Creating a barley composite map with LPmerge

Jeffrey Endelman

LPmerge uses linear programming to create a composite, or consensus, map by merging linkage maps from different populations. LPmerge depends on the R package Rglpk, which provides an interface to the GNU Linear Programming Kit. Package Rglpk must be installed separately prior to using LPmerge.

To illustrate the use of LPmerge, I will merge a set of 6 barley linkage maps published by Muñoz-Amatriaín et al. (2011) and which can be downloaded as Supplemental Table 4 from https://www.agronomy.org/publications/tpg/articles/4/3/238. I copied the maps for populations HA, HO, MB, OWBH, SM1, and MH from the Excel file and saved each one as a .csv file.

The following code reads in each file and pulls out the markers from chromosome 4H, creating a list with 6 data frames, one for each population.

```
> map.names <- c("HA","HO","MB","OWBH","SM1",”MH”)
> Maps <- list()
> i <- 1
> for (i in 1:6) {
+     filename <- paste(map.names[i],".csv",sep="")
+     input <- read.csv(filename,header=T,as.is=T,check.names=F)
+     Maps[[i]] <- input[which(input$chr=="4H"),c(1,3)]
+ }
> names(Maps) <- map.names
> str(Maps)
List of 6
 $ HA  :'data.frame':   100 obs. of  2 variables:
  ..$ marker: chr [1:100] "2_0145" "2_1228" "1_0319" "2_1359" ...
  ..$ cM    : num [1:100] 0 3.71 7.42 14.88 14.88 ...
 $ HO  :'data.frame':   97 obs. of  2 variables:
  ..$ marker: chr [1:97] "2_0145" "2_1228" "1_0319" "1_0490" ...
  ..$ cM    : num [1:97] 0 4.27 10.68 12.81 20.32 ...
 $ MB  :'data.frame':   142 obs. of  2 variables:
  ..$ marker: chr [1:142] "3_0764" "3_1324" "1_0208" "2_1056" ...
  ..$ cM    : num [1:142] 0 0 2.15 2.15 27.86 ...
 $ OWBH:'data.frame':   206 obs. of  2 variables:
  ..$ marker: chr [1:206] "2_1117" "3_1324" "3_1486" "3_0140" ...
  ..$ cM    : num [1:206] 0 1.22 6.11 6.11 6.11 ...
 $ SM1 :'data.frame':   125 obs. of  2 variables:
  ..$ marker: chr [1:125] "2_0145" "2_1228" "1_0409" "1_0208" ...
  ..$ cM    : num [1:125] 0 2.18 2.18 2.18 5.44 ...
 $ MH  :'data.frame':   55 obs. of  2 variables:
  ..$ marker: chr [1:55] "2_1228" "1_0409" "1_1345" "1_0319" ...
  ..$ cM    : num [1:55] 0 0 1.1 4.4 24 ...
```

The lengths of the linkage maps range from 100 to 141 cM, with a mean of 125:

```
> print(link.map.lengths <- unlist(lapply(Maps,function(x){max(x$cM)})))
      HA       HO       MB     OWBH      SM1       MH 
100.3641 126.9359 138.1892 123.4095 141.4964 117.8506 
> mean(link.map.lengths)
[1] 124.7076
```

The following code loads the LPmerge package and calculates composite maps for *K* = 1 to 4, where *K* is the maximum interval size (see the reference manual or journal publication).

```
> library(LPmerge)
Loading required package: Rglpk
Loading required package: slam
Using the GLPK callable library version 4.48
Loading required package: Matrix
Loading required package: lattice
> unweighted <- LPmerge(Maps,max.interval=1:4)
[1] "# markers: 353"
[1] "# bins: 240"
[1] "# constraints: 1336"
[1] "Finding maximum feasible subsystem."
[1] "Eliminated following constraints to resolve marker order conflicts: "
[1] "Map HA: 2_0411 < 2_0210"
[1] "Map HO: 2_1070 2_0606 2_0302 < 2_0411"
[1] "Generating composite map."
[1] "Max.Interval = 1"
[1] "Composite map length: 123.38"
   map  RMSE
1   HA 10.43
2   HO  6.70
3   MB 23.30
4 OWBH  8.33
5  SM1 15.97
6   MH  4.54
7 mean 11.55
8   sd  6.95
[1] "Generating composite map."
[1] "Max.Interval = 2"
[1] "Composite map length: 124.43"
   map  RMSE
1   HA 11.05
2   HO  6.50
3   MB 22.49
4 OWBH  7.74
5  SM1 14.83
6   MH  5.42
7 mean 11.34
8   sd  6.45
[1] "Generating composite map."
[1] "Max.Interval = 3"
[1] "Composite map length: 152.8"
   map  RMSE
1   HA 39.07
2   HO 28.06
3   MB 11.92
4 OWBH 25.36
5  SM1 19.27
6   MH 33.71
7 mean 26.23
8   sd  9.78
[1] "Generating composite map."
[1] "Max.Interval = 4"
[1] "Composite map length: 152.81"
   map  RMSE
```

```
1   HA 39.59
2   HO 28.62
3   MB 12.11
4 OWBH 25.63
5  SM1 19.48
6   MH 34.28
7 mean 26.62
8   sd  9.93
```

The first part of the output provides statistics on the number of markers, number of composite bins, and the initial number of ordinal constraints. If there are inconsistencies in marker order between the linkage maps, LPmerge removes ordinal constraints to resolve these conflicts and prints the deleted constraints to the screen. In this example, two inequalities, one from HA and one from HO, were deleted. When multiple markers are listed on one side of the inequality, this indicates that all of them co-segregate in one composite map bin. The list of deleted inequalities can be used to re-examine the linkage maps or genotype data to better understand the source of the marker conflict.

The second part of the output shows the RMSE between each map and the composite map, along with the mean and sd RMSE, as a function of the max.interval parameter. The length of each composite map is also printed. In this example, the *K* = 2 composite map had the lowest mean and sd for RMSE, and its length was the most similar to the mean of the component linkage maps. A peek at the output for this map looks like

```
> head(unweighted[[2]])
  marker composite HA HO   MB OWBH  SM1 MH
1 2_1117      0.00 NA NA   NA 0.00   NA NA
2 3_0764      1.22 NA NA 0.00   NA   NA NA
3 3_1324      1.22 NA NA 0.00 1.22   NA NA
4 2_0145      2.87  0  0   NA   NA 0.00 NA
5 2_1056      3.37 NA NA 2.15   NA   NA NA
6 1_0208      4.42 NA NA 2.15   NA 2.18 NA
```

The total error minimized by LPmerge can also use different weights for the linkage maps if some maps are thought to have higher accuracy. For example, one could weight each map by its population size because larger populations produce higher resolution maps:

```
> pop.size <- c(54,94,93,82,92,91)  #(HA,HO,MB,OWBH,SM1,MH)
> weighted <- LPmerge(Maps,max.interval=1:4,weights=pop.size)
[1] "# markers: 353"
[1] "# bins: 240"
[1] "# constraints: 1336"
[1] "Finding maximum feasible subsystem."
[1] "Eliminated following constraints to resolve marker order conflicts: "
[1] "Map HA: 2_0411 < 2_0210"
[1] "Map HO: 2_1070 2_0606 2_0302 < 2_0411"
[1] "Generating composite map."
[1] "Max.Interval = 1"
[1] "Composite map length: 124.15"
   map  RMSE
1   HA 12.85
2   HO  6.10
```

```
3   MB 20.80
4 OWBH  6.83
5  SM1 13.40
6   MH  6.81
7 mean 11.13
8   sd  5.73
[1] "Generating composite map."
[1] "Max.Interval = 2"
[1] "Composite map length: 123.63"
   map  RMSE
1   HA 10.73
2   HO  6.66
3   MB 22.82
4 OWBH  7.99
5  SM1 14.79
6   MH  4.98
7 mean 11.33
8   sd  6.60
[1] "Generating composite map."
[1] "Max.Interval = 3"
[1] "Composite map length: 152.8"
   map  RMSE
1   HA 40.74
2   HO 29.70
3   MB 13.00
4 OWBH 26.94
5  SM1 20.66
6   MH 35.41
7 mean 27.74
8   sd 10.00
[1] "Generating composite map."
[1] "Max.Interval = 4"
[1] "Composite map length: 152.8"
   map  RMSE
1   HA 39.92
2   HO 28.92
3   MB 12.18
4 OWBH 25.93
5  SM1 19.80
6   MH 34.70
7 mean 26.91
8   sd 10.02
```

An examination of this output shows that the $K = 1$ weighted map has even lower mean and sd RMSE compared to the best unweighted composite map.