SNP Selection Algorithms

There are regions on the genome that are in strong LD together (these are called "haplotype blocks"), which allows us to only genotype parts of the genome and impute the rest. Sites that allows for imputation of other sites are "tagSNPs". The purpose of SNP selection algorithms is to find the smallest possible subset of SNPs (tagSNPs) that are associated with the highest possible number of SNPs in the genome. This reduces the cost of genotyping, and provides us a method of data compression. It also makes it easy for scientists to compare genotype between set of individuals by focusing on certain haplotype blocks.
There are two algorithms that we will study in this section:
(1) LD-select algorithms and
(2) Informativeness.

LD-Select Algorithm

Original Publication: https://pubmed.ncbi.nlm.nih.gov/14681826/
This algorithm selects the 'maximally informative set' of common SNPs (common SNPs occur in $\geq$ 10% of the population) that are either directly assayed or indirectly assayed (i.e. exceed a threshold level of association with some other SNPs). For linkage disequilibrium, it uses the r^2 measure because r^2 is directly related to statistical power to detect association with unassayed sites (its relationship with χ^2). The algorithm can be formally defined as:

Given: Genotype for a population of cases and controls
Computes: Allele frequencies and r^2 thresholds that yield a given level of power to detect a disease associated with any common SNP in the gene. Finds groups of SNPs in association with each other.
Outputs: as few tagSNPs as possible such that all polymorphisms above a specified frequency threshold either directly or indirectly exceed a specified level of r^2 with these tagSNPs.

Pseudocode:
Input: SNPs and their frequencies in the population

a) Set MAF threshold (e.g. 10%) and take the SNPs with frequency $\geq$ MAF.

b) Find the single SNP exceeding the r^2 threshold set by the user (e.g. $r^2 \geq= 0.8$) with the maximum number of SNPs above the MAF threshold. Call this SNP the 'maximum informative site' (MIF).

c) Bin all SNP associated with MIF at or above the r^2 threshold (including the MIS SNP).

d) Iterate on the remaning (unbinned) SNPs; find a new MIS and group its associated sites into a new bin.

e) Continue until all SNPs/sites have been binned. If there are SNPs with no associated sites at or above the r^2 threshold, bin them by themselves (create bins with single SNPs for these).

f) In each bin, compute all pairwise r^2 values: all SNPs associating highly ($r^2 \geq 0.8$) with the maxmimum number of other SNPs are declared 'winners', or MAX INFO SITES

g) Select a single tag SNP from each bin (only one of the 'winners' found above), taking into account genomic annotations (coding/non-coding/repeat vs. unique region/etc.) and assay design (preferably easy to assay)

h) Return the selected tagSNPs.

Note: r^2 is not transitive in real data: not all SNPs in the bin are interchangeable because pairwise association is not (generally) transitive

Selection with Informativeness Algorithm

Relevant publications:
Informativeness: https://genome.cshlp.org/content/14/8/1633.full.pdf+html
Directed informativeness: https://link.springer.com/chapter/10.1007/978-3-642-20036-6_42
Note: These publications are available on the course website under "Class Notes" tab.

We are given a set (matrix) of SNPs and a population with one haplotype per individual For example,in the matrix below, columns are SNP loci and rows are individual haplotypes. SNPs are biallelic so for each locus, you label the more common allele (common SNP) as 0 and the less common allele (rare SNP) as 1 and translate this matrix into a binary matrix without losing information as below:

Our goal is to identify SNPs that predict genotype at other loci / has information on the other SNPs, if possible. This is a data compression problem like LD-Select because we want a subset of SNPs that is smaller than the set of all SNPs. In fact, we want the smalles number of SNPs that can explain / contain information on the entire set. This is the **"minimum informative subset (MIS)"** problem. We will achieve this through a graph theoretic approach, where we create *distinguishability edges* between 1s and 0s at each locus. We want to find out how many (and which) SNPs we need to consider to find out all edges.

Hapl./Locus	**1**	**2**	**3**	**4**
1	T	C	A	G
2	T	G	T	G
3	A	C	A	C

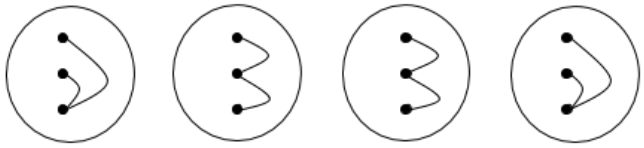
$\Longrightarrow$

Hapl./Locus	**1**	**2**	**3**	**4**
1	0	0	0	0
2	0	1	1	0
3	1	0	0	1

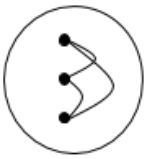
Note: A haplotype here is a series of 0's and 1's corresponding to SNP sites.

The Minimum Informative Subset of SNPs: A subset of SNPs of minimum size that has the complete information of the entire set of SNPs. Of course, complete is one thing- it means 100 percent information- but you can ask partial questions too. How about the minimum set of SNPs that contains 70 percent of the information of the entire set? This percentage can be set by the parameter ξ.
In this problem, the fundamental unit of information is the SNP. It separates two individuals, in the sense that at the SNP, one individual has allele 0, and the other can have allele 1, which distinguishes them. You are always making pairwise comparisons here based on SNPs to separate individuals. For example, in the example above, SNP at locus 1 separates individual (haplotype) no.3 from individual no.1, and SNP3 separates individual no.4 from individual no.3 etc.

Distinguishability edge (D-edge): For each locus, you create as many nodes as haplotypes and you make a D-edge between nodes when they are distinguished from each other (have different values). Nodes here represent individuals. For the example above, this would be:



Our universal set is the imposition of all these 4 sets together:



This is the total information we have and we want to find the minimum subset that can

provide complete information on the edges of the universal set (in this case locus 1 and locus 2 or locus 2 and locus 4 etc will work). Here we define the measure **"informativeness"**:

I(s,t) = information of SNP s about SNP t

$$I(s,t) = \frac{\overline{E(s) \cap E(t)}}{\overline{E(t)}}$$

E(s) = the set of edges in the SNP graph for s.
The intersection is "how many edges in s that are also in r"
$\overline{E(s)}$ is the number of edges in E(s)

Example:

$$I(s1, s2) = \frac{((1,3),(2,3)) \cap ((1,2),(2,3))}{((1,2),(2,3))} = \frac{1}{2}$$

Thanks to this graph theoretic and information theoretic approach, informativeness has unique extension to **multiple loci** unlike LD-select and attempts to break the "curse of pairwise comparisons". If we want to look at three loci, for example, we simply create a joint set of two of the loci and compare it to the third. For example:

$$I(s1, s2, s3) = \frac{(E(s1) \cup E(s2)) \cap E(s3)}{E(s3)} = \frac{[((1,3),(2,3)) \cup ((1,2),(2,3))] \cap [((1,2),(2,3))]}{((1,2),(2,3))}$$

$$I(s1, s2, s3) = \frac{((1,2),(2,3))}{((1,2),(2,3))} = \frac{2}{2} = 1$$

However, note that informativeness is not "symmetrical". If of the two loci considered s, and t, one has more edges than the other, depending on which one is in the denominator, the result will change. In this case I(s,t) might not be equal I(t,s). Similar to r^2,it also tries to achieve a level of interpretability for the intermediary values through its association with the r^2 measure , and thus the χ^2. This is expanded on in your readings for homework 2.

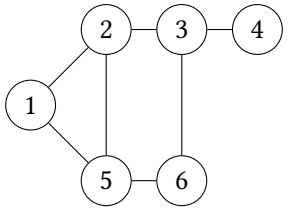
While LD-Select employs a greedy approach with "dominating set" algorithm, Informativeness uses "set cover". **However, "dominating set" and "set cover" are reducible to each other** and this is what we will show below:

Dominating Set and Set Cover

Dominant Set A **dominating set** for a graph $G = (V, E)$ is a subset D of V, $D \subseteq V$, such that every vertex not in D is joined to at least one vertex of D by an edge.[1] The *dominating number* $\gamma(G)$ is the smallest number of vertices in a dominating set of G.
Given: $G = (V, E)$; some constant k
Compute: Dominating set D for G such that $|D| \leq k$ (if it exists)

The Set cover
Given: An universe set U and a collection of subsets of U $\{S_1, \ldots, S_k\}, S_i \subseteq U \, \forall \, i$
Compute: The minimum number of subsets that cover U.(i.e. $\bigcup_{i=1}^{l} S_i = U$. Example: given a mapping between boys and girls that they know, what is the minimum number of boys that collectively know all the girls? This is NP-complete, as well.)

Example: $U = \{1, 2, 3, 4, 5, 6\}$;
$G =$



$D = \{3, 5\}$ is the smallest dominating set.
$S_3 = \{2, 3, 4, 6\}$ and $S_5 = \{1, 2, 5, 6\}$ together form a set cover: $S_3 \cup S_5 = U$.

Result:

$$\text{Dominating Set of } (G) \leftrightarrow \text{Set Cover}(U, \{S_i\})$$

Dominating Set and Set Cover Problems are Reducible to Each Other

Reduction 1: From Dominating Set to Set Cover Given graph $G = (V, E)$, $V = \{1...n\}$, we construct a set cover problem as follows:
$U = V$, the family of subsets $S = \{s_1, s_2, ..., s_n\}$ such that S_v consists of the vertex v and all vertices adjacent to v.

Suppose D is a dominating set in G. Then $C = \{S_v | v \in D\}$ is a set cover for (U, S) and C and D are the same size ($|D| = |C|$). Conversely, if $C = \{S_v | v \in T\}$ is a set cover for (U, S), then T is a dominating set for G, with $|T| = |C|$.

[1] Symbolically: $\forall \, v \notin D, \exists \, v' \in D$ s.t. $(v, v') \in E$

Reduction 2: From Set Cover to Dominating Set Let (U, S) be a general problem for set cover, U =universe, $S = \{s_i : i \in I\}$ $I \cap U = \varnothing$. Now construct $G = (V, E)$ as follows:
Edges: $\{i, j\} \in E; \forall i, j \in I; \{i, u\} \in Ei \in I; u \in S_i$
Vertices: $V = U \cup I$
Basically, everything is a vertex, and we use the sets to find edges, plus we connect all the sets to each other.

a) C is a set cover of (U, S), $C = \{s_i : i \in D\}$, $D \leq I$, then D is a dominating set.
b) Conversely, let D be a dominating set for G, then it is possible to construct another dominating set X such that $|X| \leq |D|$, $X \leq I$. Simply replace each $u \in D \cap U$ by a neighbor $i \in I$ of u. Then see the set of subsets such that $C = \{s_i : i \in X\}$ is a feasible solution to set cover $|X| = |C| = |D|$.