IBD_Haplo R Tools

Marshall Brown, updated by E. A. Thompson, Fiona Grimson

January 27, 2015

1 Introduction

`IBDhaploRtools` consists of several functions to store, analyze, and plot the output of the `IBD_Haplo`. More information regarding `IBD_Haplo` can be found at www.stat.washington.edu/thomp including a tutorial on how to use the software to generate outputs like the example data sets used in this tutorial.

2 Installing the Package

To begin, install and load the package into `R` if you have not already done so. This tutorial is also included in the package distribution as a vignette.

```
> ## to install the package
> ## install.packages("IBDhaploRtools", repos="http://R-Forge.R-project.org")
>
> ## to load the library at the start of R sesssion
> library("IBDhaploRtools")
>
> ## to bring up this tutorial
> ## vignette("IBDhaploRtools_tutorial")
```

Once the package is loaded, help files for specific functions can be accessed by typing `?[function name]` (for example `?ibdhap.make.calls`.)

It is also possible to view a list of all functions and data sets in the package

```
## list everything in the package
ls("package:IBDhaploRtools", all=TRUE)
## list all the functions and their arguments
lsf.str("package:IBDhaploRtools")
```

```
## list datasets
data(package = "IBDhaploRtools")$results
```

This tutorial provides a description of the main functions in this package, and demonstrates how to implement these functions on IBDhaplo outputs.

3 Reading in Data

IBDhaplo output consists of a qibd file and an ids file. There is the option to supply each file as either a matrix which the user has already loaded into R, or as a string specifying the location of the file.

For this tutorial we will use examples of ids and qibd files for phased (haplotypic) and unphased (genotypic) data that are included in the package. First we need to load the example data sets:

```
> ## to use the tutorial data
> data(qibd_phased)
> data(ids_phased)
> data(ids_unphased)
> data(qibd_unphased)
> data(trueibd_phased)
>
> ## to use you own data, e.g.
> ## my.qibd.filename <- '~/Documents/qibd_unphased_2011.gold'
> ## my.ids.filename <- '~/Documents/ids_unphased_2011.gold'
> ## my.trueibd.filename <- `~/Documents/outfifteen.txt'
>
> ## For data description, e.g.
> ## ?ids_phased
```

The function `ibdhap.makes.calls` will read the files and creates output that is used by all the other functions in the package. Thus, it is imperative to run this `ibdhap.make.calls` first before any other function. The function `ibdhap.make.true` is also important as this reads the true ibd states, information that is used to assess the accuracy of the calls.

ibdhap.make.calls: stores and simplifies the main output files (called "qibd_h.out" in the Gold examples) created by IBD Haplo. This is acheived by "calling" a marker to be in an ibd state if the probability of the state for the marker (conditional on the data and the model) meets the "cutoff" value. If the cuttoff is met, the

value of the marker is set to the ibd state (integer value from 1–15 or 1–9), otherwise value of the marker is set to 0 (which means this marker is a "no call".) `ibdhap.make.states` outputs a `R data.frame` where each row is a marker, and each column is a set of four haplotypes. The value at a marker for a set of haplotypes is as described above. The `R data.frame` that this function creates is expected by other functions in this package.

To run this function on the tutorial data sets, run:

```
> phased.gold <- ibdhap.make.calls( qibd.file  = qibd_phased,
+                     ids.file= ids_phased, cutoff = 0.8)

[1] "qibd matrix accepted"
[1] "ids matrix accepted"

> unphased.gold <- ibdhap.make.calls( qibd.file  = qibd_unphased,
+                     ids.file= ids_unphased, cutoff = 0.8)

[1] "qibd matrix accepted"
[1] "ids matrix accepted"

>
> ## or if you specified a file location
> ## rather than supplying pre-loaded data:
> ## phased.gold <- ibdhap.make.calls( qibd.filename = my.qibd.filename,
> ##                     ids.filename = my.ids.filename, cutoff = 0.8)
```

Both data frames have four columns (one for each chromosome) and 2,000 rows (one for each marker). For the phased data set the first 20 markers for each haplotype are:

```
> phased.gold[1:20,]

    1  2  3  4
1   5  0 15 15
2   5  0 15 15
3   5  0 15 15
4   5  0 15 15
5   5  0 15 15
6   5  0 15 15
7   5  0 15 15
8   5 15 15 15
```

```
9   5 15 15 15
10  5 15 15 15
11  5 15 15 15
12  5 15 15 15
13  0 15 15 15
14 15 15 15 15
15 15 15 15 15
16 15 15 15 15
17 15 15 15 15
18 15 15 15 15
19 15 15 15 15
20 15 15 15 15
```

For marker eight, we see that `IBD_Haplo` inferred with greater than 0.8 probability that the first set of haplotypes were in IBD state 5 while the other three sets of haplotypes were in IBD state 15 (no IBD shared). We also see that for the first set of haplotypes (the first column) IBD state 5 is inferred for markers 1 to 12, and then it transitions to state 15 with a one-marker no-call (returns 0)in between.

ibdhap.names: This function reads the identifying information of the chromosomes making up each set from the ids file. Each chromosome has a number indicating the individual it came from, and a 0/1 indicator of which of that individual's chromosomes it is. The output is a matrix with a row for every set of chromosomes. Like `ibdhap.make.calls`, the ids file can be specified by either naming a matrix already loaded into R, or a string indicating the location of the file.

For example:

```
> ibdhap.names( ids.file = ids_phased )

[1] "ids matrix accepted"
       V4 V5      V6 V7      V8 V9     V10 V11
1 1400003  0 1400003  1 1400009  0 1400009   1
2 1400005  0 1400005  1 1400007  0 1400007   1
3 1400011  0 1400011  1 1400013  0 1400013   1
4 1400015  0 1400015  1 1400017  0 1400017   1

>
>   ## or, alternatively
>   ## ibdhap.names( ids.filename = my.ids.filename)
```

ibdhap.make.true: This function reads in the true ibd states for each locus and each pair of chromosomes, if supplied in MORGAN/IBD_Create/fgl2ibd format. This information will be used later for assessing the accuracy of the calls.

The matrix of true IBD states will have a row for each marker and a column for each set. The true IBD state labelling is assumed to be the Jaquard ordering of 15 (phased) states.

This function will read colums 1 (set names) and 5 (IBD label) from the file location and output a matrix with phased jacquard ordered IBD states, with a row for each marker, and a column for each set of gametes.

```
> ## To load tutorial data:
> ## data( trueibd_phased )
>
> ## To load your own dataset,
> ## my.trueibd.filename <-`~/Documents/fgl2ibdoutput.txt'
> ## ibdhap.make.true( true.filename = my.trueibd.filename )
```

4 Data Analysis

In this section the functions that anaylse the output of `ibdhap.make.states` are demonstrated.

ibdhap.summary.calls : summarizes the data created by `ibdhap.make.calls` by calculating mean lengths of ibd segments, mean proportions of ibd shared, and counts on ibd segments. Averages are taken over all markers and all sets of haplotypes. Note that we need to specify whether we have haplotypic (h) or gentypic (g) data.

```
> ## summary statistics for phased data:
> summary.phased <- ibdhap.summary.calls( phased.gold, data.type="h")
> summary.phased

$mean.prop
any.ibd not.ibd no.call
 0.2155  0.7035  0.0810

$mean.length
    len.ibd len.not.ibd  len.nocall
   61.57143   152.02703    10.60656
```

```
$seg.counts
    ibd  no.ibd no.call
     28      37      61

> ## and for unphased data:
> summary.unphased <- ibdhap.summary.calls( unphased.gold, data.type="g")
> summary.unphased

$mean.prop
any.ibd not.ibd no.call
0.25175 0.61950 0.12875

$mean.length
    len.ibd len.not.ibd  len.nocall
   74.59259   176.92857    20.15686

$seg.counts
    ibd  no.ibd no.call
     27      28      51
```

This shows, for example, that for the phased data on average 70.35% of the chromosome is in IBD state 15 (no ibd shared among the four haplotypes), and that the mean length (number of markers) of a continuous segment of the chromosome with no ibd shared is 152.02703 markers. Also note that the unphased data finds fewer segments, so these data are failing to detect possible state changes.

ibdhap.seg.lengths : given the ibd states from a set of haplotypes/pair of genotypes (taken from a column of the output of ibdhap.make.states), this function creates a `data.frame` consisting of all segments of differing ibd state, paired with their respective length. This function is called by many of the other functions, and provides equivalent information held in a column of "qibd.gold" but just in a different form. If a vector of positions is given, segment lengths will be given in its respective units.

```
> ## load the positions data
> data(posvec)
> seg.lengths.phased <- ibdhap.seg.lengths(phased.gold[,4], position = posvec)
> seg.lengths.phased

   ibd.state seg.lengths
1         15      470519
```

```
2           0      158317
3          15     8808301
4           0        6006
5          12    10087845
6           0      150165
7          12     1595646
8           0      671258
9          15     1995495
10          0       76462
11         13      309818
12          0       11396
13         15      131300
14          0       77996
15         12     3047322
16          0       10153
17         15     1209369
```

```
> seg.lengths.unphased <- ibdhap.seg.lengths(unphased.gold[,4], position = posve
> seg.lengths.unphased
```

```
   ibd.state seg.lengths
1          9     9443143
2          0       34134
3          8     9704717
4          0      137511
5          8     2002836
6          0      734519
7          9     1827738
8          0      421510
9          9      340225
10         0      103190
11         8     2848323
12         0      184415
13         9     1035107
```

In the phased data, for the fourth set of haplotypes, this shows that there are 30 segments of differing ibd state as inferred by `IBD_Haplo`. The seventh segment is in ibd state 12 and consists of 165 markers, while the ninth segment is in ibd state 15 and consists of 117 markers.

Again, notice that fewer IBD/non-IBD segments are detected in the unphased case, and almost all the detected segments in both cases are separated by a few no-call markers (0).

ibdhap.transitions: creates a matrix of transition counts from when ibd state switches along the chromosome. This function also needs to know whether the data are genotypic or haplotypic

```
> ##set the display so the matrix displays nicely
> options(width=100)
> ## transition matrix
> transitions.phased <- ibdhap.transitions(phased.gold,
+                                          data.type="h")
> transitions.unphased <- ibdhap.transitions(unphased.gold,
+                                            data.type="g")
> ## print the transitions matrix into two parts so it
> ## fits on the document
> transitions.phased[,1:8]

      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
 [1,]    0    0    0    0    0    0    0    0
 [2,]    0    0    0    0    0    0    0    0
 [3,]    0    0    0    0    0    0    0    0
 [4,]    0    0    0    0    0    0    0    0
 [5,]    0    0    0    0    0    0    0    0
 [6,]    0    0    0    0    0    0    0    0
 [7,]    0    0    0    0    0    0    0    0
 [8,]    0    0    0    0    0    0    0    0
 [9,]    0    0    0    0    0    0    0    0
[10,]    0    0    0    0    0    0    0    0
[11,]    0    0    0    0    0    0    0    0
[12,]    0    0    0    0    0    0    0    0
[13,]    0    0    0    0    0    0    0    0
[14,]    0    0    0    0    0    0    0    0
[15,]    0    0    1    0    2    0    0    4

> transitions.phased[,9:15]

      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
 [1,]    0    0    0    0    0    0    0
 [2,]    0    0    0    0    0    0    0
```

```
 [3,]    0    0    0    0    0    0    1
 [4,]    0    0    0    0    0    0    0
 [5,]    0    0    0    0    0    0    3
 [6,]    0    0    0    0    0    0    0
 [7,]    0    0    0    0    0    0    0
 [8,]    0    0    0    0    1    0    3
 [9,]    0    0    0    0    0    0    0
[10,]    0    0    0    0    1    0    0
[11,]    0    0    0    0    0    0    6
[12,]    0    0    0    0    0    0    4
[13,]    0    0    1    0    0    0    3
[14,]    0    0    0    0    0    0    3
[15,]    0    1    5    4    2    3    0
```

```
> ## unphased transitions matrix
> transitions.unphased
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
 [1,]    0    0    0    0    0    0    0    0    0
 [2,]    0    0    0    0    0    0    0    0    0
 [3,]    0    0    0    0    0    0    0    0    1
 [4,]    0    0    0    0    0    0    0    1    2
 [5,]    0    0    0    0    0    0    0    0    0
 [6,]    0    0    0    0    0    0    0    0    3
 [7,]    0    0    0    0    0    0    0    1    0
 [8,]    0    0    1    1    0    1    1    0   11
 [9,]    0    0    0    2    0    2    0   13    0
```

Hence, in the unphased transitions, we saw 13 instances of ibd state 9 transitioning to ibd state 8, for example. The small no-call segments between the IBD/non-IBD segments are also left out.

ibdhap.barplot : Graphically displays regions of any ibd sharing, no ibd sharing and no calls along a chromosome for a set of haplotypes / pair of genotypes. The default colors are red, white, and grey for any state with ibd sharing, no ibd sharing and no calls, respectively.

For the phased data:

```
> ## Figure 1
> par(mfrow=c(4,1))
> ibdhap.barplot(phased.gold[,1], data.type="h",
```

```
+                 xlab="", ylab="", position = posvec)
> ibdhap.barplot(phased.gold[,2], data.type="h",
+                 xlab="", ylab="", position = posvec)
> ibdhap.barplot(phased.gold[,3], data.type="h",
+                 xlab="", ylab="", position = posvec)
> ibdhap.barplot(phased.gold[,4], data.type="h",
+                 xlab="", ylab="", position = posvec)
```

For the unphased data:

```
> ## Figure 2
> par(mfrow=c(4,1))
> ibdhap.barplot(unphased.gold[,1], data.type="g",
+                 xlab="", ylab="", position = posvec)
> ibdhap.barplot(unphased.gold[,2], data.type="g",
+                 xlab="", ylab="", position = posvec)
> ibdhap.barplot(unphased.gold[,3], data.type="g",
+                 xlab="", ylab="", position = posvec)
> ibdhap.barplot(unphased.gold[,4], data.type="g",
+                 xlab="", ylab="", position = posvec)
```

We can see that the first set of haplotypes does not have much (inferred) ibd shared among the four haplotypes, while the fourth set has a large portion of the chromosome in an ibd state where ibd is shared among the four haplotypes. states.

5 Comparison to True IBD

Since we simulated these haplotypes, and therefore know the true ibd states for these four sets of haplotypes, it is interesting to compare the previous barplots to similar barplots created from using the known ibd states. For the tutorial data we have a file containing the true IBD states for the phased data. We can transform this data to the true IBD states for unphased data.

h.to.g : This function is to transform the true ibd states in haplotypic (phased) format to gentoypic (unphased) format. Recall that the phased data has 15 possible IBD states, and the phased data has 9 possible ibd states. The function can be applied to any matrix with entries that are integers between 0 and 15.

The labelling of the 9 or 15 IBD states is assumed to be Jaquard ordering.

```
> ## load the true IBD states data set
> data(trueibd_phased)
> ## map phased to unphased labels
> trueibd_unphased <- h.to.g( trueibd_phased )
> ## view the first few rows of each data set
> head(trueibd_phased)

   [,1] [,2] [,3] [,4]
V3   15   15   15   15
V4   15   15   15   15
V5   15   15   15   15
V6   15   15   15   15
V7   15   15   15   15
V8   15   15   15   15

> head(trueibd_unphased)

   [,1] [,2] [,3] [,4]
V3    9    9    9    9
V4    9    9    9    9
V5    9    9    9    9
V6    9    9    9    9
V7    9    9    9    9
V8    9    9    9    9
```

Notice that in the "Reading in Data" section we saw in the IBDhaplo output `phased.gold` the inferred IBD state for the first 12 markers on the first set of haplotypes was 5. The true state was 9, so this was a false positive.

Now we can examine the true ibd states graphically using ibdhap.barplot. Phased data:

```
> ## Figure 3
> par(mfrow=c(4,1))
> ibdhap.barplot(trueibd_phased[,1], data.type="h",
+                xlab="", ylab="", position = posvec)
> ibdhap.barplot(trueibd_phased[,2], data.type="h",
+                xlab="", ylab="", position = posvec)
> ibdhap.barplot(trueibd_phased[,3], data.type="h",
+                xlab="", ylab="", position = posvec)
> ibdhap.barplot(trueibd_phased[,4], data.type="h",
+                xlab="", ylab="", position = posvec)
```

I have not plotted the barplot in the unphased case as only IBD and non-IBD are displayed, so both plots would be identical.

Comparing Figure 3 to Figures 1 and 2, we see that `IBD_Haplo` infers many of the IBD segments. However, many of the small segements of inferred IBD were false positives, such as the small region at the beginning of the first set of haplotypes for the phased data. The unphased data has more false positives than the phased.

ibdhap.compare.loci : Compares inferred ibd state with simulated "true" states, locus by locus. Calculates the proportion of markers called in the correct state, called as wrong ibd or no ibd, and no calls, for both the states that are truly ibd and states that are truly non-ibd. A table is also prepared that includes the percentage of the loci that are truly in and called to be in each ibd state.

```
> ## phased data
> ibdhap.compare.loci(phased.gold, trueibd_phased, "h")

$all

Number of sites          4.00000
Called Correctly         0.88400
Called IBD Incorrectly   0.03075
Called noIBD Incorrectly 0.00425
No-call                  0.08100

$ibd

Number of sites     1680.000000
Called Correctly      87.976190
Called as wrong IBD    0.297619
Called as no IBD       2.023810
No-call                9.702381

$nonibd

Number of sites  6320.000000
Called Correctly   88.512658
Called as IBD       3.813291
No-call             7.674051

$categories
```

```
   Category Inferred    True
0         0   8.1000  0.0000
1         1   0.0000  0.0000
2         2   0.0000  0.0000
3         3   0.1375  0.0000
4         4   0.0000  0.0000
5         5   0.8125  0.4500
6         6   0.0000  0.0000
7         7   0.0000  0.0000
8         8   0.7125  0.5125
9         9   0.0000  0.0000
10       10   0.3750  0.3125
11       11   3.3125  2.5375
12       12  11.5625 12.1875
13       13   1.8875  2.3875
14       14   2.7500  2.6125
15       15  70.3500 79.0000

> ## unphased data
> ibdhap.compare.loci(unphased.gold, trueibd_unphased, "g")

$all

Number of sites          4.00000
Called Correctly         0.79000
Called IBD Incorrectly   0.07700
Called noIBD Incorrectly 0.00425
No-call                  0.12875

$ibd

Number of sites     1680.000000
Called Correctly      83.214286
Called as wrong IBD    0.952381
Called as no IBD       2.023810
No-call               13.809524

$nonibd

Number of sites  6320.000000
```

```
Called Correctly   77.879747
Called as IBD       9.493671
No-call            12.626582

$categories
  Category Inferred    True
0        0  12.8750  0.0000
1        1   0.0000  0.0000
2        2   0.0000  0.0000
3        3   0.1125  0.4500
4        4   1.1500  0.0000
5        5   0.0000  0.0000
6        6   0.6500  0.5125
7        7   0.3125  0.3125
8        8  22.9500 19.7250
9        9  61.9500 79.0000
```

We see that the unphased data give more no-call and false positives. The number of false negatives is the same in this example.

ibdhap.compare.segs: this function summarizes how well IBDhaplo has inferred segments of ibd. A list with two elements is created. The first is a matrix of summary statistics detailing the number of segments, the proportions called correctly, incorrectly, no-ibd or no-call. The second element is a matrix in which each row represents a segment of true ibd, with a column for segment length, true ibd state, called ibd state, the modal called value for the ibd state and the proportion of loci in the segment that are correctly called.

```
> ## Run the function on the phased data
> ## NB: we can also use pos = posvec
> corr.seg.phased <- ibdhap.compare.segs( phased.gold,
+                                         trueibd_phased,
+                                         "h", 0.8, pos = NA)
> ## First element of output is summary statistics
> corr.seg.phased$seg.stats

                              [,1]
Number segments           40.00000
Number of IBD segments    19.00000
IBD Segs called correctly 68.42105
IBD Segs called no-IBD    15.78947
```

```
IBD Segs called wrong IBD  0.00000
IBD Segs with no call     15.78947

> ## Same thing for the unphased data
> ## first converting true states to unphased
> trueibd_unphased <- h.to.g( trueibd_phased )
> corr.seg.unphased <- ibdhap.compare.segs( unphased.gold,
+                                           trueibd_unphased,
+                                           "g", 0.8, pos = NA )
> corr.seg.unphased$seg.stats

                             [,1]
Number segments           40.00000
Number of IBD segments    19.00000
IBD Segs called correctly 52.63158
IBD Segs called no-IBD    15.78947
IBD Segs called wrong IBD  0.00000
IBD Segs with no call     31.57895

>
```

We can use output of `ibdhap.correct.bylen` to create Figures 5 and 6. The x axis is the length of the segment, and the y axis is proportion of SNPs called correctly within the true segment.

```
> par(mfrow=c(1,1)) ## reset the plot window to square
> ## Figure 4
> plot( corr.seg.phased$seg.info[,1], corr.seg.phased$seg.info[,5],
+        xlab="segment length", ylab="proportion correct",
+        main = "Phased Data")
>  lines(lowess(corr.seg.phased$seg.info[,1],
+        corr.seg.phased$seg.info[,5]), col="blue")

> ## Figure 5
> plot( corr.seg.unphased$seg.info[,1], corr.seg.unphased$seg.info[,5],
+        xlab="segment length", ylab="proportion correct",
+        main = "Unphased Data")
> lines(lowess(corr.seg.unphased$seg.info[,1],
+       corr.seg.unphased$seg.info[,5]), col="blue")
```

We can see, for example, in Figure 4 that `IBD\_Haplo` does very well with ibd segments that are above 25 to 50 snps in length. In these data, this corresponds with ibd segments that are roughly one centimorgan in length. Comparing the two plots, we see again that the unphased data does less well than the phased, with lower proportions correct.

This completes our quick run through of the main functions of `IBDHaploRtools`. Please see the man pages for more information regarding function arguments, descriptions, and examples.

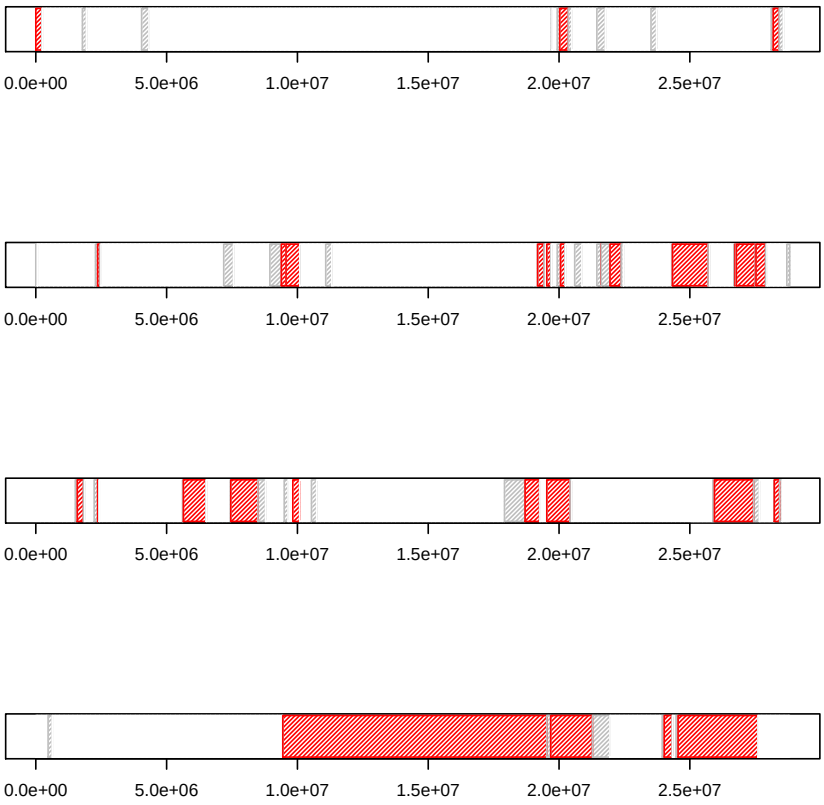

Figure 1: Barplots for four sets of four PHASED haplotypes using IBD_Haplo output. Red means some ibd shared, white is no ibd shared and grey means "no call."

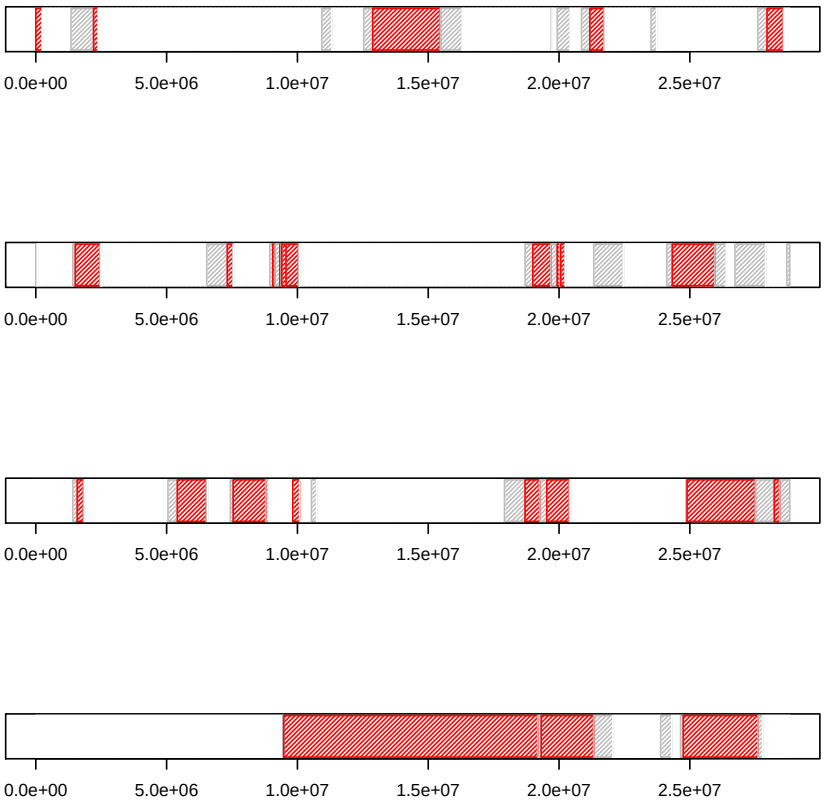

Figure 2: Barplots for four sets of four UNPHASED haplotypes using IBD_Haplo output. Red means some ibd shared, white is no ibd shared and grey means "no call."

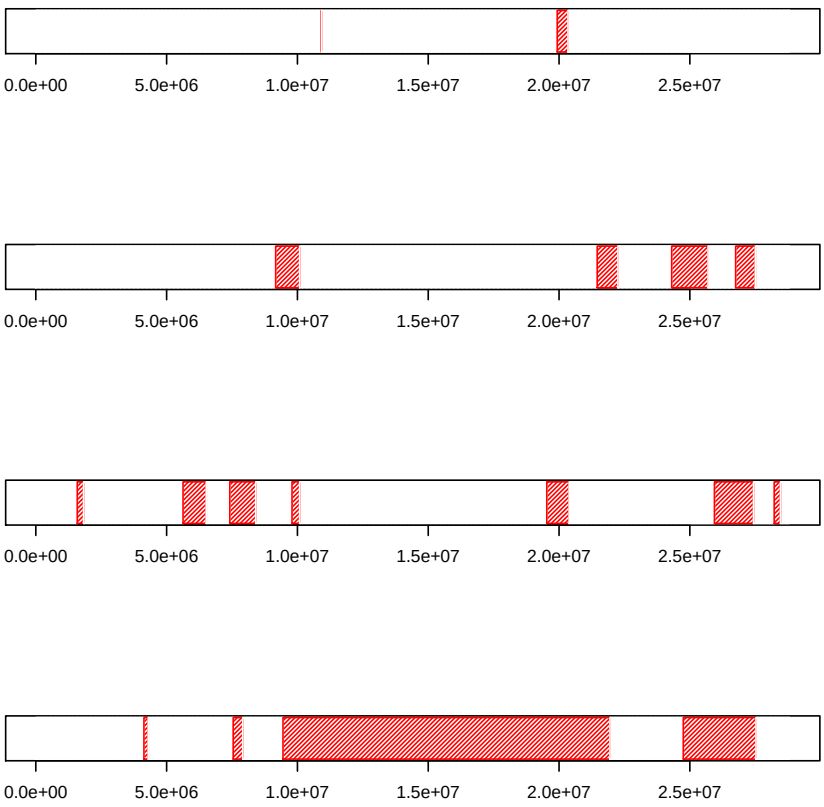

Figure 3: Barplots for four sets of four PHASED haplotypes using true IBD states of simulated data. Red means some ibd shared, and white is no ibd shared

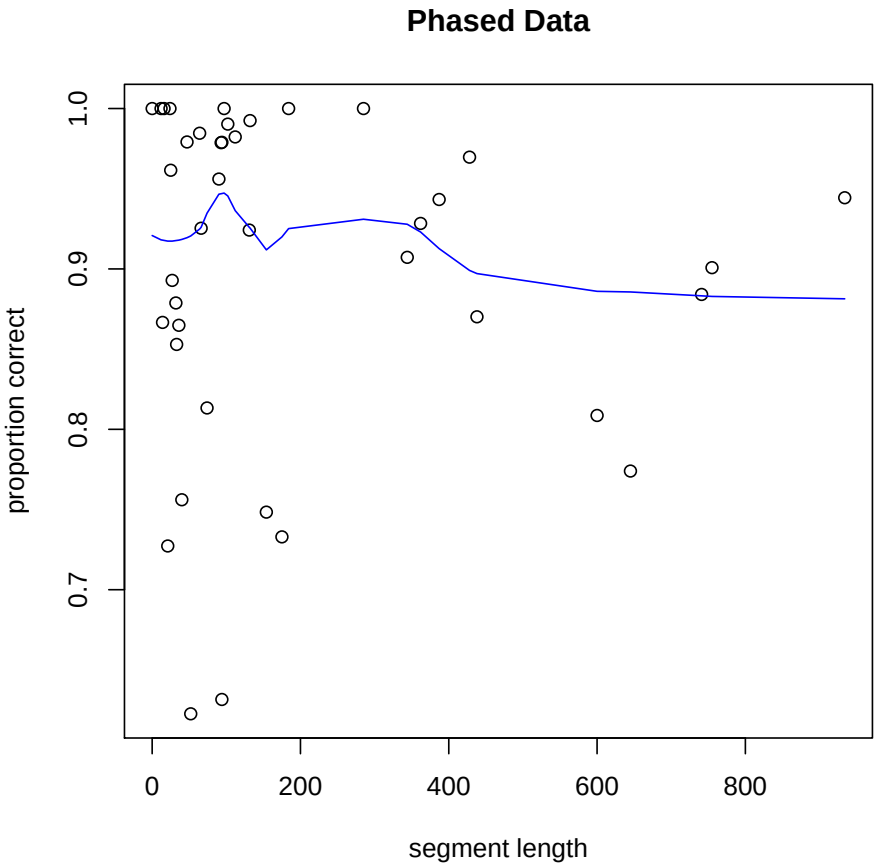

Figure 4: Each point in the plot is a segment of true (simulated) ibd (shared or not shared).The x axis is the length of the segment, and the y axis is proportion snps called correctly within the true segment

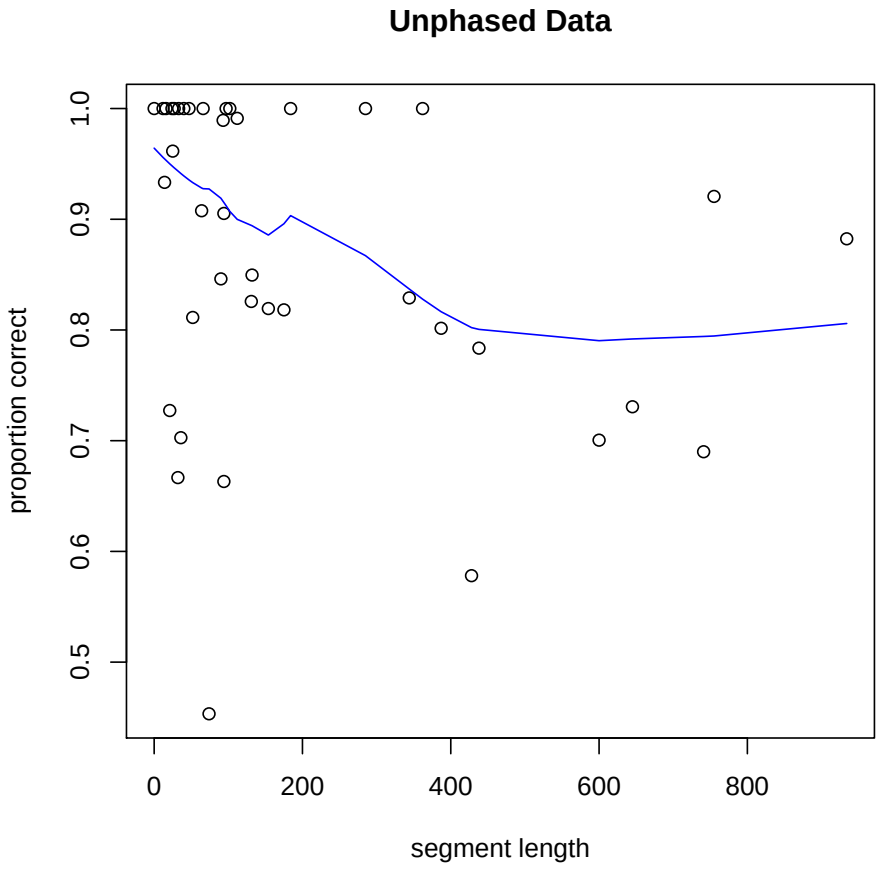

Figure 5: Each point in the plot is a segment of true (simulated) ibd (shared or not shared).The x axis is the length of the segment, and the y axis is proportion snps called correctly within the true segment