

Friday, 27 July

- *Server disk-space issue fixed!*
- Linkage mapping with OneMap
- Class photograph ~10:45

See `HandsOnDArT/lectures_and_exercises/onemap_notes.txt` for a detailed walk-through

Getting set up for R and OneMap

- To run R and OneMap on your computer, you should have:
 - R
 - OneMap
 - Ideal: <https://github.com/augusto-garcia/onemap> dev version
 - Also OK: `install.packages("onemap", dependencies=TRUE)`
 - **Mac users:**
 - Xcode (download from App Store)
- If all else fails, run on the server, and download plots to look at them

Loading the OneMap package and reading data

To load the OneMap package, use `require()`
`require(onemap)`

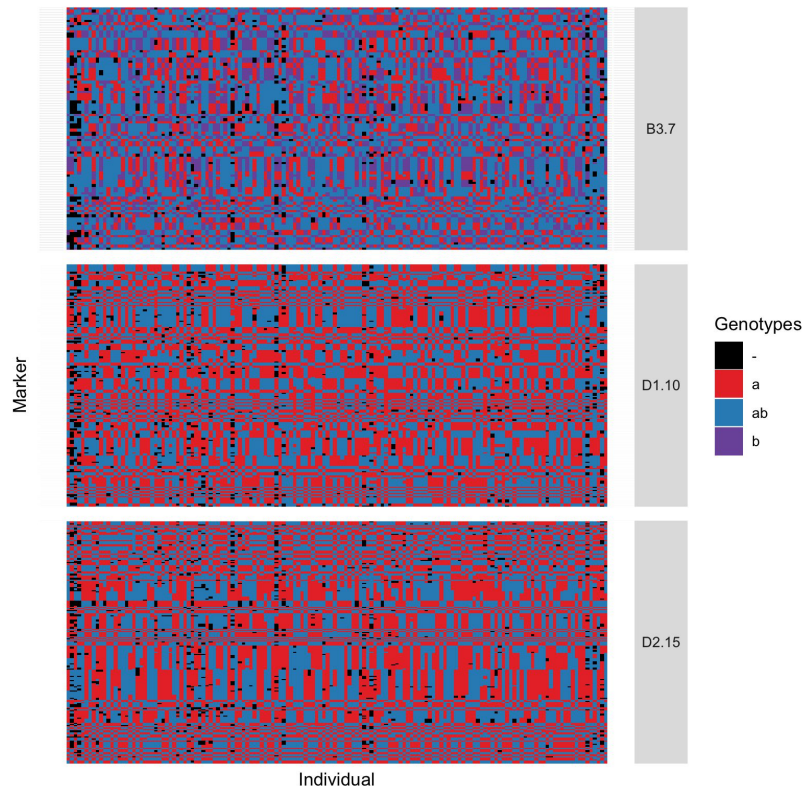
When you load mapping data, cross type is read from the file
`cross = read_onemap("/path/to/dir", "F1.raw")`

Checking your data quality

Plot colored genotype matrix:
`plot(cross, all=FALSE)`

Assess consistency of genotype patterns within colored blocks.

A pixel of a color in the block of another color is indicative of a genotyping error. How pervasive is it?

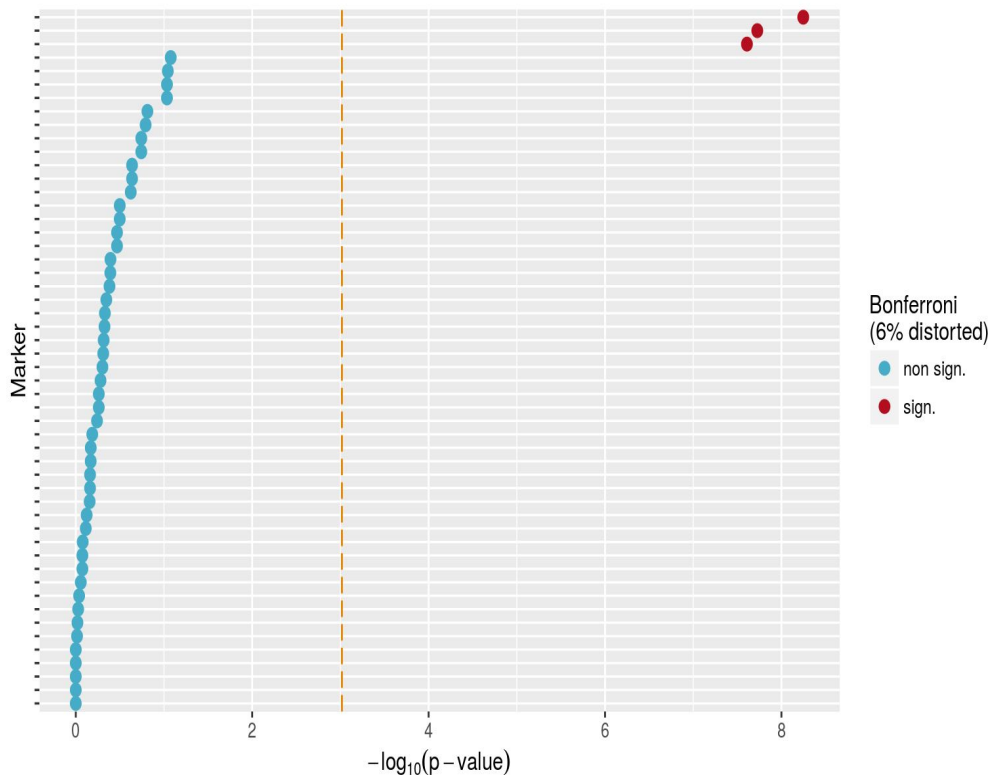


Checking your data quality

Was our chi-squared test stringent enough?

```
plot(test_segregation(cross))
```

Best markers are on the left of the orange vertical line; the presence of markers on the right suggests your χ^2 test threshold should be more stringent.



Linkage mapping

Three critical parameters:

Calculating linkage groups

1. Minimum LOD threshold (LOD)
2. Maximum recombination fraction (`rf.max`)

Estimating marker order

3. Number of initial markers (`n.init`)

Calculating linkage groups

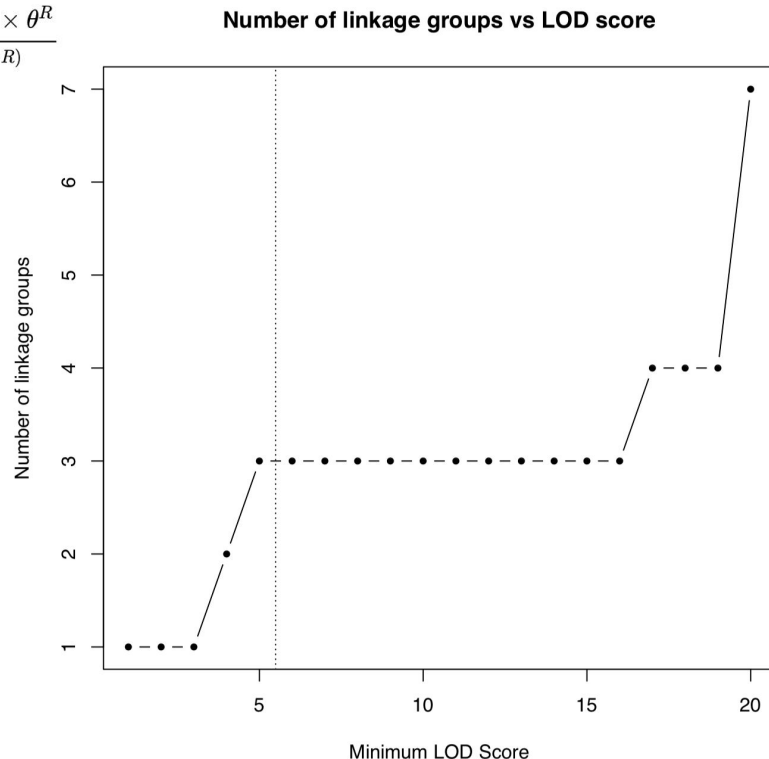
$$LOD = Z = \log_{10} \frac{\text{probability of birth sequence with a given linkage value}}{\text{probability of birth sequence with no linkage}} = \log_{10} \frac{(1 - \theta)^{NR} \times \theta^R}{0.5^{(NR+R)}}$$

OneMap can recommend a LOD score:

```
lod.suggested = suggest_lod(cross)
```

It is always good practice to check that it is a valid suggestion.

- Calculate the number of linkage groups over a range of minimum LOD value and look for a plateau of stability in the plot, indicating strong/stable linkage groupings.



Calculating linkage groups

We must calculate the genetic distance between every pair-wise combination of markers to estimate the strength of linkage:

```
recombination.fractions = rf_2pts(cross, LOD=0, rf.max=0.5)
```

Select which markers we want to group (all):

```
markers = make_seq(recombination.fractions, "all")
```

Partition markers with strong linkage into linkage groups:

```
linkage.groups = group(markers, LOD=min.lod, rf.max=0.5)
```


Estimating marker order

Given a group, produce a linear order of markers, representing the chromosome (Maximum Likelihood)

Extract LG i markers from OneMap “group” object and order them:

```
marker.ord.i = order_seq(make_seq(linkage.groups, i), n.init=7)
```

Extract marker order from from OneMap “order” object

```
marker.map.i = make_seq(marker.ord.i, "force")
```

“force” = all markers

“safe” = only those that pass the specified LOD threshold

Estimating marker order

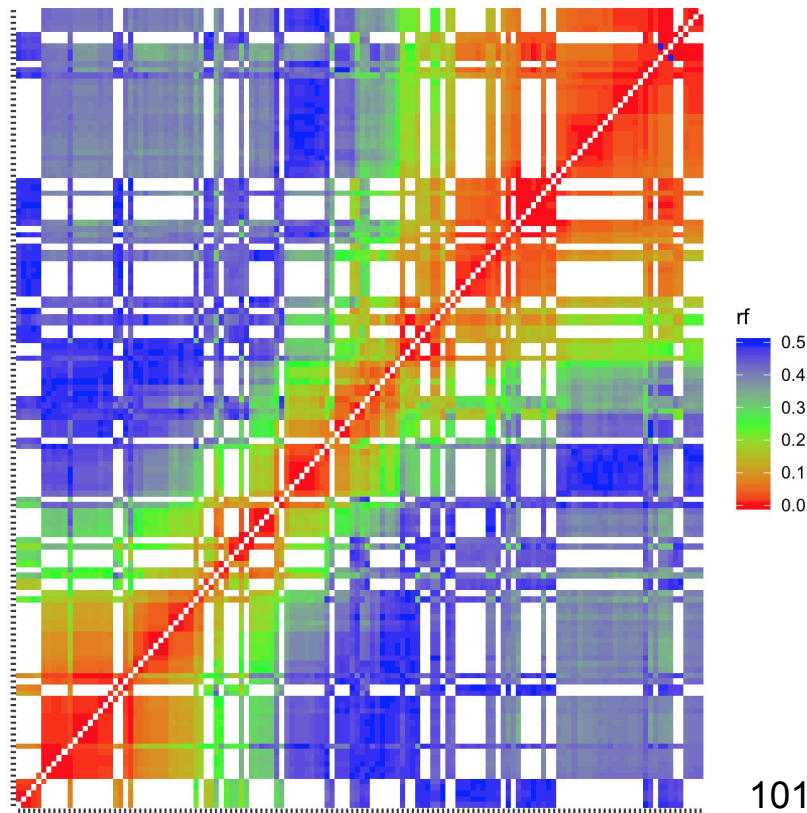
Refine the marker order by “rippling” markers across the linkage group:

```
ripple_seq(marker.map.i, ws=5)
```

ws = “window size”, number of markers to ripple each iteration

Plot a heatmap of recombination fractions between pairs of markers:

```
plot(rf_graph_table(marker.map.i,  
    inter=F, graph.LOD=F, n.colors=3))
```



Visualizing the map

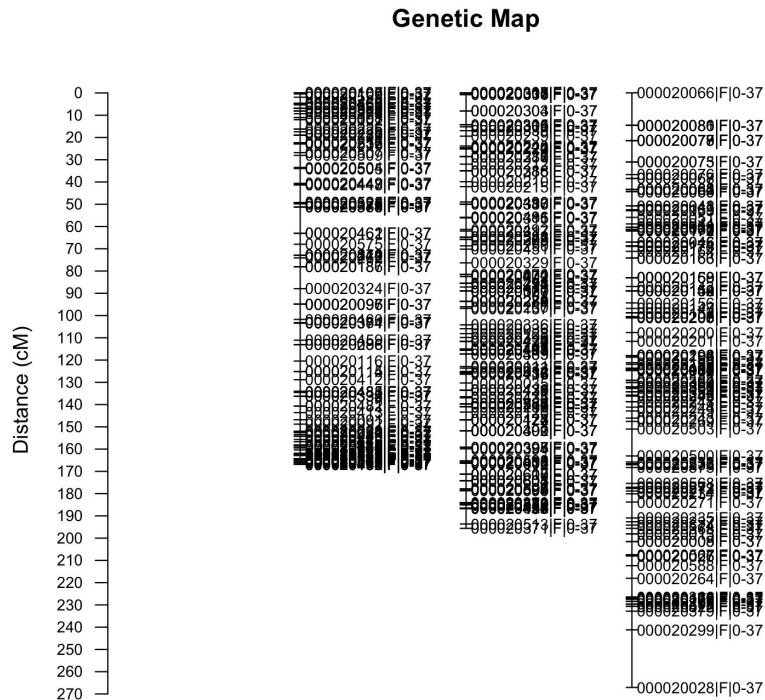
Draw a nice graphic of the linkage groups:

```
draw_map(group.table, names = TRUE)
```

Linkage groups with lengths much larger than 100 cM indicates the presence of many genotyping errors in the data

100 cM = 1 Morgan

= 1 recombination per chromosome



Saving your maps

Use the `write_map` function to write a space-delimited file of marker orders:

```
write_map(marker.map.i, file=paste("file.prefix",i,"txt", sep='.'))
```

```
1 000020105|F|0-37 0
1 000020106|F|0-37 9.99999999981846e-05
1 000020107|F|0-37 0.000199999999996369
1 000020102|F|0-37 0.676102444861517
1 000020555|F|0-37 2.02796546722068
1 000020465|F|0-37 4.67331089992381
1 000020469|F|0-37 4.67341089992381
1 000020468|F|0-37 4.67351089992381
1 000020467|F|0-37 4.6736108999238
```