Helen Huang (Xin)
Professor Eran Halperin
CS CM 124 Programming Project (haplotype phasing)

Project Report

In this project, I performed 3 rounds of **Expectation Maximization (EM)** on each segment of haplotype. Three iterations are usually enough for the probabilities to converge given 50 individuals (or if it does not converge after 3 rounds, it will never converge). Each segment has a dynamic window size, which is capped at 18. Depending on the number of unknowns (‘1’s) in the segment, the window size was adjusted to an optimal length. Between two adjacent segments, 6 SNPs are overlapped to align the joints.

The optimal window size is determined by the function findWinSize(), which iteratively calls addlist(): For each individual, I generated a list (tempList) to include all possible haplotypes given that individual’s genotype. If the tempList has a length > 512 (2^9, meaning more than 9 ‘1’s appearing in that segment), then the function immediately return -1 to ask the outer function findWinSize() to decrease window size. Otherwise, the tempList is extended to the overall list of all possible haplotypes for all individuals (e.g. 50), the hapList. The hapList is then sorted and the duplicates removed. The length of hapList have its own threshold, so that when the number of possible haplotypes for all individuals exceed 2000, findWinSize() also decreases the window size. These adjustments even out the EM calculation time for each segment, and thus makes the overall running time more controllable. Theoretically, the window size can be anything between 9 and 18. These threshold parameters are determined by training on the example dataset.

Run-Time (s)	Speed (SNP/s)	Overlap	hapLen limit	Single hapLen limit	winSize limit	Switch Accuracy
3287	12.0	0	5000	5000	32	0.8621136
1051	37.6	0	2000	2000	18	0.8492043
739	53.4	0	static	static	12	0.8304711
770	51.3	0	2000	512	18	0.84502
869	45.4	0	2000	512	22	0.8478094
774	51.0	0	3000	512	18	0.8459252
1581	25.0	0	3000	1024	18	0.8521456
1494	26.4	3	2000	512	22	0.8960169
954	41.4	5	1000	256	18	0.9015664
1367	28.9	7	1000	256	18	0.905208
1117	35.3	6	2000	256	18	0.9042548
2016 -> 1186	19.5 -> 33.3	6	2000	512	18	0.9067542
2036	19.4	5	2000	512	18	0.903525

Multiple indices are built in functions buid_idx_hapQueue(), build_idx_indivQueue(), and build_pvalue_indivQueue() for cross-indexing. The EM algorithm is performed in the function updateP(). This function updates the probability values of each possible haplotype, which is originally initialized to $\frac{1}{\#\ of\ all\ haplotypes}$ in the function build_pvalueInitList(). A two-dimensional list (maxList) is generated at the end of updateP() to store the most probable haplotype pair (haplotypList [kkk], haplotypList [lll]) along with the corresponding probability product for each individual. EM calculation simply follows the formula we learnt in lectures:

$$p_i{}^{m+1} = \frac{\Sigma_{\mathrm{i}=1}^{n} E[h_j \mid g_i]}{2n}$$

In the function storeList(), the last 6 SNPs of the selected haplotype pair ([0] at the top, [1] at the bottom) for each individual are stored in a two-dimensional list (oldMaxList) for later retrieval. If the current segment is not the first segment, then function orient() aligns the haplotype pairs by matching the first 6 SNPs in the current segment to the last 6 SNPs in the last segment, and generates a revised version of maxList, the newList. Function fileOut() then write out the appropriately formatted haplotypes following the example solution.