Report:

In general, our algorithm performs Expectation Maximization (EM) on a sequence of several SNPs for each individual at a time, returns a phase with maximum probability, and moves on to the next SNP sequence of the same window size until all SNPs are covered. We use parsimony approach on an overlap region of each two consecutive sequences to arrange and connect the chunk phases.

We first transform the data file into a 2D list that each column represents the genotype of the SNP over the 50 individuals, while each row is an individual.

To generate all possible phases of a genotype sequence, our algorithm runs through each entry in the list of the genotype sequence and then add corresponding haplotype entry each time (either 0 or 1). When encountering the first '1' in the genotype, the algorithm only adds '1' to the existing haplotype sequences to avoid creating phase duplicates. If another '1' is encountered, the algorithm duplicates the current haplotypes, adds '0' to the duplicate, and adds '1' to the original haplotypes. By the end of the algorithm, we have one haplotype for each phase listed without its complement. We then use a function to generate the complements, and create a phase list in which each index is a list of haplotype pair of that genotype.

For expectation maximization, the algorithm first creates a list of all individuals' phases and a list of probabilities of each individual's phase probability initialized to 1.0 with the same dimension so that each index of the two lists corresponds to each other. The algorithm also initializes a haplotype probability dictionary and a haplotype location dictionary. The algorithm goes through each individuals' phases and add strings of the haplotypes and their complements into the haplotype dictionary, during which the number of distinct haplotypes is updated. The value of each haplotype key is its probability and is initialized to 1.0/(number of haplotypes). The location of that haplotype is stored in the haplotype location dictionary that the haplotype itself is the key, while the value is the list of locations ([individual][phases]) of that haplotype. After the initialization, EM iterates for 3 times. For each iteration, the algorithm calculates and updates the probabilities of each phase and each haplotype. This function returns a list of phases with highest probability among each individual's phases.

For each phase of an individual, there are two haplotypes with the length equal to the window size. There exists two ways to add them to the individual: appending them directly to the end and switching them before appending them to the end. To find the optimum way of appending, the algorithm takes out the overlapping region of the latter part of the last chunk of haplotype pair and the former part of the current chunk of haplotype pair for the individual. The algorithm goes through one individual at a time. For the first individual, the choice was random and the chosen overlap region is added to a known overlapping region list. Starting from the second, the algorithm compares the degree of similarity between the current overlapping region and the list of known overlapping region. The arrangement of haplotype pair with the higher degree is chosen and appended to the haplotype pair of that individual. Every time a haplotype pair is chosen, the corresponding pair of overlapping region is added to the known list for the reference later individuals. Therefore, the logic is to always pick the haplotype pair that is most similar to that of other individuals, which is achieved by storing the optimum arrangement in the known list of overlapping region for later individual's haplotype pair comparison.