LD vignette
Measures of linkage disequilibrium

David Clayton

October 17, 2016

Calculating linkage disequilibrium statistics

We shall first load some illustrative data.

```
> data(ld.example)
```

The data are drawn from the International HapMap Project and concern 603 SNPs over a 1mb region of chromosome 22 in sample of Europeans (`ceph.1mb`) and a sample of Africans (`yri.1mb`):

```
> ceph.1mb

A SnpMatrix with  90 rows and  603 columns
Row names:  NA06985 ... NA12892
Col names:  rs5993821 ... rs5747302

> yri.1mb

A SnpMatrix with  90 rows and  603 columns
Row names:  NA18500 ... NA19240
Col names:  rs5993821 ... rs5747302
```

The details of these SNP are stored in the dataframe `support.ld`:

```
> head(support.ld)

          dbSNPalleles Assignment Chromosome Position Strand
rs5993821          G/T        G/T      chr22 15516658      +
rs5993848          C/G        C/G      chr22 15529033      +
rs361944           C/G        C/G      chr22 15544372      +
rs361995           C/T        C/T      chr22 15544478      +
rs361799           C/T        C/T      chr22 15544773      +
rs361973           A/G        A/G      chr22 15549522      +
```

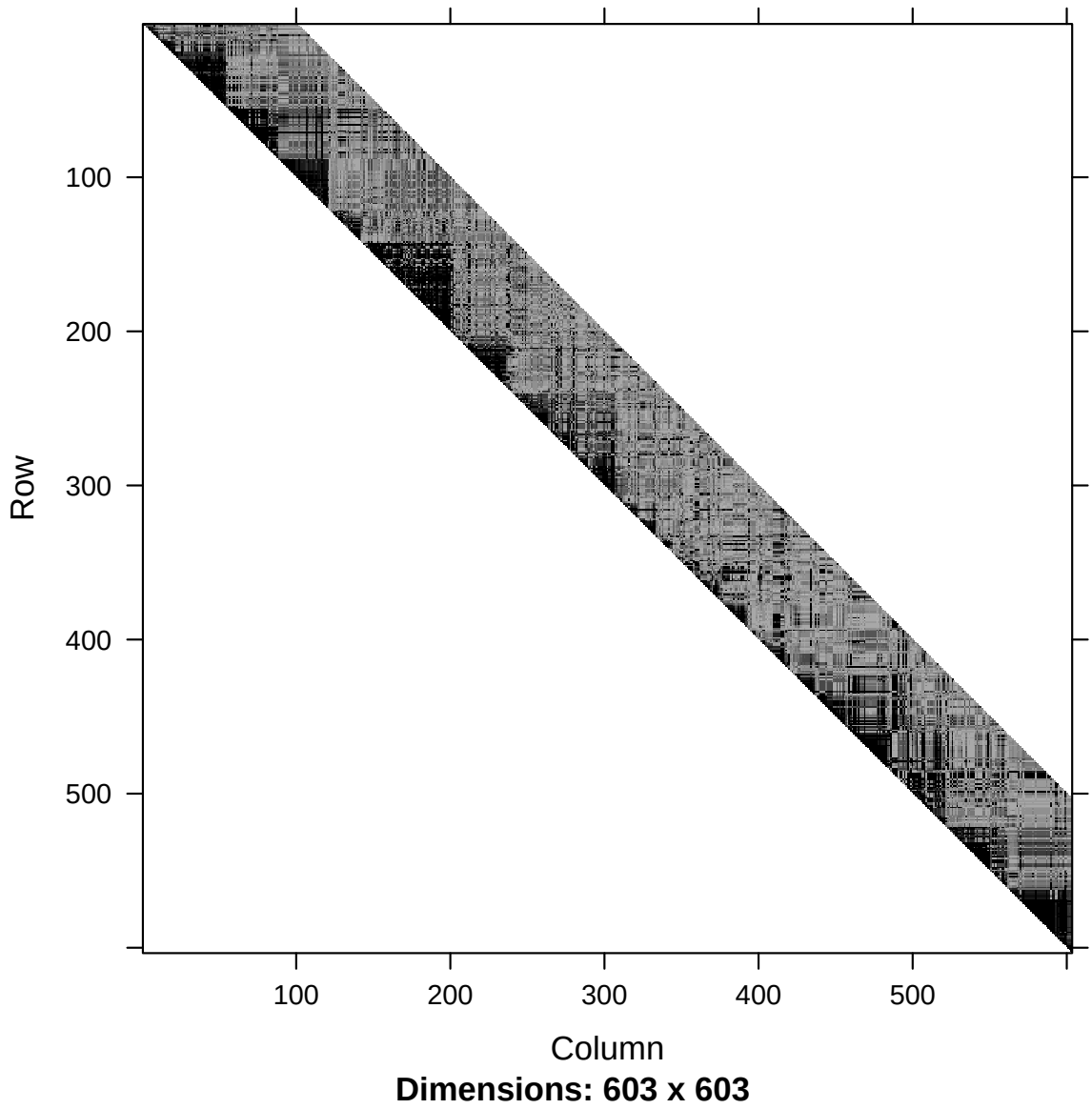
The function for calculating measures of linkage disequilibrium (LD) in `snpStats` is `ld`. The following two commands call this function to calculate the D-prime and R-squared measures of LD between pairs of SNPs for the European and African samples:

```
> ld.ceph <- ld(ceph.1mb, stats=c("D.prime", "R.squared"), depth=100)
> ld.yri <- ld(yri.1mb, stats=c("D.prime", "R.squared"), depth=100)
```

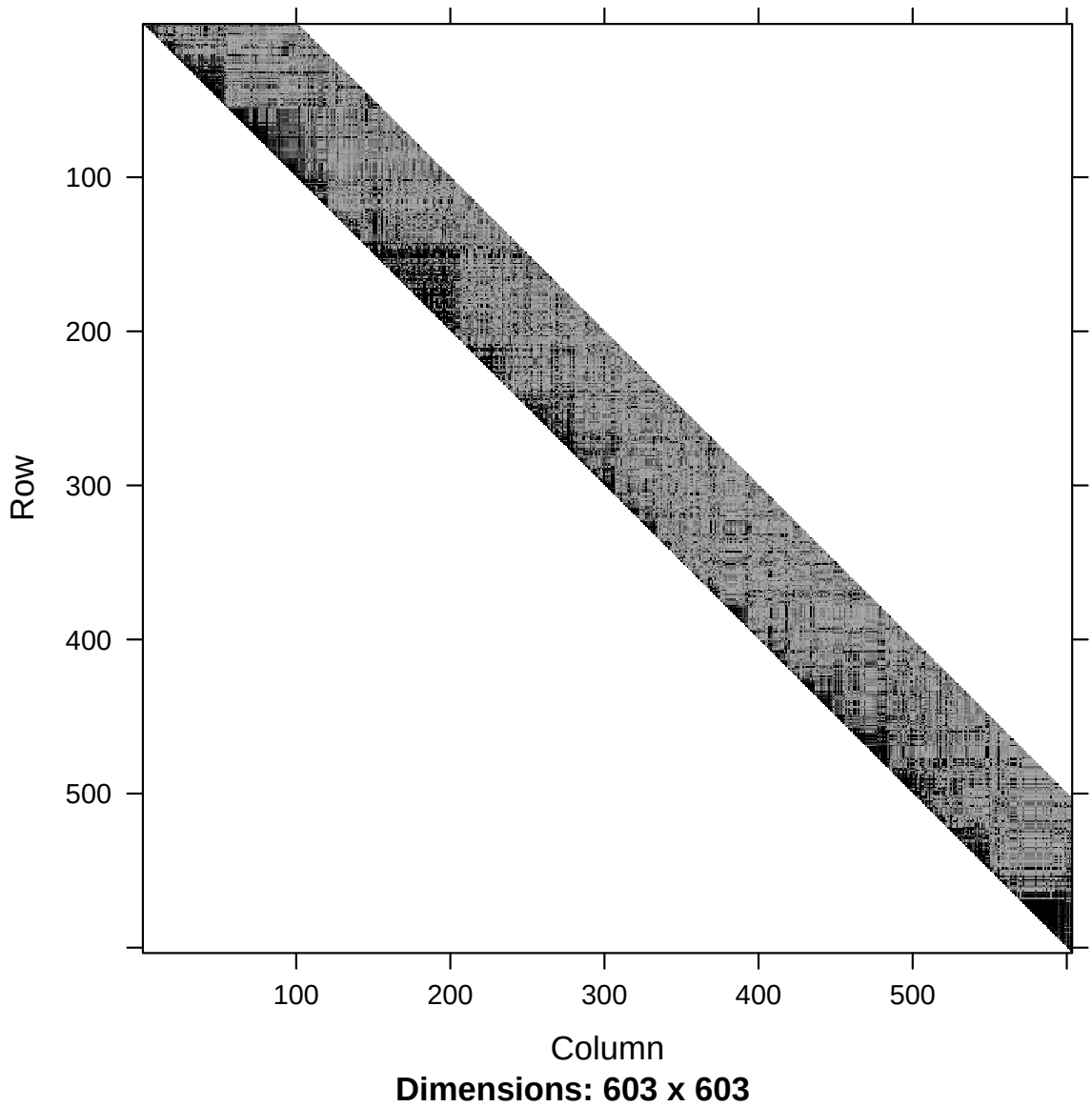
The argument `depth` specifies the maximum separation between pairs of SNPs to be considered, so that `depth=1` would have specified calculation of LD only between immediately adjacent SNPs.

Both `ld.ceph` and `ld.yri` are lists with two elements each, named `D.prime` and `R.squared`. These elements are (upper triangular) band matrices, stored in a packed form defined in the `Matrix` package. They are too large to be listed, but the `Matrix` package provides an `image` method, a convenient way to examine patterns in the matrices. You should look at these carefully and note any differences.

```
> image(ld.ceph$D.prime, lwd=0)
```




```
> image(ld.yri$D.prime, lwd=0)
```



The important things to note are

1. there are fairly well-defined "blocks" of LD, and
2. LD is more pronounced in the Europeans than in the Africans.

The second point is demonstrated by extracting the D-prime values from the matrices (they are to be found in a slot named `x`) and calculating quartiles of their distribution:

```
> quantile(ld.ceph$D.prime@x, na.rm=TRUE)

        0%       25%       50%       75%      100%
0.0000000 0.1284448 0.2966491 0.6373796 1.0000000
```

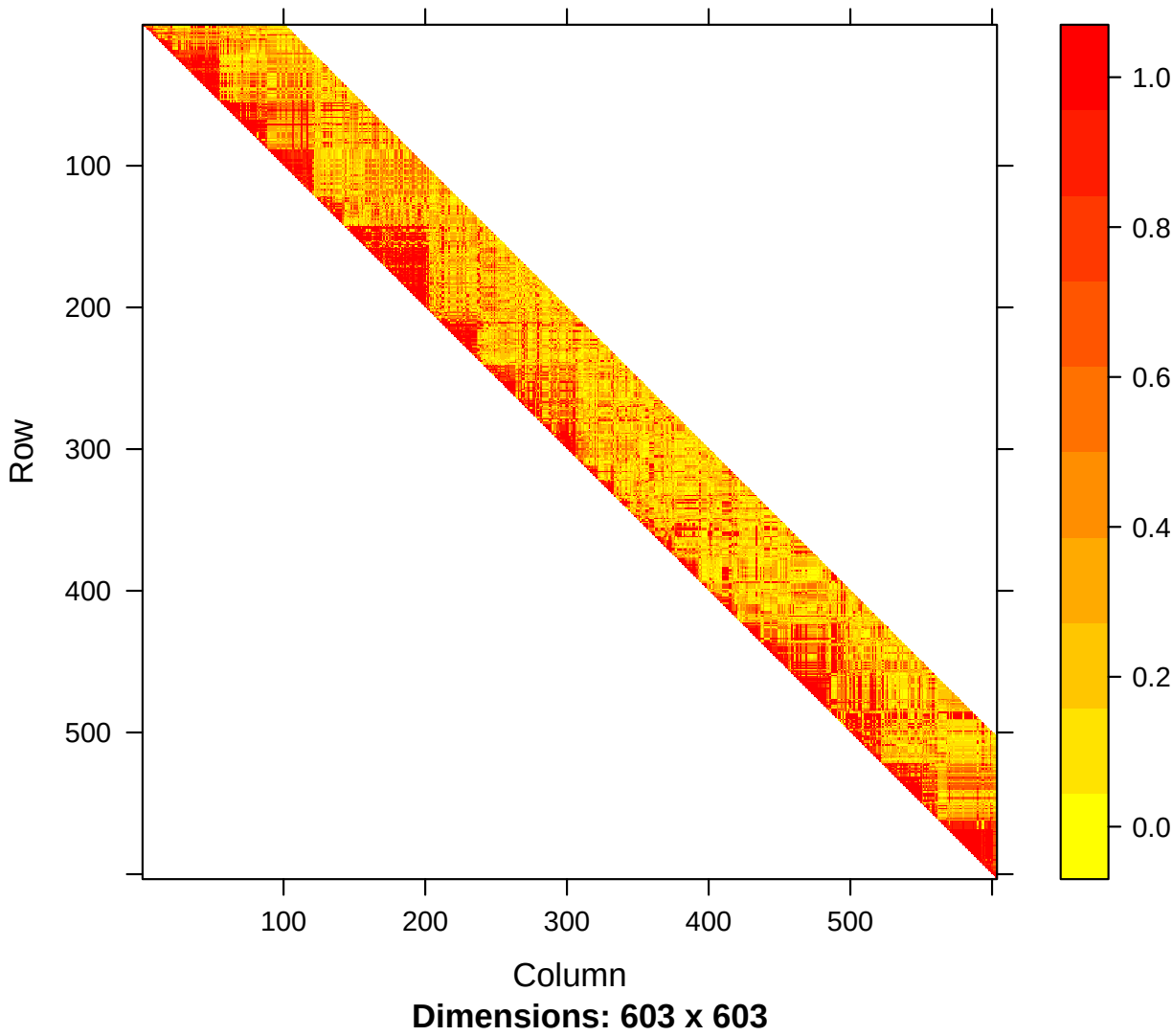
```
> quantile(ld.yri$D.prime@x, na.rm=TRUE)

       0%       25%       50%       75%      100%
0.0000000 0.1066341 0.2438494 0.5098348 1.0000000
```

If preferred, `image` can produce colour plots. We first create a set of 10 colours ranging from yellow (for low values) to red (for high values)

```
> spectrum <- rainbow(10, start=0, end=1/6)[10:1]
```

and plot the image, with a colour key down its right hand side

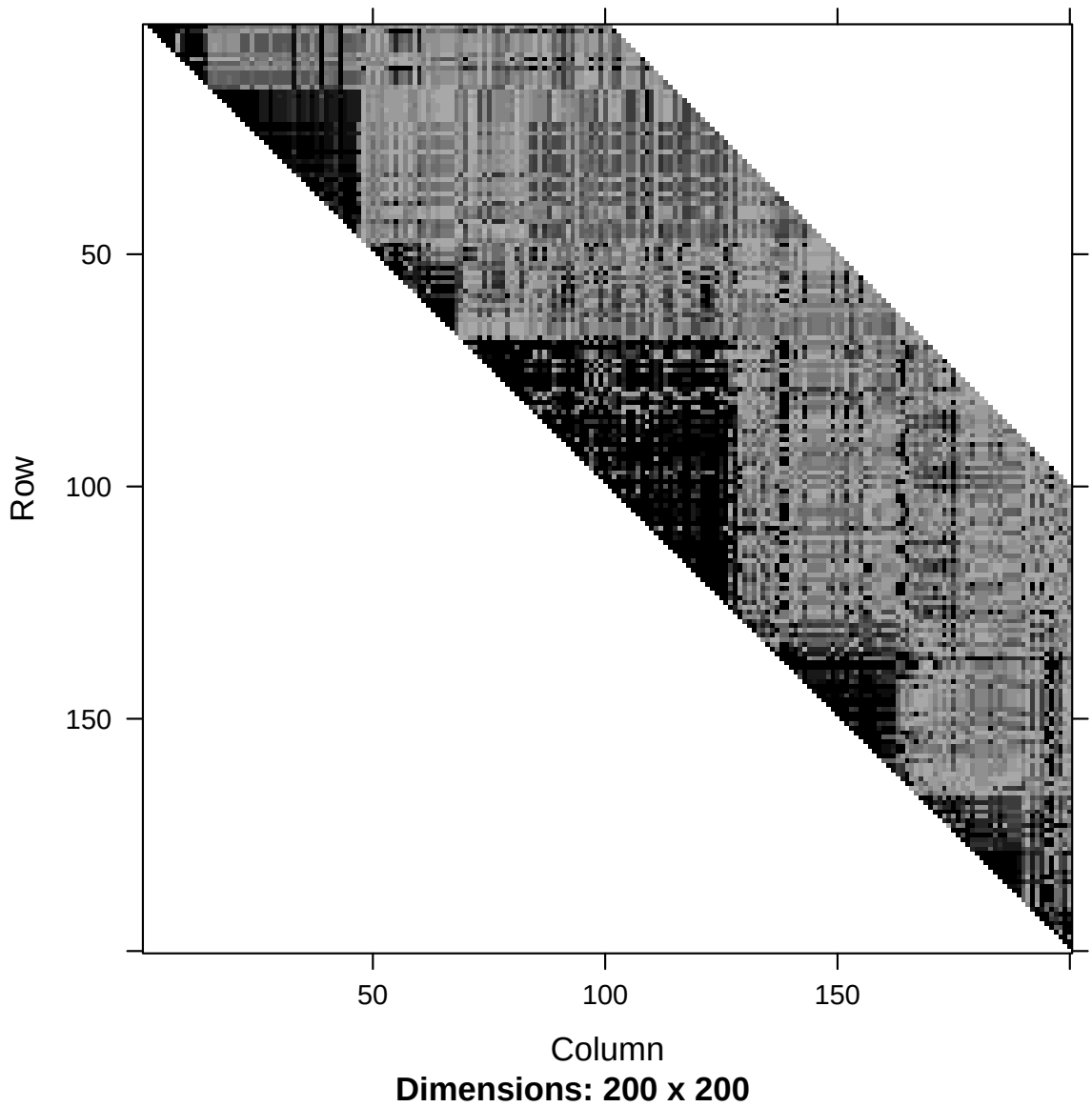
```
> image(ld.ceph$D.prime, lwd=0, cuts=9, col.regions=spectrum, colorkey=TRUE)
```



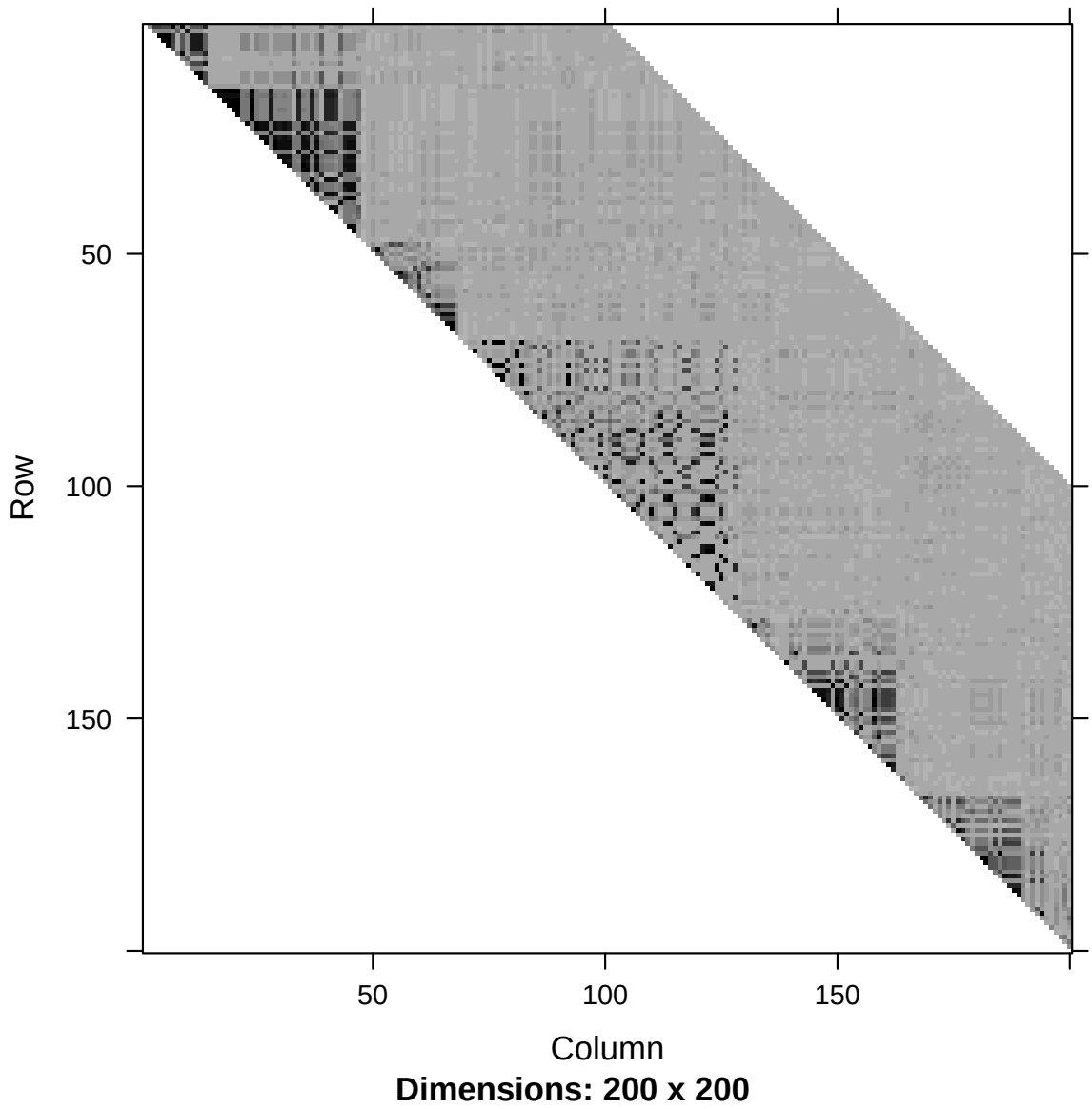
The R-squared matrices provide similar pictures, although they are rather less regular. To show this clearly, we focus on the 200 SNPs starting from the 75-th, using the European data

```
> use <- 75:274
```

```
> image(ld.ceph$D.prime[use,use], lwd=0)
```




```
> image(ld.ceph$R.squared[use,use], lwd=0)
```



The R-squared values are smaller and there are "holes" in the LD blocks; SNPs within an LD block do not necessarily have large R-squared between them. This is further demonstrated in the next section.

D-prime, R-squared, and distance

To examine the relationship between LD and physical distance, we first need to construct a similar matrix holding the physical distances. This is carried out, by first calculating each off-diagonal, and then combining them into a band matrix

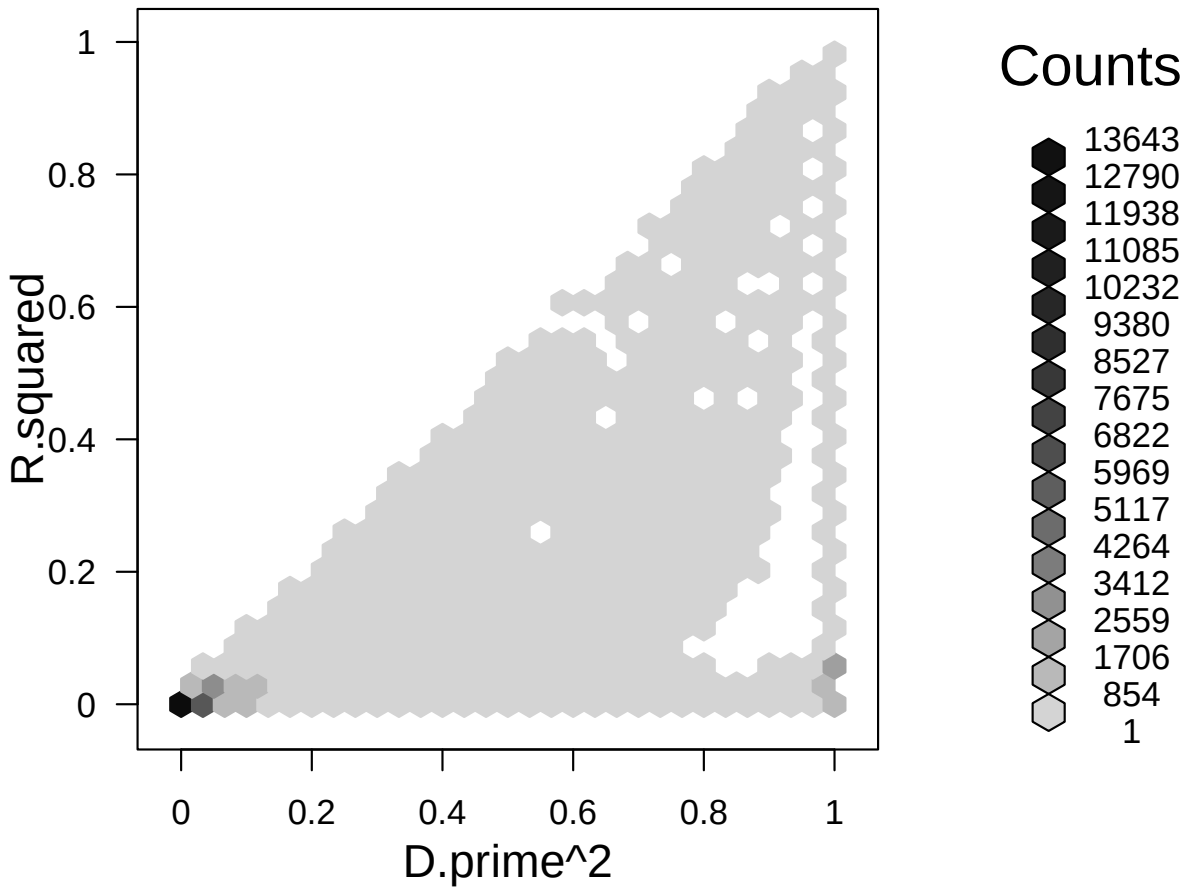
```
> pos <- support.ld$Position
> diags <- vector("list", 100)
> for (i in 1:100) diags[[i]] <- pos[(i+1):603] - pos[1:(603-i)]
> dist <- bandSparse(603, k=1:100, diagonals=diags)
```

The values in the body of the band matrix are contained in a slot named `x`, so the following commands extract the physical distances and the corresponding LD statistics for the Europeans:

```
> distance <- dist@x
> D.prime <- ld.ceph$D.prime@x
> R.squared <- ld.ceph$R.squared@x
```

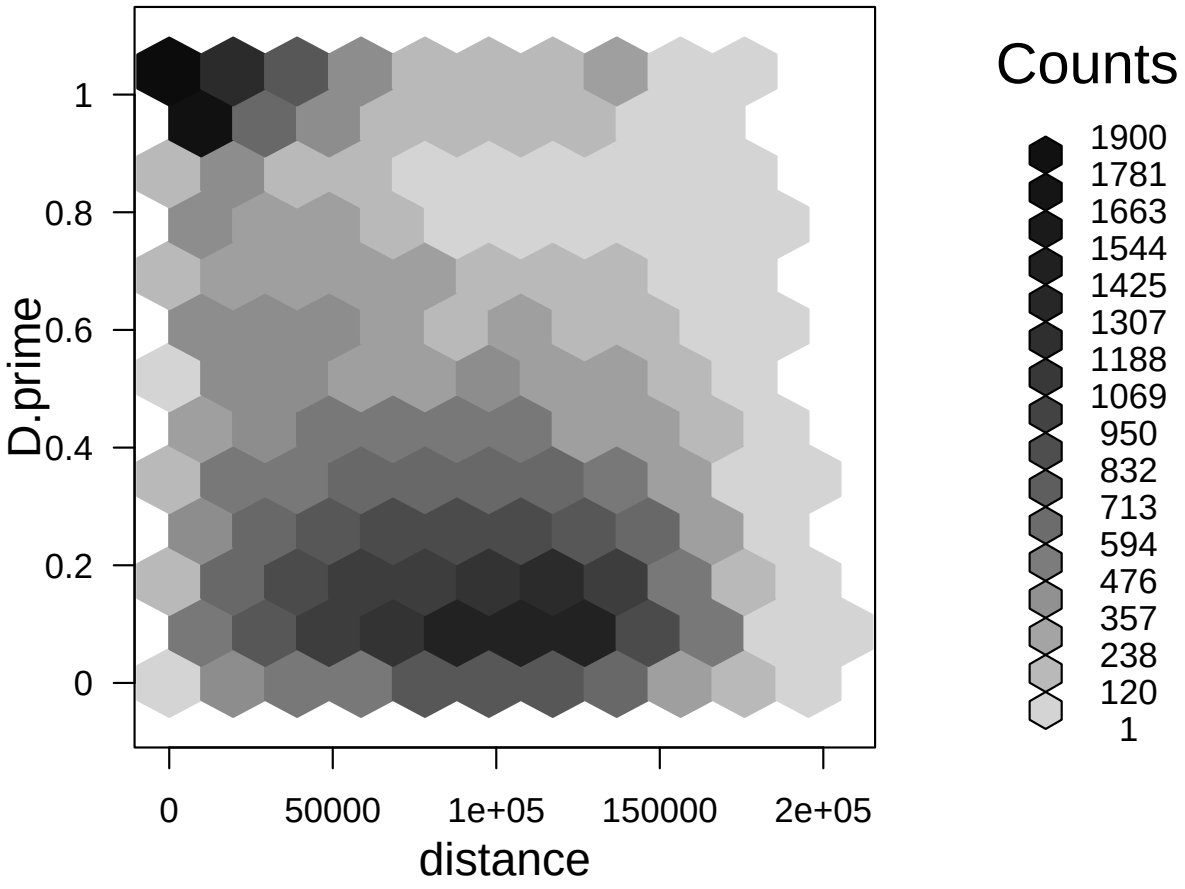
These are very long vectors so we use the `hexbin` package to produce abreviated plots. We first demonstrate the relationship between D-prime and R-squared

```
> plot(hexbin(D.prime^2, R.squared))
```

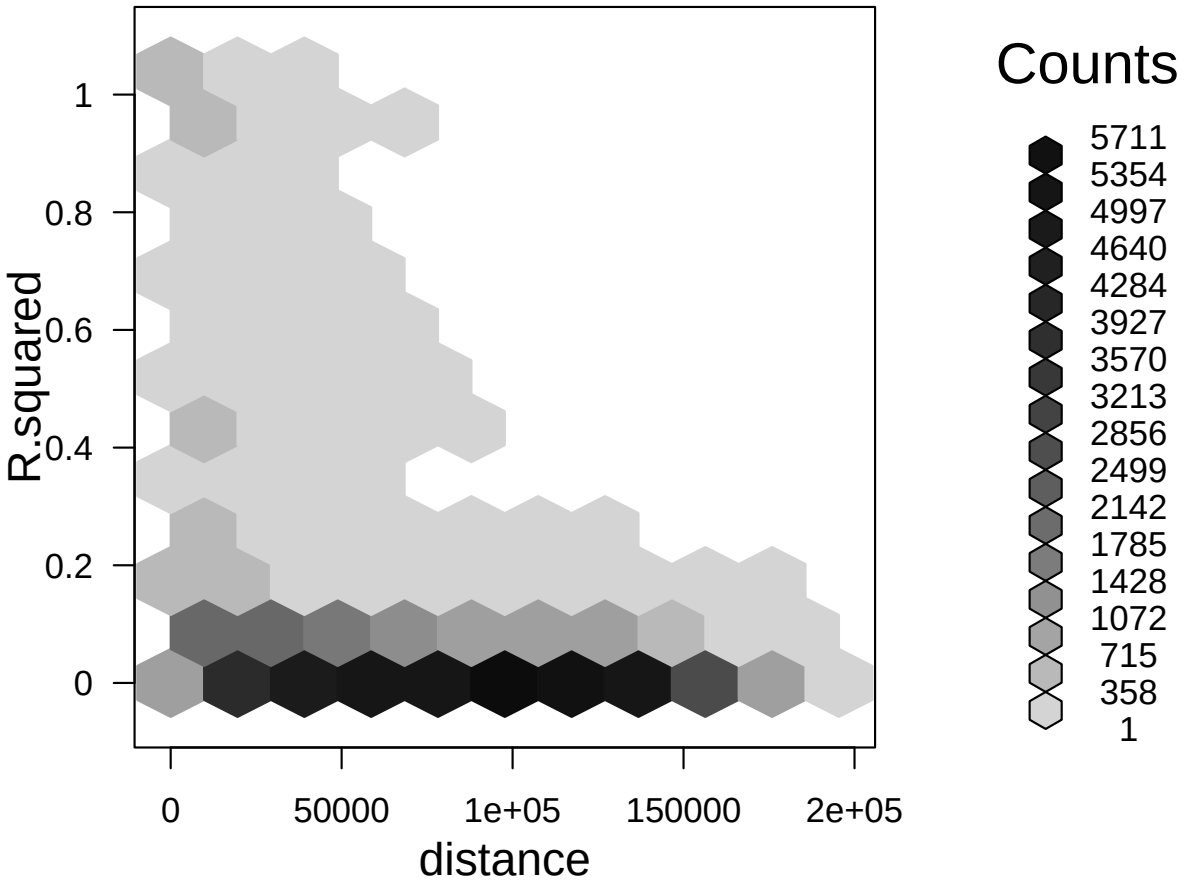


We see that the square of D-prime provides an upper bound for R-squared; a high D-prime indicates the *potential* for two SNPs to be highly correlated, but they need not be. The following commands examine the relationship between the two LD measures and physical distance

```
> plot(hexbin(distance, D.prime, xbin=10))
```




```
> plot(hexbin(distance, R.squared, xbin=10))
```



Although the data are very noisy, the first plot is consistent with an approximately exponential decline in mean D-prime with distance, as predicted by the Malecot model.

A view of the calculations

To understand the calculations let us consider the first and fifth SNPs in the Europeans. We shall first converting these to character data for legibility, and then tabulate the two-SNP genotypes, saving the 3×3 table of genotype frequencies as `tab33`:

```
> snp1 <- as(ceph.1mb[,1], "character")
> snp5 <- as(ceph.1mb[,5], "character")
> tab33 <- table(snp1, snp5)
> tab33

     snp5
snp1  A/A A/B B/B
  A/A   6  21  17
  A/B   2  18  17
  B/B   0   2   7
```

These two SNPs have a moderately high D-prime, but a very low R-squared:

```
> ld.ceph$D.prime[1,5]

[1] 0.570161

> ld.ceph$R.squared[1,5]

[1] 0.06628534
```

The LD measures cannot be directly calculated from the 3×3 table above, but from a 2×2 table of *haplotype frequencies*. In only eight cells around the periphery of the table we can unambiguously count haplotypes and these give us the following table of haplotype frequencies:

	rs361799	
rs5993821	A	B
A	35	72
B	4	33

However, in the central cell of the 3×3 table (*i.e.* `tab33[2,2]`) we have 18 doubly heterozygous subjects, whose genotype could correspond either to the pair of haplotypes `A-A/B-B` or to the pair of haplotypes `A-B/B-A`. These are said to have *unknown phase*. The expected split between these possible phases is determined by a further measure of LD — the odds ratio. If the odds ratio is θ, we expect a proportion $\theta/(1+\theta)$ of the doubly heterozygous subjects to be `A-A/B-B`, and a proportion $1/(1+\theta)$ to be `A-B/B-A`.

We next use `ld` to obtain an estimate of this odds ratio[1] and, using this, we partition the doubly heterozygous individuals between the two possible phases:

```
> OR <- ld(ceph.1mb[,1], ceph.1mb[,5], stats="OR")
> OR
```

[1] Here `ld` is called with two arguments of class `SnpStats` and, since only the odds ratio is to be calculated, it returns the odds ratio rather than a list.

```
          rs361799
rs5993821    4.163

> AABB <- tab33[2,2]*OR/(1+OR)
> ABBA <- tab33[2,2]*1/(1+OR)
> AABB

          rs361799
rs5993821    14.51

> ABBA

          rs361799
rs5993821    3.486
```

We are now able to construct the table of haplotype frequencies:

	rs361799	
rs5993821	A	B
A	49.51	75.49
B	7.486	47.51

It is easy to confirm that the odds ratio in this table, $(49.51 \times 47.51)/(75.49 \times 7.486)$, corresponds closely with that given by the `ld` function. Having obtained the 2×2 table of haplotype frequencies, any LD statistic may be calculated.

Of course, there is a circularity here; we needed to know the odds ratio in order to be able to construct the 2×2 table from which it is calculated! That is why these calculations are not simple. The usual method involves iterative solution using an EM algorithm: an initial guess at the odds ratio is used, as in the calculations above, to compute a new estimate, and these calculations are repeated until the estimate stabilizes. However, in `snpStats` the estimate is calculated in one step, by solving a cubic equation.

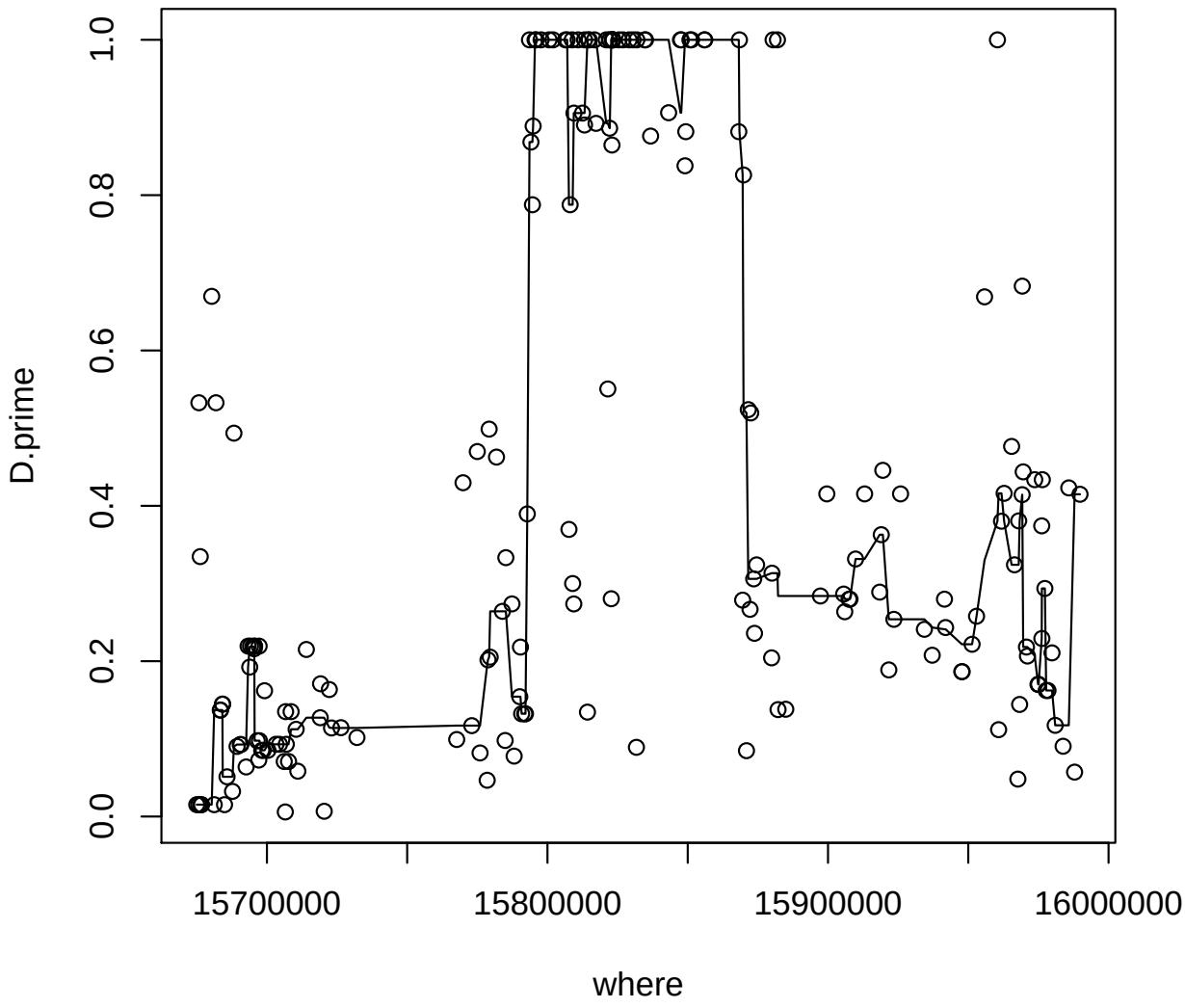
The extent of LD around a point

Often we wish to guage how far LD extends from a given point (for example, from a SNP which is associated with disease incidence). For illustrative purposes we shall consider the region surroung the 168-th SNP, rs2385786. We first calculate D-prime values for the 100 SNPs on either side of rs2385786, and their positions:

```
> lr100 <- c(68:167, 169:268)
> D.prime <- ld(ceph.1mb[,168], ceph.1mb[,lr100], stats="D.prime")
> where <- pos[lr100]
```

We now plot D.prime against position, adding a simple smoother:

```
> plot(where, D.prime)
> lines(where, smooth(D.prime))
```



Although the data are somewhat noisy (the sample size is small), the region of LD is fairly clearly delineated.

Selecting tag SNPs

Several ways have been suggested to select a set of "tag" SNPs which can be used to test for associations in a given region. That described below is based upon a heirarchical cluster

analysis. We shall apply it to the region of high LD identified in the previous section, which lies between positions 1.579×10^7 and 1.587×10^7.

The following commands identify which SNPs lie in this region, and extracts the relevant part of the R^2 matrix, as a symmetric matrix rather than as an upper triangular matrix.

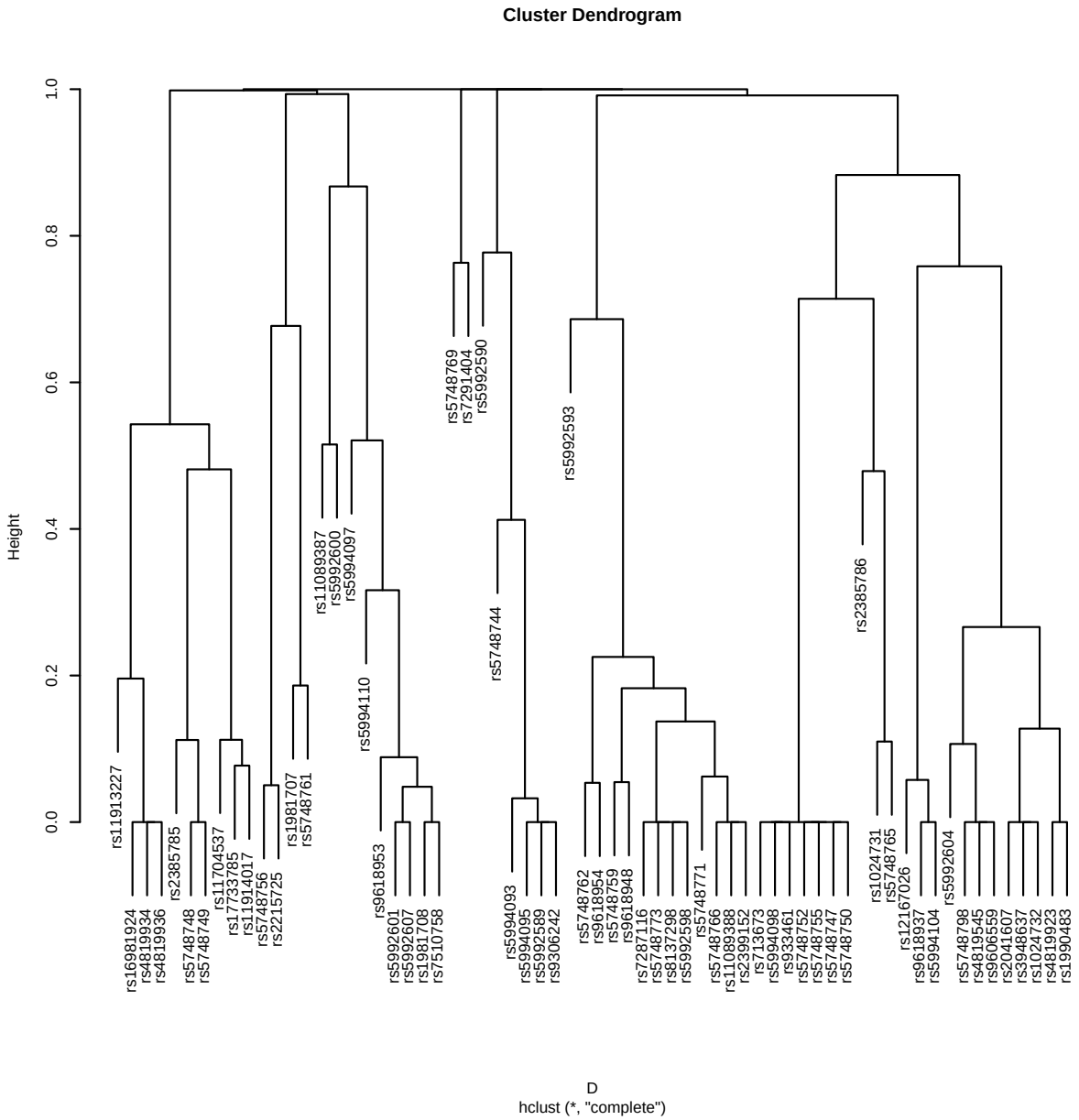
```
> use <- pos>=1.579e7 & pos<=1.587e7
> r2 <- forceSymmetric(ld.ceph$R.squared[use, use])
```

The next step is to convert $(1 - R^2)$ into a distance matrix, stored as required for the hierachical clustering function `hclust`, and to carry out a complete linkage cluster analysis

```
> D <- as.dist(1-r2)
> hc <- hclust(D, method="complete")
```

To plot the dendrogram, we must first adjust the character size for legibility:

```
> par(cex=0.5)
> plot(hc)
```



The interpretation of this dendrogram is that, if we were to draw a horizontal line at a "height" of 0.5, then this would divide the SNPs into clusters in such a way that the value of $(1 - R^2)$ between any pair of SNPs in a cluster would be no more than 0.5 (so that R^2 would be at least 0.5). This can be carried out using the `cutree` function, which returns the cluster membership of each SNP:

```
> clusters <- cutree(hc, h=0.5)
> head(clusters)

rs5994093 rs5748744 rs5992589 rs9306242 rs5994095 rs5992590
        1         1         1         1         1         2
```

```
> table(clusters)

clusters
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
 5  1  7  1  3  6  2  2  6  1  9 12  3  1  1  1  4  1
```

It can be seen that there are 18 clusters. To have a reasonable chance of picking up an association with the SNPs in this 80kb region, we would need to type a SNP from each one of these clusters. Of these, 7 SNPs would only tag themselves!

A threshold R^2 of 0.5 might seem rather low. However, this is a "worst case" figure and most values of R^2 would be substantially better than this, particularly if an effort is made to choose tag SNPs which are in the center of clusters rather than on their edges. Also, this process has only considered tagging by single SNPs; it can be that two or more tag SNPs, taken together, can provide substantially better prediction than any one of them alone.