Linkage Disequilibrium: Lab 2

Introduction

In the previous lab, we performed linkage disequilibrium calculations of r^2 on haplotype data. In a perfect world we would always have haploptype data for each study participant. Unfortunately, it is much easier and common to obtain genotype data. A problem emerges in that if we consider two sites on the same chromosome, and denote the major and minor alleles at the first chromosome as A and a, and likewise as B and b at the second site. If an individual has the genotype $AaBb$, we cannot distinguish between haplotype combinations (Ab, aB) or (AB, ab). Therefore, in order to calculate LD using genotype data, an extra step is need to estimate haplotypes given genotype data.

Linkage Disequilibrium with Genotype Data

In this lab we will use the `genetics` package in order to perform haplotype estimation and estimate r^2.

```
install.packages("genetics")
require(genetics)
```

Next, let's load some genotype data. `geno1` contains genotype data for 2096 individuals for the same 3586 sites as the `hap1` object. The data is in numeric format, where the value pertains to the number of minor alleles present in each genotype. We will need to convert this to the `genotype` format, using the `as.genotype.allele.count()` function. Let's first find the r^2 value for the sites from columns 999 & 1000.

```
load("genotypes_chr1.RData")
genosite1 <- as.genotype.allele.count(geno1[, 999])
genosite2 <- as.genotype.allele.count(geno1[, 1000])
```

It's simple to find X^2 and N using the `LD` function. Since the n output from `LD()` pertains to population size, we will double to find the number of homologous chromosomes.

```
LD(genosite1, genosite2)
X2geno <- LD(genosite1, genosite2)$"X^2"
twoN <- 2 * LD(genosite1, genosite2)$n
r2geno <- X2geno/twoN
r2geno
```

`LD()` will can also find r^2 directly.

```
LD(genosite1, genosite2)$"R^2"
```

Notice, that the r^2 value for the genotype data is much smaller than that for the haplotype data. One reason is we can't expect the same LD values between two different populations. Another reason, is the nature of the way the haplotypes are estimated for the genotype data, which adds uncertainty to our estimates.

Exercise 1

Compare the following r^2 values to the r^2 values we found in lab 1 for these site combinations:

1. sites from columns 75 and 76
2. sites from columns 75 and 77
3. sites from columns 75 and 80

LD Maps

In order to create LD maps, similar to the one's we found in Lab 1, we first need to convert the numeric data for each site using the `as.genotype.allele.count()` function. We can do this column by column, or use another `apply` function, and then format the data once more using the `makeGenotype()` function. This is demonstrated below for sites from columns 1-5.

```
convertgenos <- function(start, end) {
    as.genos <- function(x) {
        as.genotype.allele.count(geno1[, x])
    }
    siteS <- as.matrix(c(start:end))
    sitekeep <- which(!colSums(geno1[, start:end], na.rm = TRUE) == 0)
    siteS <- as.matrix(siteS[sitekeep, ])
    genos <- (apply(siteS, 1, as.genos))
    colnames(genos) <- colnames(geno1)[which(!colSums(geno1[, start:end], na.rm = TRUE) ==
        0)]
    genos <- makeGenotypes(genos)
}

genos <- convertgenos(1, 5)
```

Exercise 2

We can now use the `LDheatmap()` function to create heat maps of r^2 values. The use of this function is demonstrated below:

```
install.packages("LDheatmap")
require(LDheatmap)
LDheatmap(genos, LDmeasure = "r", SNP.name = colnames(genos))
```

1. Create a new genotype object of the first 50 sites from the `geno1` object and create a heat map from this object. *Both steps may take a little while.*
2. What pairs of sites appear to have the greatest LD?
3. Are there groups of sites that you notice to have higher LD than others? Between which sites does these occur?

   You can also use the `LD()` function to create LD maps of r^2 values.

   ```
   LD(genos)$"R^2"
   ```

4. What pair of sites have the greatest LD?
5. What is the average LD of the first 50 sites?

Suppose we are interested in finding areas of high LD without looking looking at either LD map. One way this can be performed is by taking a moving average of the r^2 values across a subset of sites. In order to do this I've created the `avgld` function. This function takes 4 arguments:

- `genos`: The `genotype` object.
- `start`: The column containing the first site in the `genotype` object.
- `end`: The column containing the last site in the `genotype` object.
- `length`: The length of the subset of sites we wish to find the average LD for.

The function will find the average LD for the first `length` sites and then move up one site and do it again, until it reaches the `end` site.

The output will be a matrix with three columns:

- "AvgLD": The average LD for each subset of sites.
- "Start": The start position for each subset of sites.
- "Stop": The stop position for each subset of sites.

```
avgld <- function(genos, start, end, length) {
    lo <- start:(end - length + 1)
    hi <- length:end
    vec <- rbind(lo, hi)
    avgld <- function(vec) {
        start <- vec[1]
        end <- vec[2]
        genomat <- genos[start:end]
        mat <- LD(genomat)$"R^2"
        avg <- mean(mat, na.rm = TRUE)
        cbind(avg, start, end)
    }
    avgLD <- t(apply(vec, 2, avgld))
    colnames(avgLD) <- c("AvgLD", "Start", "Stop")
    avgLD
}
```

6. Plot the moving average of the first 50 sites, with a window length of 10? Is it consistent with your results from *question 2*?