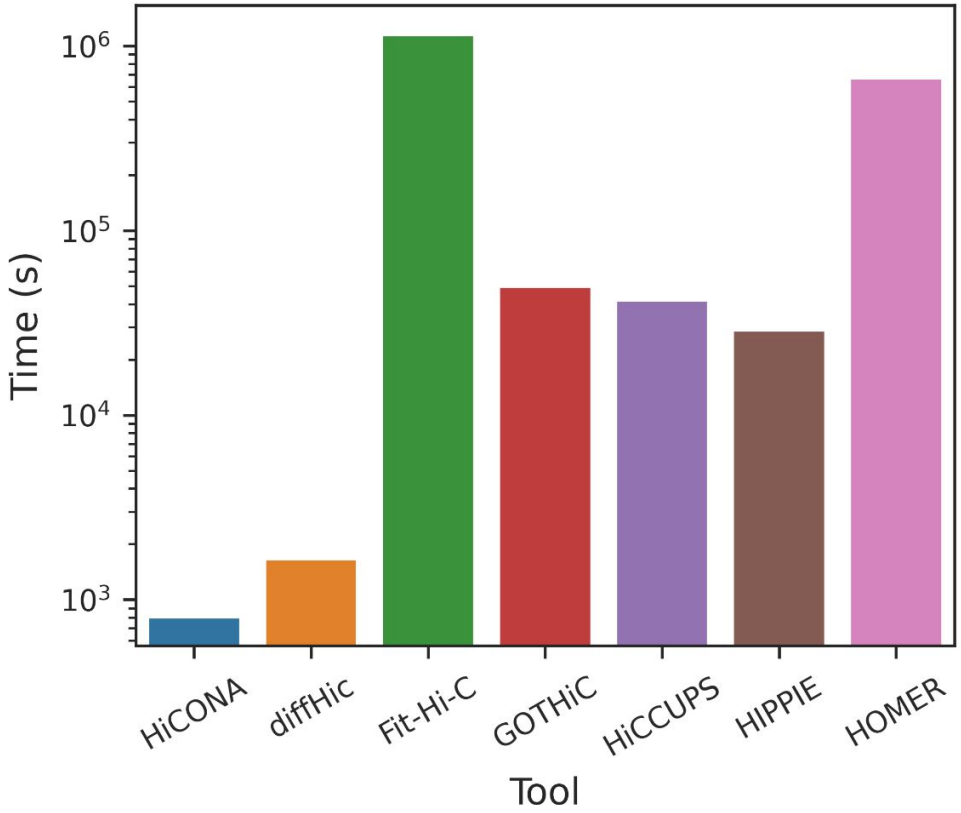
Consistency on replicates



Jaccard - loop callers

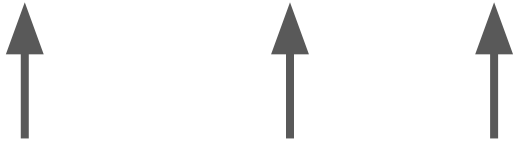


Runtime - loop callers



Quantile filtering

8,7,6,6,6,5,1,1,1,1,1,1,1,1,1,1,0.2,0.1,0.1



0.4 0.2 0.1

Quantile	Remaining	Reduc
0.1	8,7,6,6,6,5,1,1,1,1,1,1,1,1,1,1,0.2	10%
0.2, 0.4	8,7,6,6,6,5	70%