1. Suplemental Material

1.1. Tagging SNP selection algorithms

Following the notation in the text, note that maximizing $Pr(D_{NT}|D_T)$ is equivalent to minimizing $Pr(D_T)$, the probability of the set of tags. Therefore to produce a tagging set that approximates T_1^*, we suggest a greedy algorithm that starts by picking the first tag in the same way as the ldSelect algorithm does. Then, the algorithm computes the PAC likelihood of each single SNP together with the first tag, and identifies the single SNP that attains the minimum value. This SNP becomes the second tagging SNP identified. Then, the PAC likelihood of each SNP together with the first and second tagging SNP identified so far is computed. The SNP that minimize the likelihood becomes the third tag. This process is iterated until the desired number of tags have been chosen. In other words, the second tagging SNP selected will be

$$t_2^* = \text{argmin}_j Pr(h_{1(j,t_1^*)}, ..., h_{n(j,t_1^*)}|\rho),$$

and the third tagging SNP selected will be the one that satisfies

$$t_3^* = \text{argmin}_j Pr(h_{1(j,t_1^*,t_2^*)}, ..., h_{n(j,t_1^*,t_2^*)}|\rho),$$

and so on for the following tagging SNPs to be selected. $h_{i(j,k)}$ corresponds to the ith haplotype evaluated at the jth and kth SNPs.

To approximately find the SNPs in T_2^* one does the following. First, compute the difference between the PAC likelihood that excludes one SNP from each haplotype minus the PAC likelihood computed including all SNPs in the haplotypes. If haplotypes are formed with S SNPs, then there are S of such differences. Then, identify the single SNP that attains the minimum difference. This SNP becomes the first tagging SNP identified.

This process is iterated, checking each time that the SNP to be selected is not exceeding an r^2 threshold with any of the SNPs already selected as tagging SNPs. The idea behind this approach is to identify the SNPs that add *little* or no information to the full likelihood and which in general will be SNPs that find many SNPs correlated with itself in the set of all SNPs. Speciffically,

$$s_1^* = \text{argmin}_j l(h_{1(-j)}, ..., h_{n(-j)}|\rho) - l(h_1, ..., h_n|\rho).$$

The second tagging SNP selected will be among the set of SNPs that are not correlated with s_1^* and that satisfy

$$s_2^* = \text{argmin}_j l(h_{1(-j,s_1^*)}, ..., h_{n(-j,s_1^*)}|\rho) - l(h_1, ..., h_n|\rho),$$

and so on for the following tagging SNPs to be selected. A subset of all the SNPs is returned as a list of sorted SNPs. The remaining SNPs (the ones not in the list of the returned sorted SNPs) are by construction in high r^2 with at least one SNP from the list.

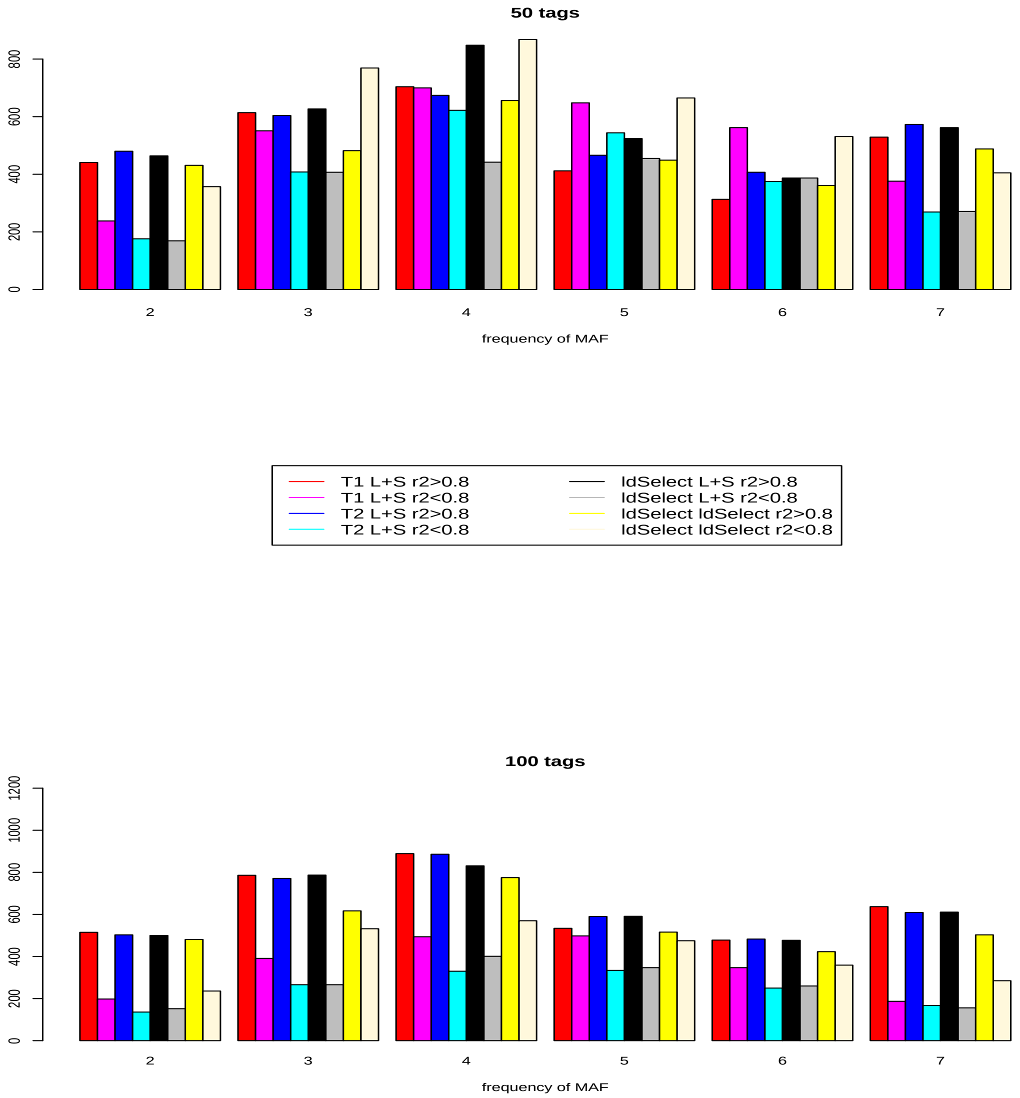

Fig. 1.— Barplots of the frequency of captured and uncaptured SNPs for values of the frequency of the MAF between 2 and 7. These barplots are made up of 8 bars which correspond to the following: the first two bars correspond to the number of non-tags captured by $\hat{T}_1$ tags and our method for predicting non-tags followed by the number of non-tags uncaptured; the third and fourth correspond to the number of non-tags captured by $\hat{T}_2$ tags and our method for predicting non-tags followed by the number of non-tags uncaptured; the fifth and sixth bars correspond to the number of non-tags captured by ldSelect tags and our method for predicting non-tags followed by the number of non-tags uncaptured; the last two columns corresponds to the the number of non-tags capture by ldSelect tags followed by the number of non-tags uncaptured. The upper panel shows the performance of the algorithms using 50 tags and the lower panel the performance of the algorithms using 100 tags.