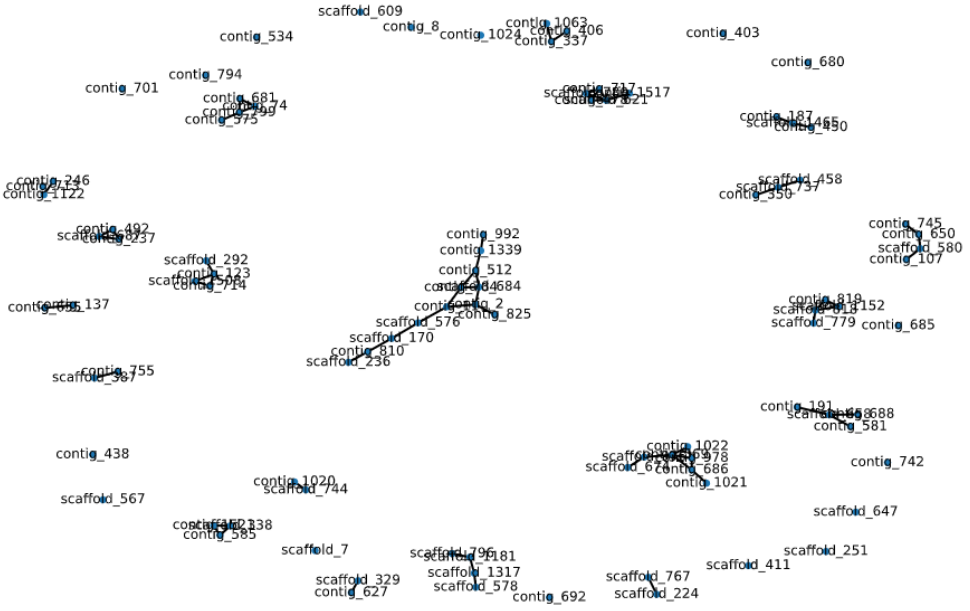
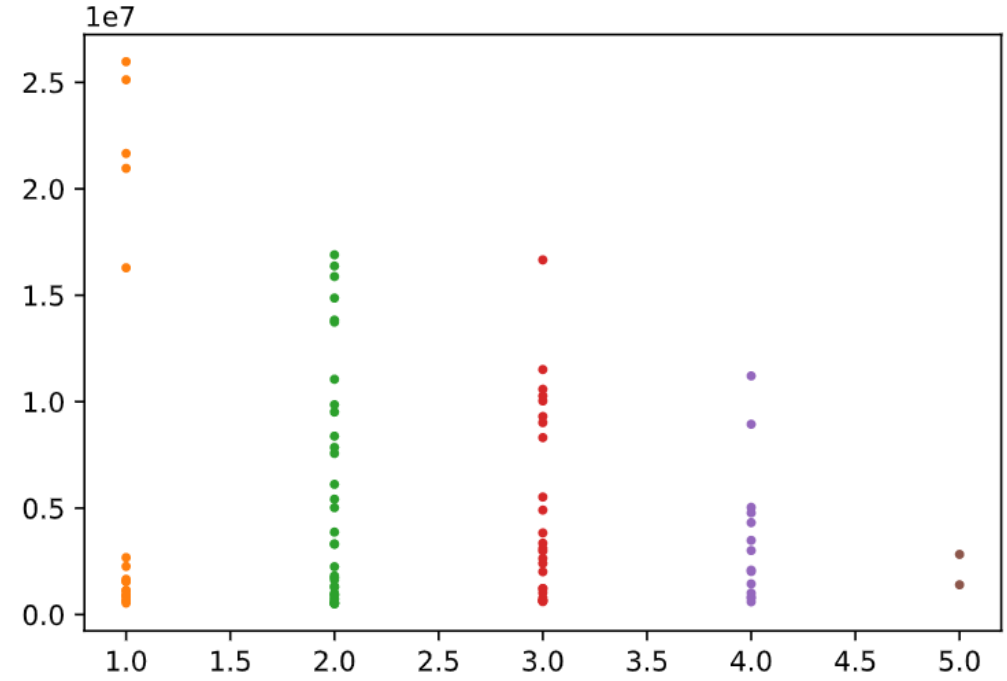
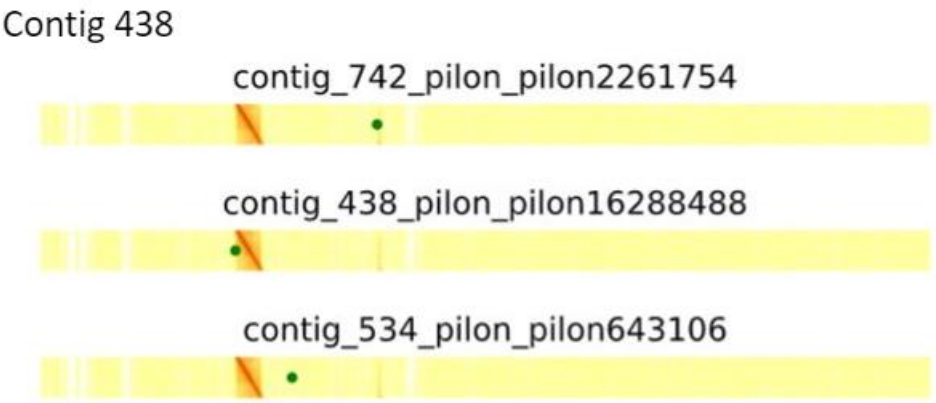
**Project outcome**

This project is a one-year project; the complete task is to use python code to automatically identify abnormal regions (contigs) in the HIC heatmap before using some method to automatically correct and sort the several contigs belonging to one chromosome to sort them correctly; after which all chromosomes are stitched together to form a heatmap that should Because of time constraints, the progress I have made so far has been to use code to complete the automatic recognition function. However, due to the specific nature of the HIC heatmap, there is no more traditional or standard value for the threshold of peak detection for each different chromosome or contig. Specifically, it may be that for contig1, a threshold of mean + 2 * std (around 95%) will successfully detect all unusual contigs without. However, for contig2, this threshold may fail to detect a particular contig peak, and for contig3, it may incorrectly detect a point that is not a peak. Since there may be hundreds or thousands of contigs in a heatmap, it is impossible to set different thresholds for each contig (contig1 95%, contig2 70%, etc.) if automated processing is to be performed. Therefore, after completing this function, I experimented with and optimized this peak detection method differently. First, the previous peek detection method, which transformed a 2D array into 1D, used np. np.mean, calculating the mean of each column and adding them to the sequence, thus forming a 1D array with constant length but changing the width to 1, and then smoothing it using np. convolve When the above problem was found, I calculated the mean of each contig matrix in the 2D array instead of doing it on a per-column basis. However, this method still worked mediocrely; according to biological theory, if contig2 is found on contig1, then contig1 must be found on contig2, so I constructed an optimization method where two contigs are only found to be together if they are each detected on each other; in the meantime I I chose to lower the threshold so that the algorithm would rather detect some points that do not peak than miss those peaks, and then proceeded with this approach; the result was partially improved. At the same time, I generated several different images (below) for evaluation. I then divided the code differences into several classes and wrote documentation to facilitate the work of the following participants.






**Experiences and reflections**

During this internship, I have used many of the knowledge or experiences, and habits I learned during the course. First of all, the project is composed entirely of python, so the basic understanding of python that I learned in my undergraduate studies has provided me with a good foundation, and at the same time, even if it is a different voice because the code logic is always between, the subsequent learning and experience of programming the project can also assist me in coding very well. Because there are many details and different packages in any programming language, even when I was studying, I couldn't cover everything the teacher taught me. However, I could substitute other parts in my assignments, so during my university years, I made it a habit to learn how to write and some standard packages on various forums, youtube, and Google. Finally, as I was working on this project alone, I had to do a meeting and presentation from time to time to introduce the project and its progress. When I was doing the master's course, I did a lot of meetings and presentations because I had gained enough experience to go through these processes in an organized and unstressed way.