tspOrder_tutorial

JT Lovell

2/27/2017

Contents

---

email: johntlovell@gmail.com – website: lovelleeb.weebly.com – github: github.com/jtlovell/qtlTools

---

1 Simulate a genetic map and genotype data

For the purposes of this tutorial, we will use paramters that mimick some types of NGS data:

- 10% Missing data
- 0.1% Genotyping error
- 3 chromosomes
- 50 markers / chromosome (.2 - .5 markers /cM)
- unequal marker density across the chromosomses

1.1 Load the required packages - install if needed

```
library(devtools)
install_github("jtlovell/qtlTools")
install_github("mckaylab/TSPmap")
```

```
libs2load<-c("ASMap","qtlTools","TSP","TSPmap")
suppressMessages(sapply(libs2load, require, character.only = TRUE))
```

1.2 Simulate the genetic map

```
set.seed(42)
map <- sim.map(len=c(150,200,100),
               n.mar=c(50,50,50),
               include.x =FALSE, sex.sp =F)
plot.map(map)
```

```
map.orig<-map
```

1.3 Simulate the R/qtl cross object

```
cross.sim <- sim.cross(map, type="riself", n.ind=250, map.function = "kosambi",
                       error.prob = 0.001, keep.qtlgeno = F,missing.prob=0.1)
cross<-cross.sim
```

1.4 Randomize the positions of the markers so that we can re-order them

```
newMarkerList<-lapply(chrnames(cross), function(x) sample(markernames(cross, chr = x)))
names(newMarkerList)<-chrnames(cross)
# randomize the positions of markers
cross<-newLG(cross, markerList = newMarkerList)
```

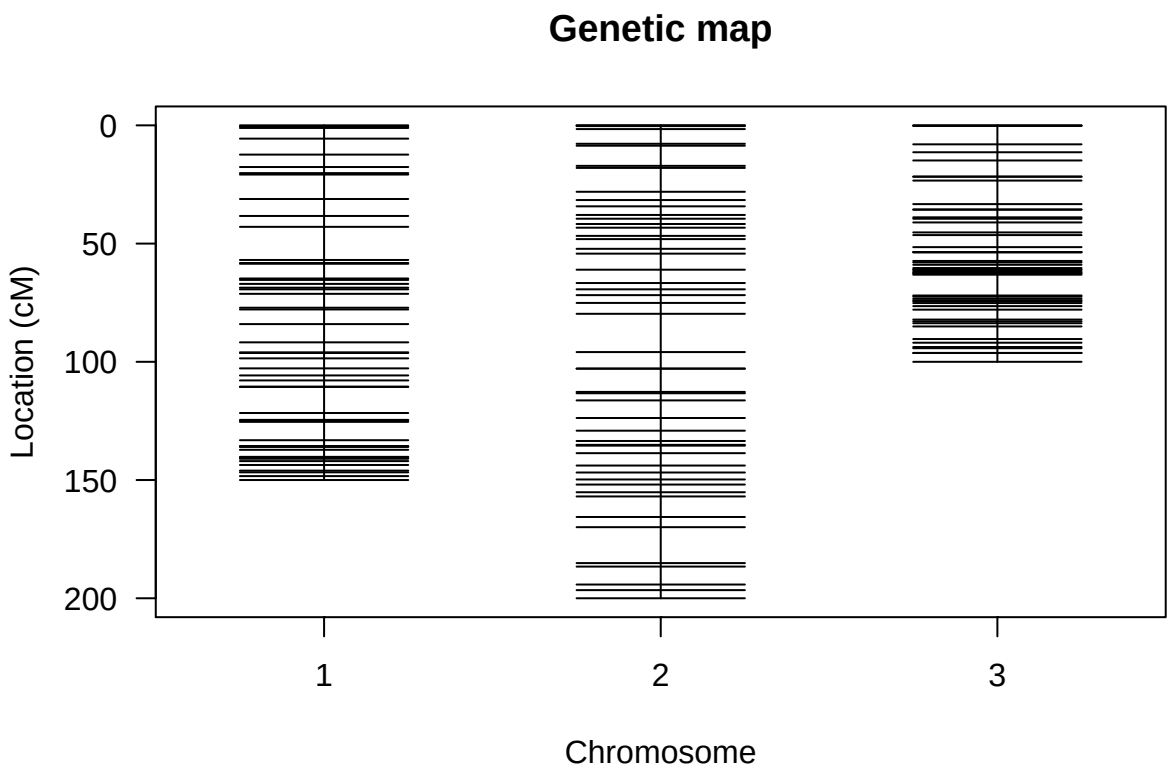

Figure 1: the simulated genetic map for the tutorial. Each horizontal line indicates 1 marker.

2 The structure of the data pre-ordering

2.1 Estimate recombination fractions

To order markers, we must calculate chromosome - specific recombination fractions. For very large genetic maps (>5k markers for a RIL, >3k markers for a 4-way), we should not attempt to do this for the entire map. Instead, we can do it chromosome-by-chromosome.

For the purposes of this tutorial, let's calculate it for the whole population. Notice that

```
cross<-est.rf(cross)
plot.rf(cross)
```

3 Order markers using TSPmap

3.1 Overview

There are many methods to order markers. While joinmap4 remains the industry standard, it has drawbacks - it's slow and is not free. Recently several other methods have been proposed using graph theory. MSTmap is a good option, but it can only handle populations with 2-genotype markers (BC/RIL/DH/etc), not F2, 4-way etc. mapping populations. To overcome these issues, it is optimal to assess marker orders through evaluation of the recombination fraction matrix only.

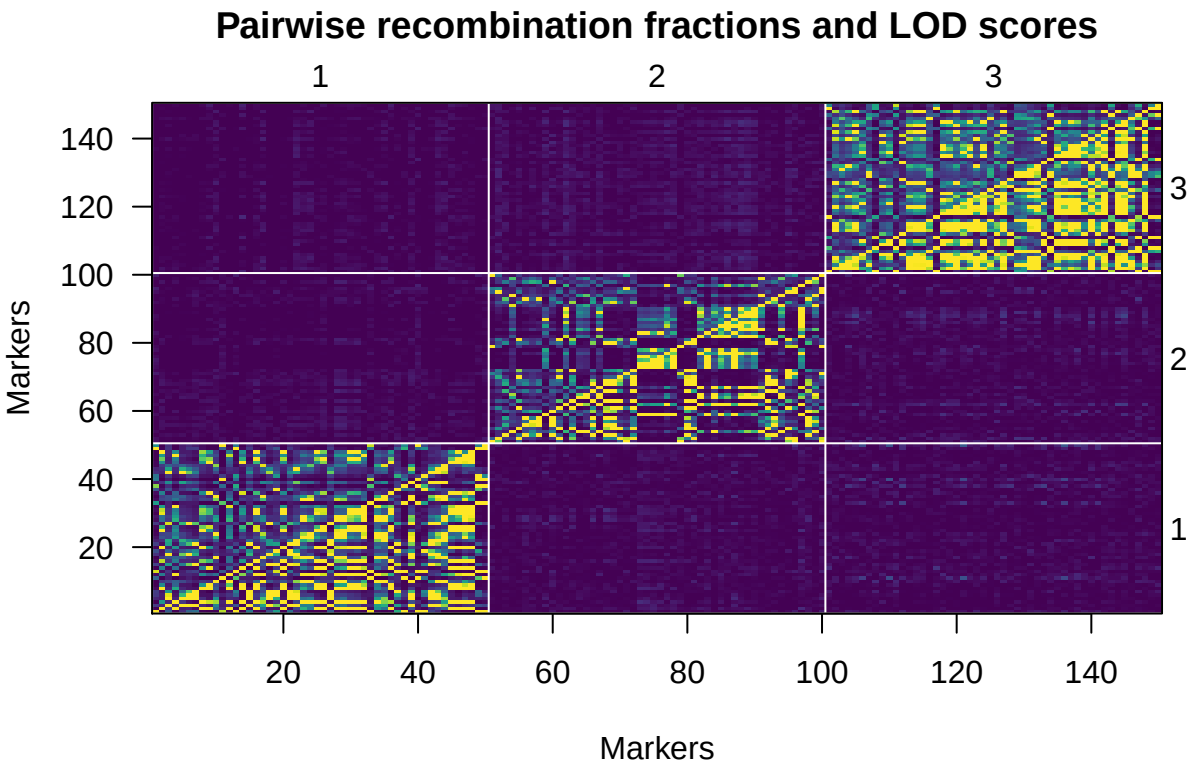

Figure 2: Pairwise recombination fraction plot following linkage group construction, but before marker ordering. Note that recombination fractions are always high between chromosomes (blue). All pairs of highly correlated markers (low recombination fractions, yellow) exist on the same chromosome. Since marker order is random, it is not surprising that close positions on a chromosome do not show low recombination fractions

3.2 Marker ordering using TSP solvers

Here, we build upon the `TSPmap` protocol and employ a travelling salesperson problem solver to find the shortest path through the recombination fraction matrix, and allow the user to directly pipe an R/qtl cross object into the TSPmap protocol. The optimal method is `concorde`, however, this requires a separate installation of the concorde program. See `?tspOrder` for details on how to do this.

```
cross.order<-tspOrder(cross = cross,hamiltonian = TRUE,
                  method="concorde",concorde_path = "/Users/John/Documents/concorde/TSP")
```

3.3 Visualize the new marker order

```
cross.order<-est.rf(cross.order)
plot.rf(cross.order, main = "")
```

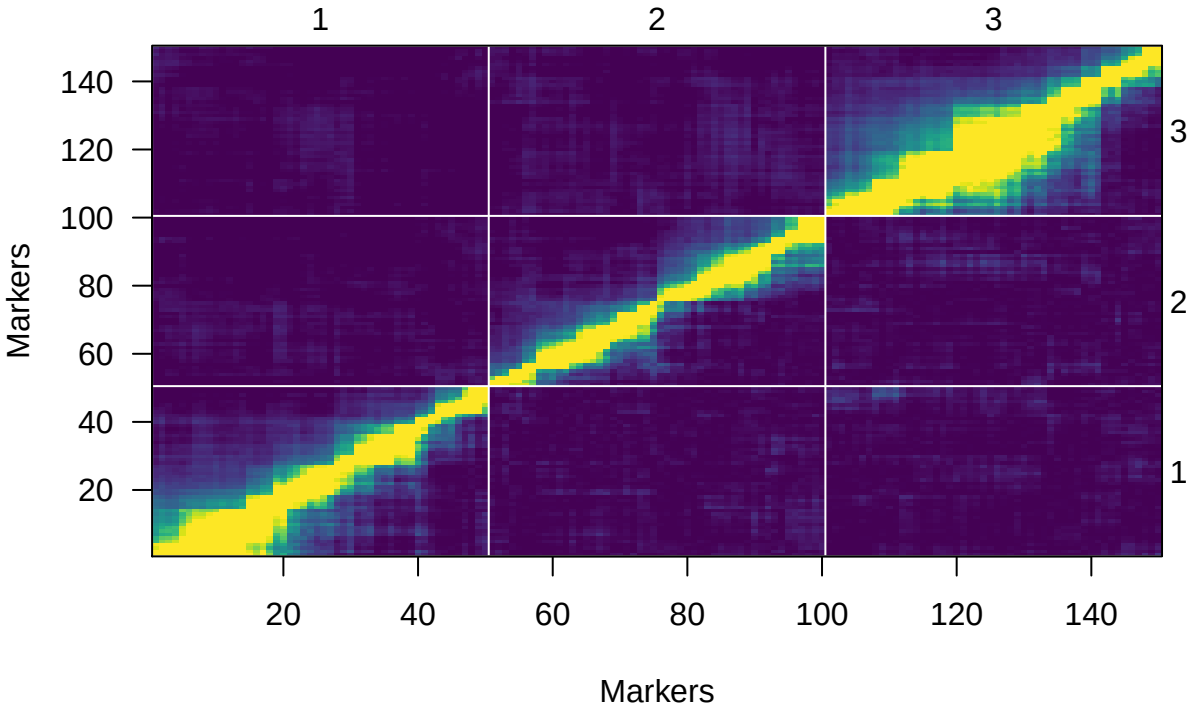

Figure 3: Pairwise recombination fraction plot following marker ordering. Note that following the marker reordering, there is a decay in correlations between markers from the diagonal. This is an indication of a high-quality map, since distant markers are not correlated.

3.4 Compare to the original

```
cross.sim<-est.rf(cross.sim)
plot.rf(cross.sim, main = "")
```

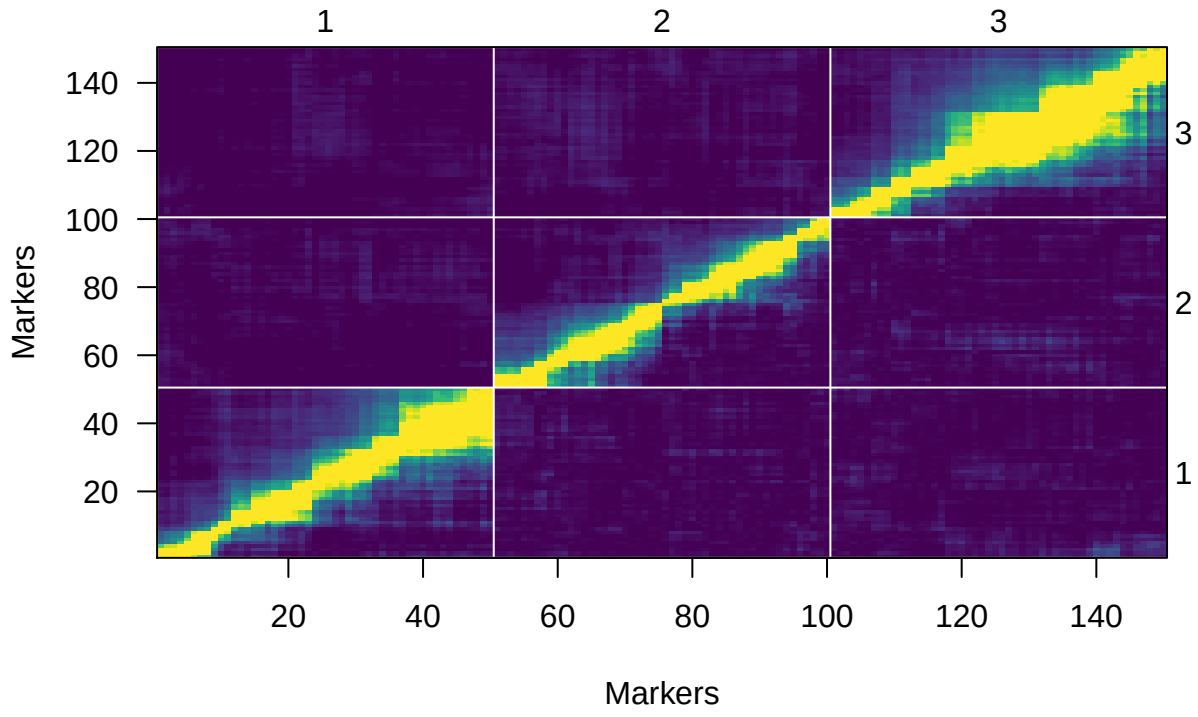

Figure 4: Recombination fractions of the simulated genetic map

4 Estimate the genetic maps and compare to the original simulated data

4.1 Estimate the genetic map for the new cross

```
newmap<-est.map(cross.order, map.function = "kosambi", error.prob = 0.001)
cross.out<-replace.map(cross.order, newmap)
```

4.2 Compare the new to the simulated genetic maps

Note that the marker order is identical between the two crosses

```
plot.map(cross.sim, cross.out)
```

4.3 Switch chromosome orientation to match simulated data

TSPmap runs agnostic to the orientation of the chromosomes. Since we started with simulated data, it is appropriate to re-orient the chromosomes to match these

```
map.sim<-pullMap(cross.sim)
colnames(map.sim)[2:3]<-c("orig.chr","orig.pos")
map.out<-pullMap(cross.out)
map<-merge(map.out, map.sim, by = "marker.name")
map<-map[match(map.out$marker.name, map$marker.name),]
cross.out<-matchMarkerOrder(cross.out, marker.chr.bp = map$orig.chr, marker.pos.bp = map$orig.pos)
```

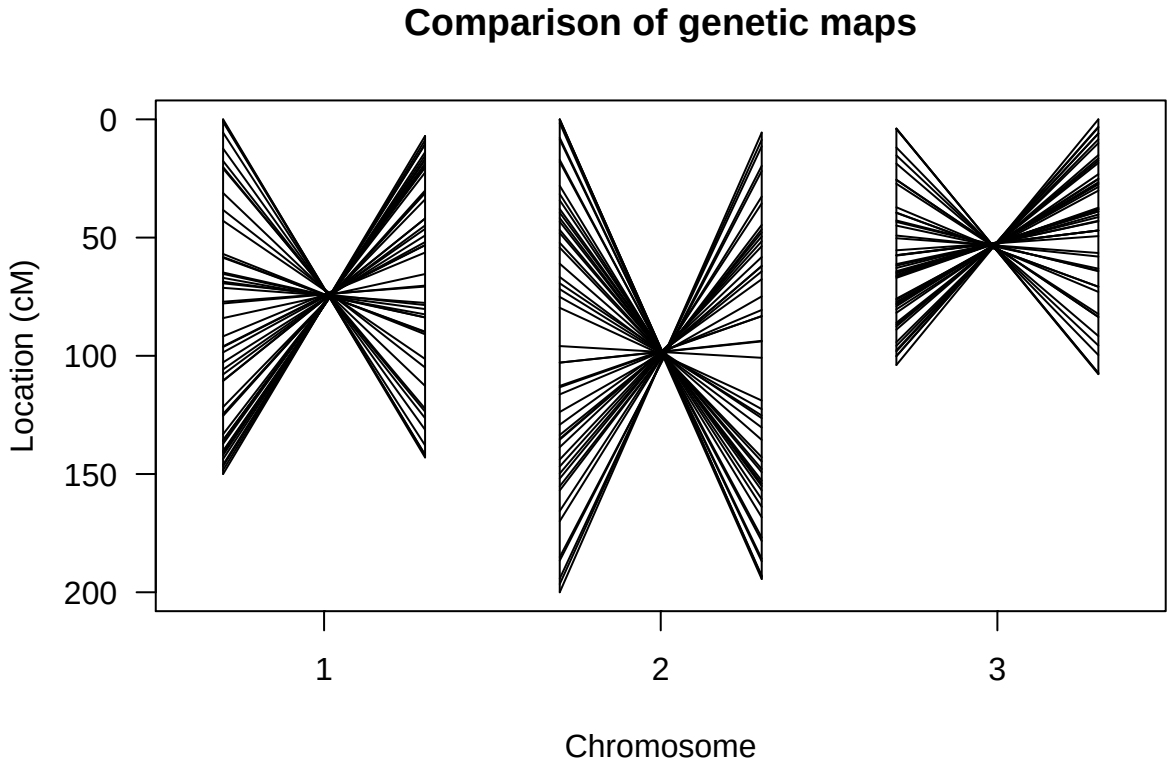

Figure 5: comparison of the genetic maps before flipping marker orientation

```
plot.map(cross.sim, cross.out)
```

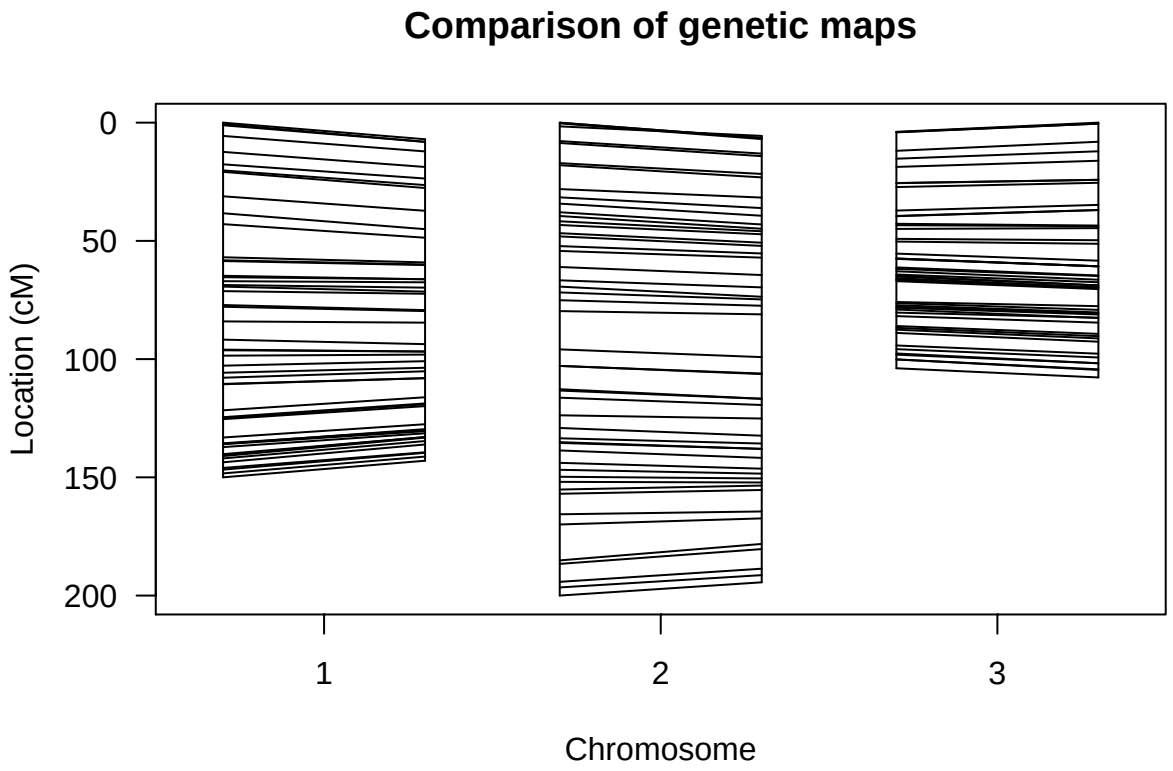

Figure 6: final comparison of the genetic maps. The horizontal lines connect the marker position of the simulated data (left) with the TSPmap re-ordered markers (right). Note that none of the marker order is nearly identical. Such differences are due to the simulated missing data and erroneous genotypes.