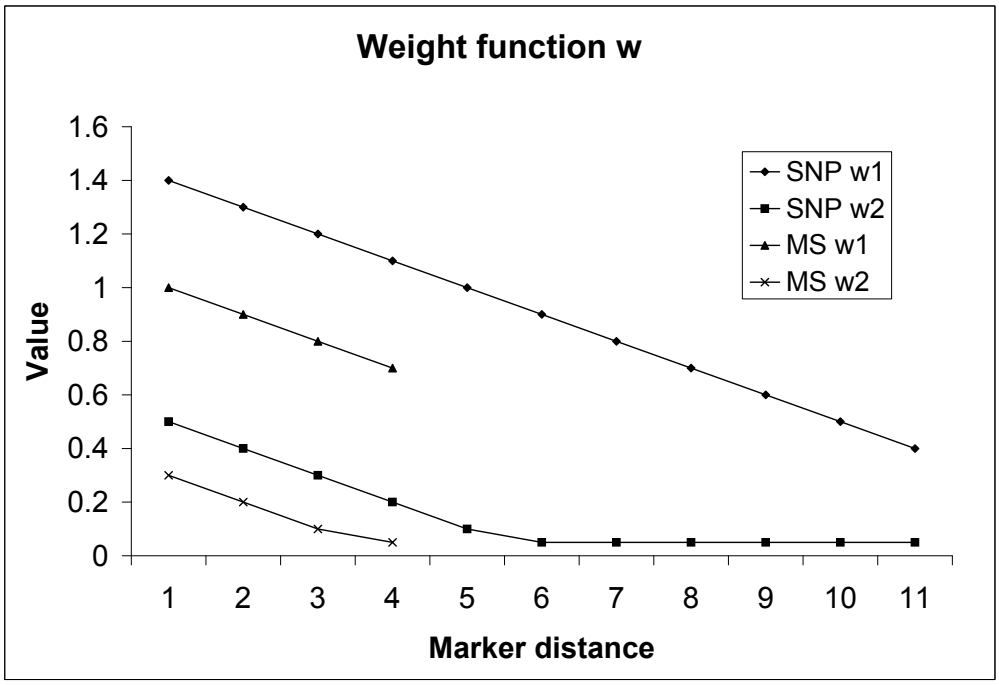

Figure 1: The weighting functions used in our simulation studies of dataset I.

Dataset II: fine mapping of disease genes

The evolutionary model used in generating the dataset II was similar to the model used in a recent paper by Zollner and Pritchard (2005) with slight modification. We assumed that the underlying model is consistent with the coalescence theory and therefore the MS program from Hudson (2002) was used to generate the haplotype data. A haploid population with a constant size of 10,000 chromosomes was simulated with mutation and recombination modelled. The recombination rate was assumed uniformly distributed. The mutation rate was set to be 1E-9 per site per generation and the recombination rate was set to be 1E-8 per site per generation. A candidate region of length one Mb (megabasepairs) was considered. All other parameters of MS were set at their default values. Only markers with minor allele frequencies larger than 0.1 were retained thereafter. In modelling the disease and assigning phenotype to each individual, a SNP was selected from all the SNPs randomly at uniform as the disease mutated marker. A disease was simulated with penetrances of 0.80 (P_{hm}), 0.10 (P_{he}) and 0.05 (P_{hw}) for the homozygous mutant genotype, the heterozygous genotype and the homozygous wild genotype, respectively. We took the same sampling strategy used in Zollner and Pritchard (2005) to select the cases and controls. More specifically, cases and controls were sampled without replacement. Let n be the remaining number of wild-type chromosomes and m the remaining number of mutant chromosomes in the population. Then the next case individual was homozygous for the mutation with probability $P_{hm}m(m-1)(P_{hm}m(m-1)+2P_{he}mn+P_{hw}n(n-1))^{-1}$, heterozygous with probability $2P_{he}mn(P_{hm}m(m-1)+2P_{he}mn+P_{hw}n(n-1))^{-1}$ and otherwise homozygous for the wild allele. The diplotypes for each case were then created by sampling the corresponding number of mutant or wild type chromosomes. Control individuals were generated analogously. The true haplotype assignment of each individual was assumed known.

Results on dataset II

To illustrate the performance of HapMiner on gene fine mapping, we present the results on dataset II involving a genomic region of 1M basepairs with 110 SNP markers, generated via the procedure described earlier. A random SNP was chosen as the disease locus, which is at marker position 62. The frequency of the mutant allele was 0.64, corresponding to the population prevalence of 0.38 based on the penetrance parameters mentioned above. And 100 affected individuals and 100 normal individuals were selected based on the sampling strategy mentioned previously. The phase information was assumed known. We first tested the robustness of HapMiner with respect to the set of parameters. While evaluating one parameter, all the other parameters took their default values based on the consideration of the properties of the input data. The default values of parameters were different from those for dataset I and they were set as follows: the segment length was defined as 15 markers, the weighting functions as exponential and identical functions, the radius as 0.2, and the density threshold as 25% percentile. Figure 2a shows the results for three different segment lengths. As in dataset I, with the increase of the segment length, the score profile tends to be smoother. But the scores near the signal region was rather strong no matter which value we took. We compared two types of weighting functions. The first was an exponential (e^{-10x}) one where x was the genetic distance in Morgan and the second was a constant function ($w_1 = w_2 = 1$) assuming that the markers were evenly spaced. We assumed that 1 cM is equal to 1 Mbps when using the exponential weight functions. HapMiner was robust against these two functions and reported identical results for them (data now shown). We tested three values of ϵ, namely, 0.1, 0.2, and 0.3, respectively. For each ϵ, we first calculated the number of neighbors for every haplotype and chose *MinPts* based on the user-specified percentile parameter. The experimental results in Figure 2b and 2c show that HapMiner performed consistently well across different values of ϵ and *MinPts*. We fixed ϵ to be 0.2 and the percentile for *MinPts* to be 0.25 while testing other parameters thereafter.

To evaluate the accuracy of the proposed method, we compared our haplotype based method with the single marker based method. Results (Figure 2d) from the single marker method show that the scores of adjacent markers might differ drastically thus it is difficult to interpret them. While HapMiner gave more consistent results, especially in the region around the mutated allele. In real data analysis, the functional allele may not be typed. To test the performance of the method in the situation where the DS allele is missed, we deleted the DS marker and tested the algorithm again on the same dataset. Almost identical results (Figure 2e) were obtained, which imply that HapMiner could provide accurate estimation based solely on linkage disequilibrium. For all the cases we have tested, the predicted locations were rather accurate with errors less than three markers away from the true location. Permutation tests were also performed on the same dataset with default parameters for 1,000 iterations. The results in Figure 2f show that the prediction was significant with an empirical p-value less than 0.001.

We further compared HapMiner with CLADHC (and the χ^2 test for single SNP) in terms of the significant level near the disease marker (Figure 3a). The x coordinate represents the marker positions and the y coordinate represents the significant levels, and they cross at the disease marker position on x. The overall significant level p for the whole region was 0.05. The value ($y_{i,j}$) at marker i for method j was defined as $y_{i,j} = (-log(p_{i,j})) - (-log(p)/n_j)$, where $p_{i,j}$ is the significant value of method j at position i (both HapMiner and the single SNP method use the χ^2 test with 1 df; CLADHC uses the likelihood ratio test), and n_j is the number of total multiple tests

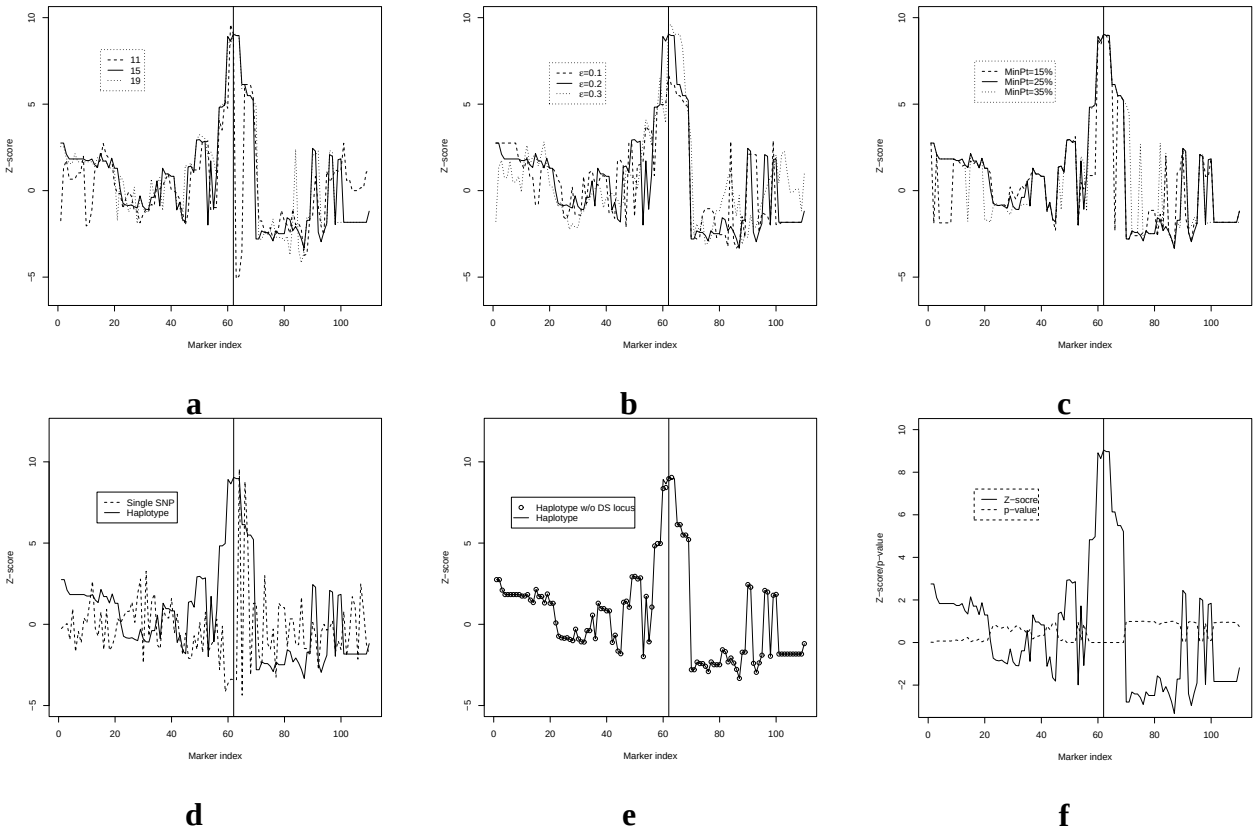

Figure 2: Results on dataset II. HapMiner is robust with respect to several parameters: haplotype segment length (a), ϵ (b), and *MinPts* (c). HapMiner is more consistent than the single SNP based method (d), even when the functional allele marker is deleted (e). Its the result is significant when performing permutation tests (f).

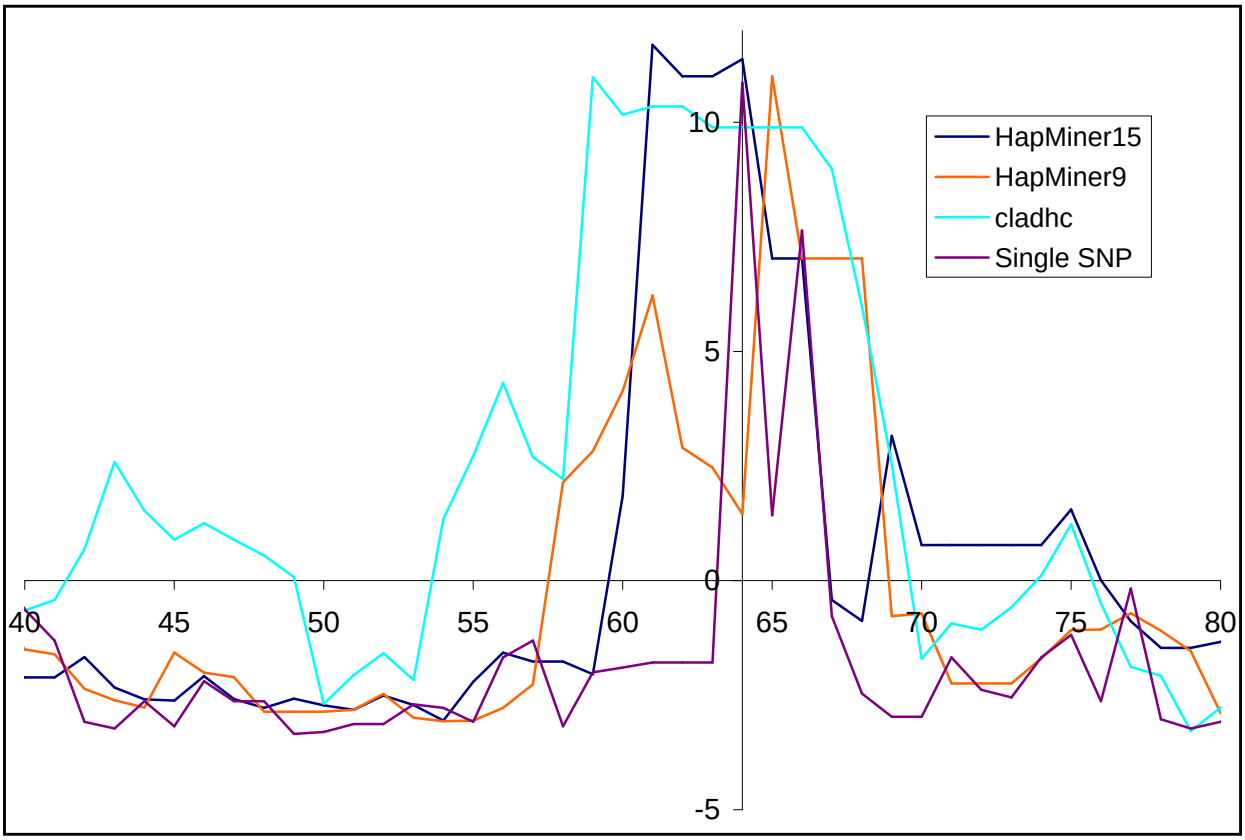

Figure 3: Point estimations on dataset II: significant levels adjusted by multiple tests.

over the region for method j (CLADHC has two levels of multiple tests, so the number is different from the number of SNPs). So in Figure 3a (as well as in Figure 3b), a marker with a positive value means it is significant at the 0.05 level. For this particular dataset, the signal of the simple χ^2 test at the disease locus is rather strong, but values of nearby markers might differ drastically. The situation was improved for HapMiner with segment length 9 but still existed. CLADHC (with segment length 9) reported a relatively broad region and the marker with highest value shifted from the real position. HapMiner with segment length 15 showed a narrow region with a smaller shift. CLADHC couldnot successfully finish the calculation using the segment length 15. For this dataset, HapMiner with length 15 is more reliable than it with length 9 and the simple χ^2 test, and it is more accurate than CLADHC. The two programs might have different segment lengths since they use different weight functions.

References

Hudson, R.R. 2002. Generating samples under a Wright-Fisher neutral model. *Bioinformatics* 18:337-8.

Zollner, S. and Pritchard, J.K. 2005. Coalescent-based association mapping and fine mapping of complex trait Loci. *Genetics*. **169**:1071-92