Supplementary Information

TAGster: Efficient Selection of LD Tag SNP in Single or Multiple Populations

---Algorithms implemented in *TAGster*

Consider a set *S* which contains *M* bi-allelic SNP markers $a_1, a_2 ..., a_M$ in *K* populations $S = \bigcup_{i=1}^{K} S_i$ and S_i contains M_i SNP markers $s_{i1}, s_{i2}, ..., s_{iM_i}$ in population *i*. First, we estimated pairwise LD measure r^2 for each SNP pair within each population. Two markers s_{im} and s_{in} are said to be in strong LD if the $r^2(s_{im}, s_{in})$ is greater than or equal to a pre-specified threshold value r_0. Both are considered tag SNP for each other, in that s_{im} can be used as a surrogate for s_{in}, or vice versa.

Our aim is to find a tag SNP set, denoted by T, such that for $\forall s_{im} \in S_i,\ i = 1,..., K, \quad \exists a_j \in T$ that satisfies $r^2(a_j, s_{im}) \geq r_0$. In our presentation, we introduce intermediate SNP sets, P and Q_i, $i = 1,..., K$. $P = \bigcup_{i=1}^{K} P_i$, where, P_i is called the candidate set which contains all the SNPs in population i that are eligible to be chosen as a tag SNP, Q_i contains SNPs in population i that are already tagged by at least one of tag SNPs in T, i.e. $\forall s_{im} \in Q_i,\ i = 1,..., K, \quad \exists a_j \in T$ that satisfies $r^2(a_j, s_{im}) \geq r_0$. We implemented several algorithms in *TAGster* to select tag SNP set T.

Algorithm 1: A greedy algorithm for single or multiple populations

(1) Set $T = \varnothing$, $P_i = S_i$ and $Q_i = \varnothing$, for any $i = 1,..., K$;

(2) For each SNP a_j in P, calculate

$$C_i(a_j) = \begin{cases} \sum_{m=1, s_{im} \notin Q_i}^{M_i} 1(r^2(a_j, s_{im}) \geq r_0) & \text{if } a_j \in P_i \\ 0 & \text{if } a_j \notin P_i \end{cases}$$

(3) Find the SNP $a_{\max}$ that has the highest $\sum_{i=1}^{K} C_i(a_j)$, and add $a_{\max}$ to T. If $a_{\max} \in P_i$, add any SNP s_{im} in P_i with $r^2(a_{\max}, s_{im}) \geq r_0$ to Q_i and then exclude $a_{\max}$ from P_i;

(4) Repeat Steps 2-3 until $Q_i = S_i$ for any $i = 1,..., K$;

Algorithm 2: An optimal solution for single population tag SNP

An exhaustive Search is performed within each population to find minimal number of population specific tag SNPs T_i for $i = 1,..., K$.

(1) Set $T_i = \varnothing$ and $P_i = S_i$, for $i = 1,..., K$;

(2) Within population i, partition SNPs in P_i into disjoint precinct P_{ij}, $j = 1,..., n$, so that $r^2(s_{im}, s_{in}) < r_0$ for any two SNPs s_{im} and s_{in} that belong to different precincts.

(3) Within a precinct P_{ij},

i. For any two SNPs s_{im} and s_{in} in precinct P_{ij}, if $\sum_{l, s_{il} \in P_{ij}} abs((1(r^2(s_{im}, s_{il}) \geq r_0) - 1(r^2(s_{in,} s_{il}) \geq r_0)) = 0$, we exclude one with smaller $\sum_{l, r^2(s_{ih}, s_{il}) \geq r_0, h=m \text{ or } n} r^2(s_{ih}, s_{il})$ from precinct P_{ij}.

```
            ii. Conduct an exhaustive search to find a set of minimum number of
                tag SNPs for SNPs in precinct $P_{ij}$ and add these tag SNPs into $T_i$;
    (4) Repeat step (3) for each precinct.
```

Algorithm 3: Two-stage solution for multi-populations

```
    (1) Conduct Algorithm 2 within each population to select a set of population specific
        tag SNPs $T_i$ for $i = 1,..., K$;
    (2) Set $T = \varnothing$, $P_i = S_i$ for $i = 1,..., K$;
    (3) For each SNP $t_{ij}$ in $T_i$, find any SNP $s_{im}$ ($s_{im} \in P_i$ and $s_{im} \notin T_i$) that satisfy
        $r^2(t_{ij}, s_{im}) \ge r_0$, and then add them as well as $t_{ij}$ into LD bin $B_{ij}$ and exclude
        them from $P_i$;
    (4) With each LD bin $B_{ij}$, set $T_{ij} = \varnothing$. Find any SNP $s_{im}$ in $B_{ij}$ that satisfy
        $r^2(s_{im}, s_{in}) \ge r_0$ for any SNP $s_{in}$ in $B_{ij}$, and then add $s_{im}$ to $T_{ij}$;
    (5) Set $P = \bigcup_{i=1}^{K} P_i$, $P_i = \bigcup_{j} T_{ij}$. For each SNP $\tau_l$ in $P$, $l = 1,...,|P|$, construct a one
        dimensional array $A_l$ with $K$ elements, where
            $\begin{cases} A_{li} = j & \text{if } \tau_l \in T_{ij} \\ A_{li} = 0 & \text{if } \tau_l \notin P_i \end{cases}$
    (6) Cluster SNPs in $P$ so that any two SNPs $\tau_m$ and $\tau_n$ in a cluster satisfy
        $\sum_{i=1, A_{mi} \neq 0, A_{ni} \neq 0}^{K} abs(A_{mi} - A_{ni}) = 0$;
```

(7) Set $\Psi = \varnothing$. Find one SNP τ_l in each cluster with maximum $\sum_{i=1}^{K} 1(A_{li} \neq 0)$ and add it to Ψ.

(8) Cluster SNPs in Ψ so that any two SNPs τ_m and τ_n in a cluster satisfy

$$\sum_{i=1, A_{mi} \neq 0, A_{ni} \neq 0}^{K} 1(A_{mi} - A_{ni} = 0) > 0;$$

(9) For each cluster, set LD bin set $B = \varnothing$, record the LD bins in each population that can be tagged by any SNP in the cluster to B, and then conduct an exhaustive search to find a minimum set of tag SNPs in the cluster that can tag all LD bins in B. Add this set of SNPs to T.