Linkage Disequilibrium: Lab 1

Introduction

Consider two sites on the same chromosome. At each site there are two possible alleles. We can denote the major and minor alleles at the first site as A and a, and likewise as B and b at the second site. The probability of each allele occurring is denoted by P_A, P_a, P_B, and P_b. Assuming independence we can model the probability of any haplotype, AB, Ab, aB, and ab as:

	B	b	
A	$P_A P_B$	$P_A P_b$	P_A
a	$P_a P_B$	$P_a P_b$.	P_a
	P_B	P_b	1

If we are measuring the haplotype counts from n individuals, each with two homologous chromosomes, then we have a distribution of expected haplotype counts from sample size $N = 2n$:

	B	b	
A	$NP_A P_B$	$NP_A P_b$	NP_A
a	$NP_a P_B$	$NP_a P_b$.	NP_a
	NP_B	NP_b	$N = 2n$

However, under linkage disequilibrium there is a departure D from this distribution of haplotype counts, which can be represented by:

	B	b	
A	$N(P_A P_B + D)$	$N(P_A P_b - D)$	NP_A
a	$N(P_a P_B - D)$	$N(P_a P_b + D)$.	NP_a
	NP_B	NP_b	$N = 2n$

r^2: A measure of LD

r^2 is a commonly used measure of LD. It is relatively straightforward to calculate based on haplotype data, using Pearson's χ^2-statistic. This measures the departure of the observed distribution of haplotype counts from the expected distribution of haplotype counts under independence. It has the form:

$$r^2 = \chi_1^2 / N.$$

Remember that N is the total number of homologous chromosomes, not necessarily the number of individuals. The χ^2-statistic has the form

$$\chi_1^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})}{E_{ij}},$$

where i and j refer to rows and columns of the previous tables for the expected and observed haplotype counts.

Linkage Disequilibrium with Haplotype Data

Now, lets check out some real data! Luckily, due to the efforts of the HapMap and 1000 Genomes projects, there is plenty of readily available haplotype data. Let's start by uploading haplotype data from HapMap Phase 2. The data set contains single haplotypes from 60 CEU (Northern Europeans from Utah) individuals for 3586 sites on chromosome 1. It will be saved to your workspace as `hap1`.

```
load("haplotypes_chr1.RData")
```

The sites are ordered by position on the chromosome. In order to pull out the data for two relatively close sites, we pull out two adjacent columns from the `hap1` object.

```
site1 <- hap1[, 999]
site2 <- hap1[, 1000]

table(site1)
```

```
## site1
##  A  B
## 40 19
```

```
table(site2)
```

```
## site2
##  A  B
## 37 21
```

Note, that contrary to the previous example, A and B pertain to the two possible alleles at each site. This notation choice can be confusing, however it is consistent with common practices and programming formats. For example the same formatting is used by the `genetics` package in the next lab.

Using the `chisq.test()` function we can demonstrate the expected and observed distribution of haplotype counts.

```
chisq.test(site1, site2)$expected
```

```
##      site2
## site1     A      B
##     A 24.63 14.368
##     B 11.37  6.632
```

```
chisq.test(site1, site2)$observed
```

```
##      site2
## site1  A  B
##     A 36  3
##     B  0 18
```

Notice, that the values for `expected` and `observed` counts do not match, and are actually fairly far apart.

Due to missing data at either site we will need to set N as the sum of haplotype counts in which data for both sites is present.

```
N <- sum(chisq.test(site1, site2)$observed)
```

So, $N = 57$. Now we can calculate χ^2_1.

```
X2 <- chisq.test(site1, site2, correct = FALSE)$statistic
X2
```

Now, we only need to divide this by N to get r^2.

```
r2 <- X2/N
r2
```

We have $r^2 = 0.79$. So we can say that these two sites have high LD. It is important to point out that r^2 will always be between 0 and 1, with perfectly correlated sites having $r^2 = 1$ and completely independent sites having $r^2 = 0$.

Exercise 1

1. Pull out the site combination from sites 75 & 76 from the `hap1` object.
   - Demonstrate the expected and observed haplotype distributions for these two sites?
   - What is the value of χ^2_1 and r^2?
   - What can you say about the LD of these two sites?
2. Find the r^2 values of on site combination from columns 75 & 77, and 75 & 80 of the `hap1` object. What do you notice about the LD relationship between these two combinations (recall that they are in order by position)?

LD Maps

LD maps are triangular matrices demonstrating the LD between a number of ordered sites.

We can use loop functions like `apply()` to build larger functions to run the previous algorithm recursively in order to build LD maps.

Check out the `ld.Rsq()` function.

```
ld.Rsq <- function(hapmat) {
    numvec <- as.matrix(c(1:ncol(hapmat)))
    ld <- function(hapcol, hapmat) {
        x1 <- hapmat[, hapcol]
        rsq <- function(z) {
            N <- sum(chisq.test(x1, z)$observed)
            chisq.test(x1, z, correct = FALSE)$statistic/N
        }
        apply(hapmat, 2, rsq)
    }
    map <- apply(numvec, 1, ld, hapmat)
    colnames(map) <- rownames(map)
    bot <- upper.tri(map, diag = TRUE)
    map[bot] <- NA
    signif(map, 4)
}

hapmat <- hap1[, 1:5]
ldmap <- ld.Rsq(hapmat)
ldmap
```

The lower triangle of this matrix represents the r^2 values of each combination of sites.

Exercise 2

1. Run `ld.Rsq()` on sites from columns 75-80 from the `hap1` object and compare the map with your values from Exercise 1. We can use the `ggplot2` and `reshape` packages to build heat maps in R. Heat maps are matrices in which value is demonstrated via a color gradient rather than the number. There are many ways this can be done, one of which is demonstrated via the `heatmapHaps()` function below. Here, we can input our LD map from the `ld.Rsq()` output in order to generate the LD heat map:

```
install.packages("reshape")
install.packages("ggplot2")
require(reshape)
require(ggplot2)

heatmapHaps <- function(hapmap) {
    meltedr2 <- melt(hapmap)

    meltedr2$X1 <- factor(meltedr2$X1, as.character(meltedr2$X1))
    meltedr2$X2 <- factor(meltedr2$X2, as.character(meltedr2$X2))

    p <- ggplot(meltedr2, aes(X1, X2)) + geom_tile(aes(fill = value), colour = "white") +
        scale_fill_gradient(low = "white", high = "steelblue", name = "r^2") +
        xlab(NULL) + ylab(NULL) + ggtitle("LD Measured by r^2")
    p
}
heatmapHaps(ldmap)
```

2. Build a new LD map of the first ten SNPs, and then input it into the `heatmapHaps() function`. What site appears to have the highest average LD with the other sites in the LD map? Which pairs of sites appear to have the highest LD? Which of these pairs actually has the highest LD?