**Issues in /previous\_versions/version3\_debug/WIP\_improved\_version\_4.py**

During calculating enrichment score for each candidate stripe position, I stored

1) Hi-C matrix of the neighboring region (Line 175-182),

2) the enrichment score (i.e., -log10(P) values) (Line 261-263), and

3) the chosen slice (Line 360-362) in folder /temp.

Then I used vis.py to visualize these matrices into folder /temp\_fig. An example region is:

Graphical user interface

Description automatically generated

The upper one is the Hi-C region with the candidate stripe in the middle (spanning length = 700 bins = 17.5 Mb).

The lower subfigure is the enrichment scores-

blue: -log10(P) calculated for each of the 700 bins,

gray: threshold (0.15)

red: the max slice found by the stripe caller

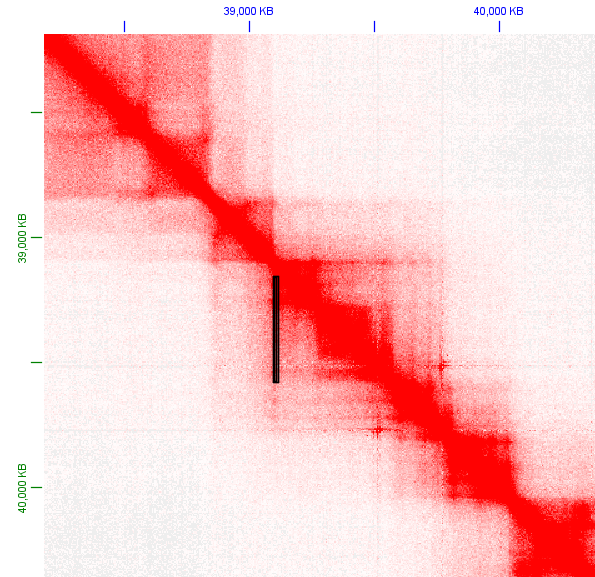
I copied the parameter from the code you shared with me, but I think max\_range = 17.5 Mb might be too large, which is the reason that we cannot see obvious stripes in the figures in /temp\_fig.

From JuiceBox visualization, I think >10Mb “stripes” are more like short A/B compartments interacting with other longer compartments.

A screenshot of a computer

Description automatically generated with medium confidence

One example “stripe” (the black line is the called stripe):



In the stripe caller, the getPeakAndWidths() function takes a 1800x1800 matrix. Even if improving the resolution to 5 K, this is still 9 Mb region. Is it possible to make it smaller?

Here I also tried another earlier version:

**Issues in /previous\_versions/version3\_Fan\_debug/stripe\_caller\_improved\_v3.py**

The major differences are:

1). The part for finding candidate stripes are my previous slower version without dynamic stripe widths,

2). I used “dump observed KR” instead of “dump oe KR” (Line 25), which skips normalization by distance,

3). max\_range=2000000 (2 Mb), resolution=5000 (5 Kb),

4). remove some “black-list” regions (centromere/start or end of the chromosome).

I also saved the 3 matrices in /temp and visualize them in /temp\_fig.

Since there are too many candidate stripes, I only visualized those with a picked “max slice length” > 50 with vis.py.

An example region:

Graphical user interface, application

Description automatically generated

However, when I checked the called stripes (GM12878\_HiC\_stripes\_chr1.bedpe) with JuiceBox, I found there are still some false negatives:

An example (green circles are potential false negative stripes):

A picture containing background pattern

Description automatically generated