R/mpMap Workshop

Part 4: Advanced Topics

Emma Huang
TAMU, 3 Sep. 2015

CHOOSE YOUR OWN ADVENTURE

CHOOSE FROM HUNDREDS OF POSSIBLE ENDINGS!

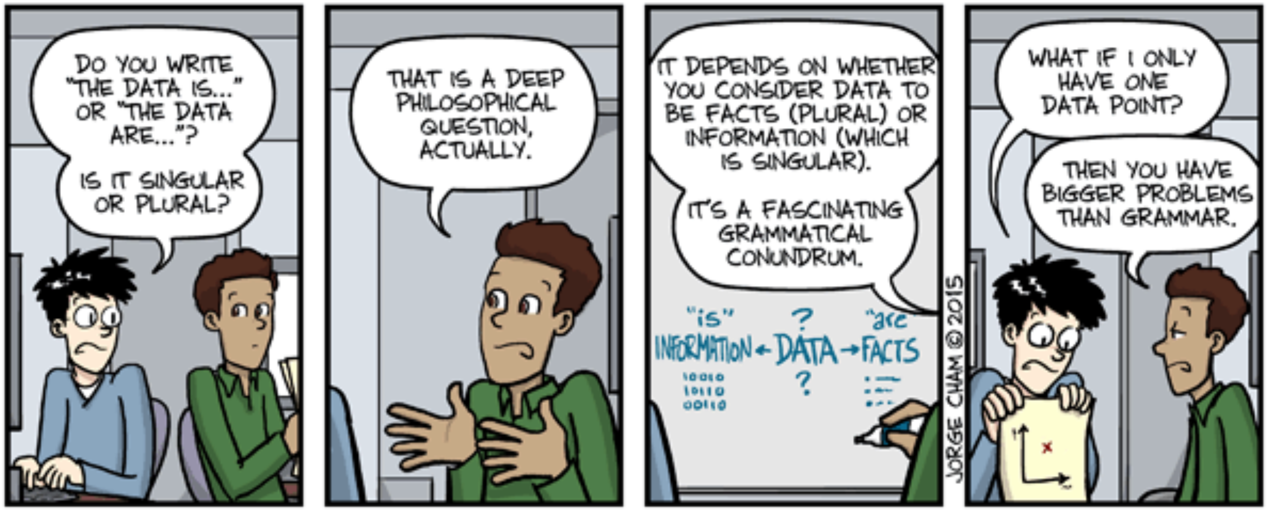
SOFTWARE PROJECT
SUCCESS OR FAILURE

You have the power…

Plan

10:30-11:30

- Part 4: Advanced Topics (45 min)
    - Imputation
    - Selective phenotyping
    - Simulation/Recombination
    - Visualization
- Exercises (10 min)
- Questions (5 min)


DO YOU WRITE "THE DATA IS..." OR "THE DATA ARE..."?
IS IT SINGULAR OR PLURAL?
THAT IS A DEEP PHILOSOPHICAL QUESTION, ACTUALLY.
IT DEPENDS ON WHETHER YOU CONSIDER DATA TO BE FACTS (PLURAL) OR INFORMATION (WHICH IS SINGULAR).
IT'S A FASCINATING GRAMMATICAL CONUNDRUM.
"is"
?
"are"
INFORMATION ← DATA → FACTS
?
10010
10110
00110
JORGE CHAM © 2015
WHAT IF I ONLY HAVE ONE DATA POINT?
THEN YOU HAVE BIGGER PROBLEMS THAN GRAMMAR.

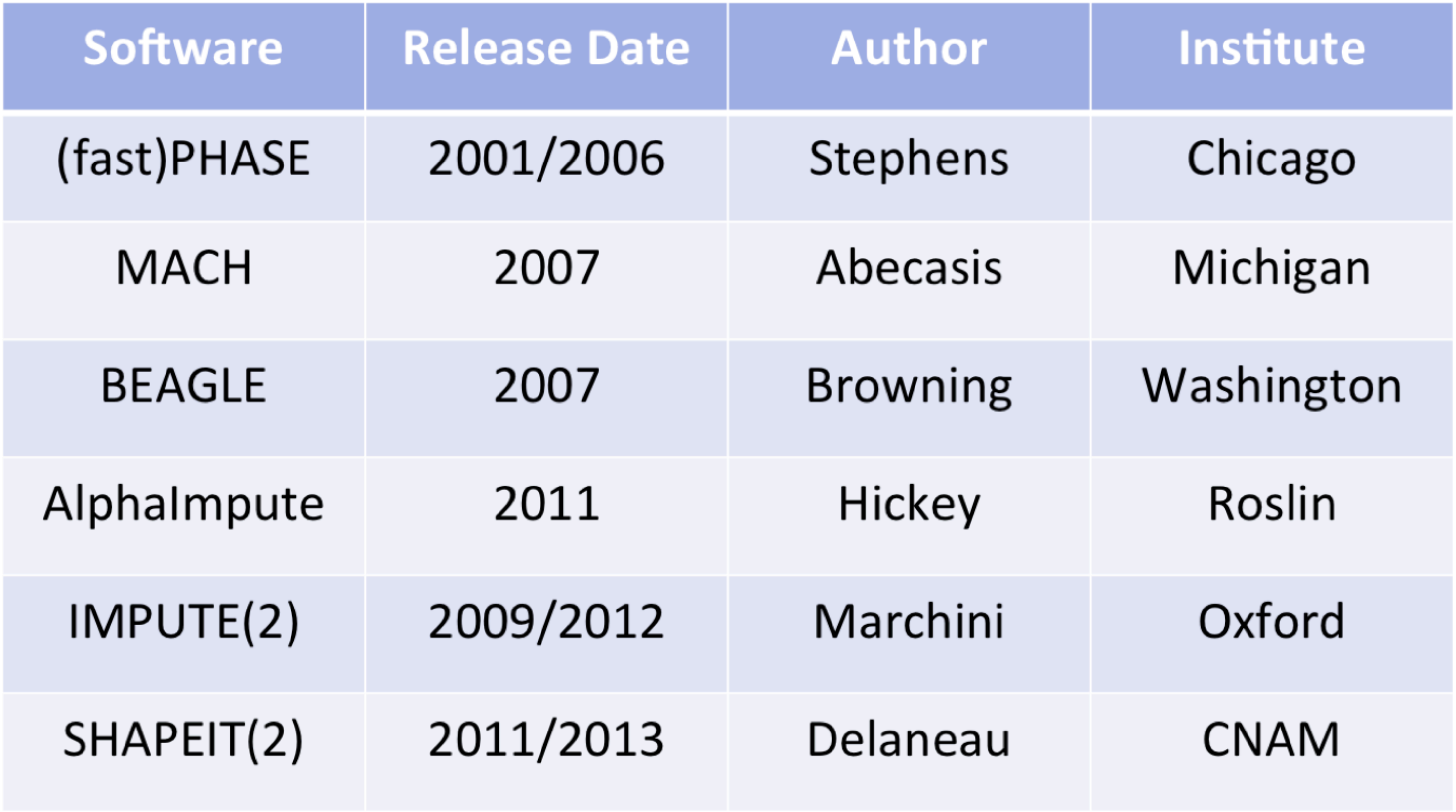
WWW.PHDCOMICS.COM

Imputation of Missing Data

Missing Data

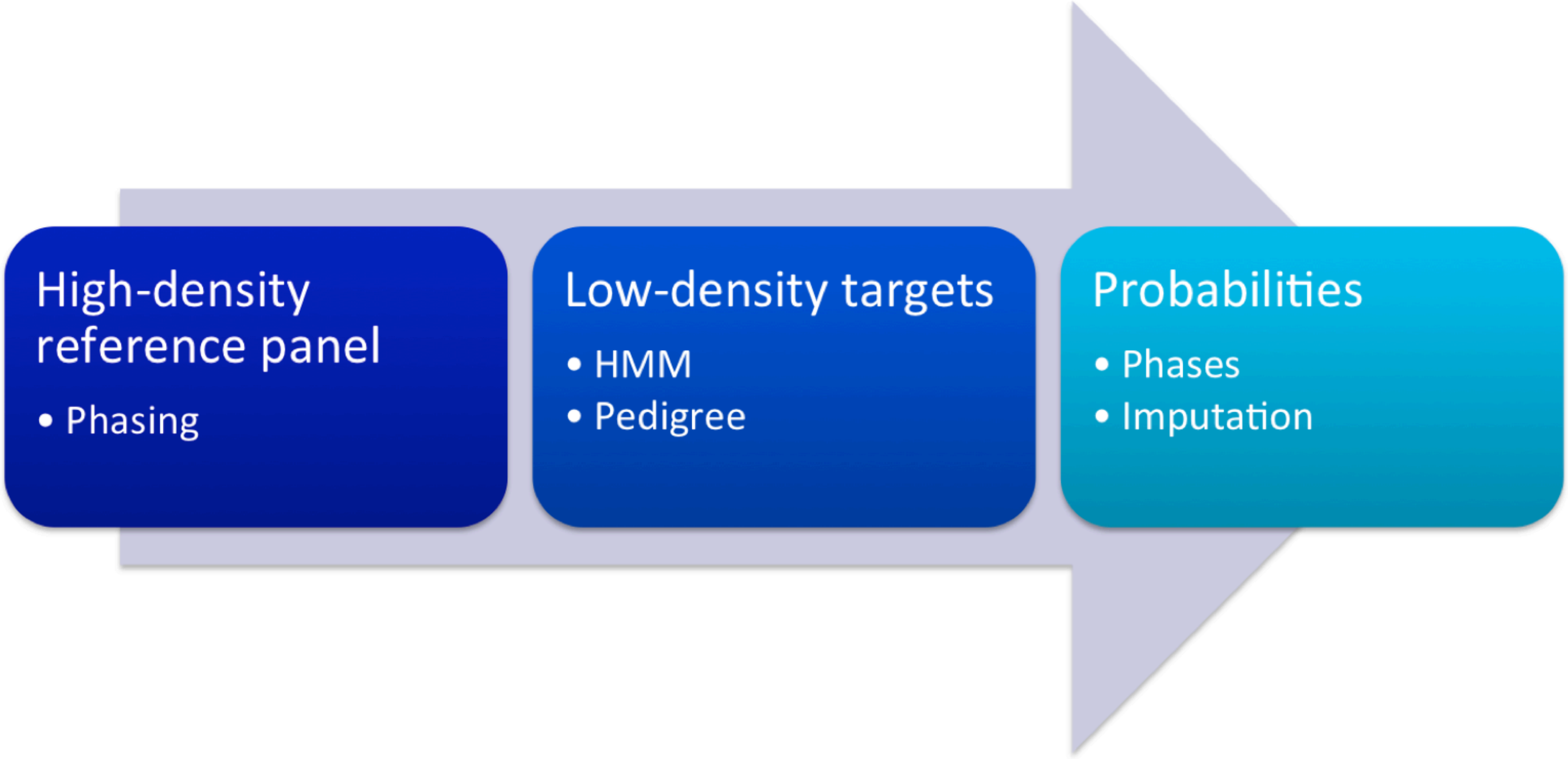
- Causes
    - GBS - alignment
    - Quality
    - SNP - hets not called
    - Different platforms

Typical Approaches



Software	Release Date	Author	Institute
(fast)PHASE	2001/2006	Stephens	Chicago
MACH	2007	Abecasis	Michigan
BEAGLE	2007	Browning	Washington
AlphaImpute	2011	Hickey	Roslin
IMPUTE(2)	2009/2012	Marchini	Oxford
SHAPEIT(2)	2011/2013	Delaneau	CNAM

High-density reference panel



High-density reference panel

- Phasing

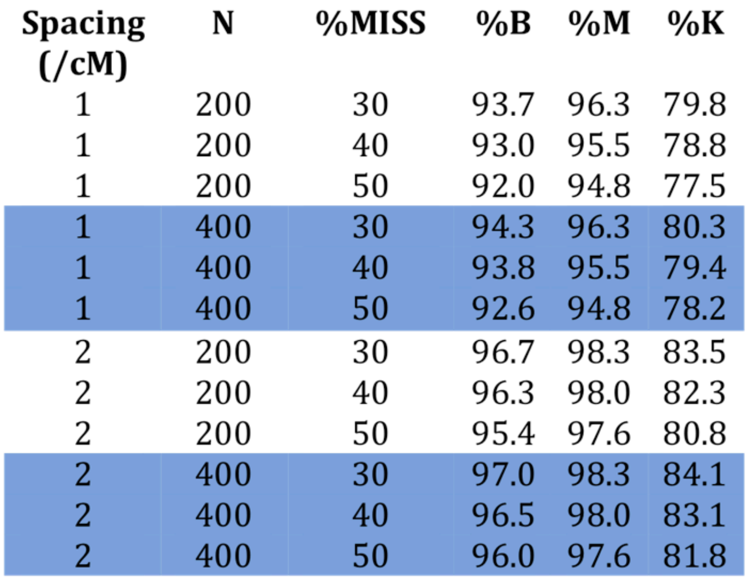
Low-density targets

- HMM
- Pedigree

Probabilities

- Phases
- Imputation

Good performance



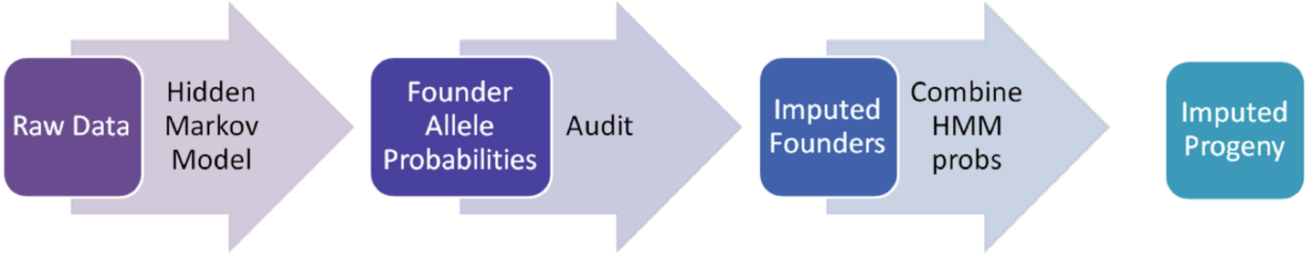
Spacing (/cM)	N	%MISS	%B	%M	%K
1	200	30	93.7	96.3	79.8
1	200	40	93.0	95.5	78.8
1	200	50	92.0	94.8	77.5
1	400	30	94.3	96.3	80.3
1	400	40	93.8	95.5	79.4
1	400	50	92.6	94.8	78.2
2	200	30	96.7	98.3	83.5
2	200	40	96.3	98.0	82.3
2	200	50	95.4	97.6	80.8
2	400	30	97.0	98.3	84.1
2	400	40	96.5	98.0	83.1
2	400	50	96.0	97.6	81.8

- But what if the reference panel is incomplete?

Simplest solution: get more data

- Higher coverage
- Different platform
- More replicates
- …
- But sometimes that's not possible

Impute using all data


Raw Data

Hidden Markov Model

Founder Allele Probabilities

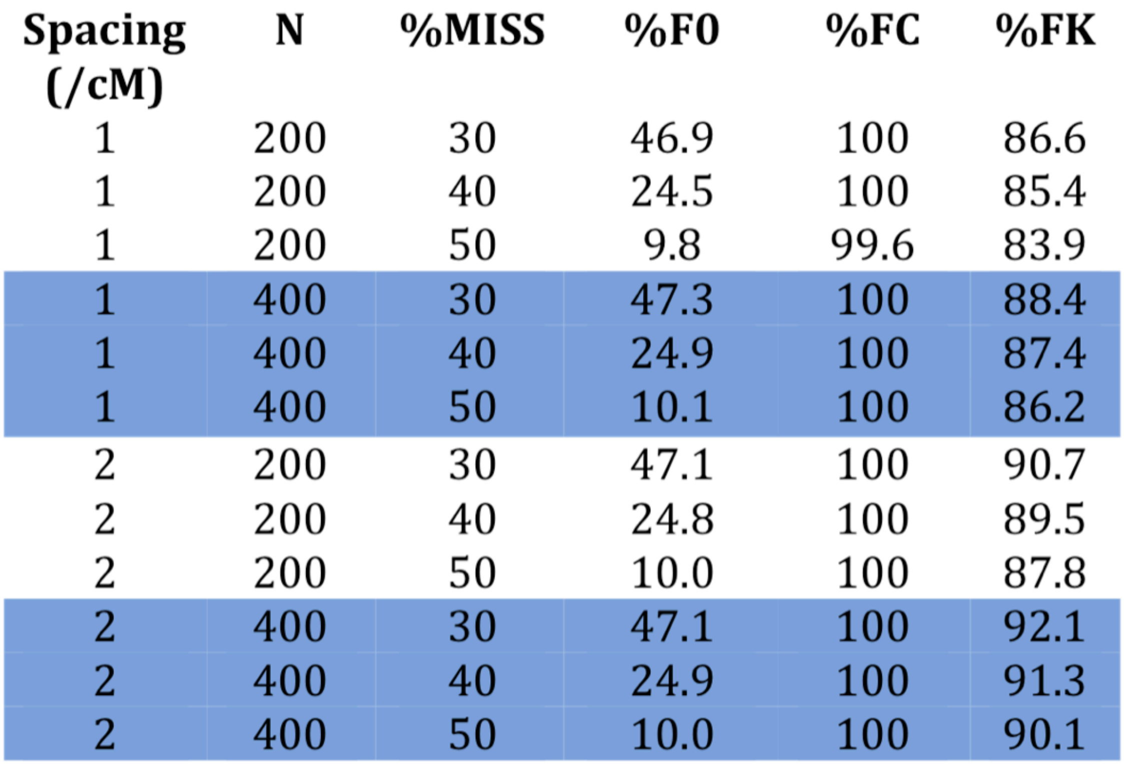
Audit

Imputed Founders

Combine HMM probs

Imputed Progeny

Simulation results



Spacing (/cM)	N	%MISS	%FO	%FC	%FK
1	200	30	46.9	100	86.6
1	200	40	24.5	100	85.4
1	200	50	9.8	99.6	83.9
1	400	30	47.3	100	88.4
1	400	40	24.9	100	87.4
1	400	50	10.1	100	86.2
2	200	30	47.1	100	90.7
2	200	40	24.8	100	89.5
2	200	50	10.0	100	87.8
2	400	30	47.1	100	92.1
2	400	40	24.9	100	91.3
2	400	50	10.0	100	90.1

Dataset simulated with missing data

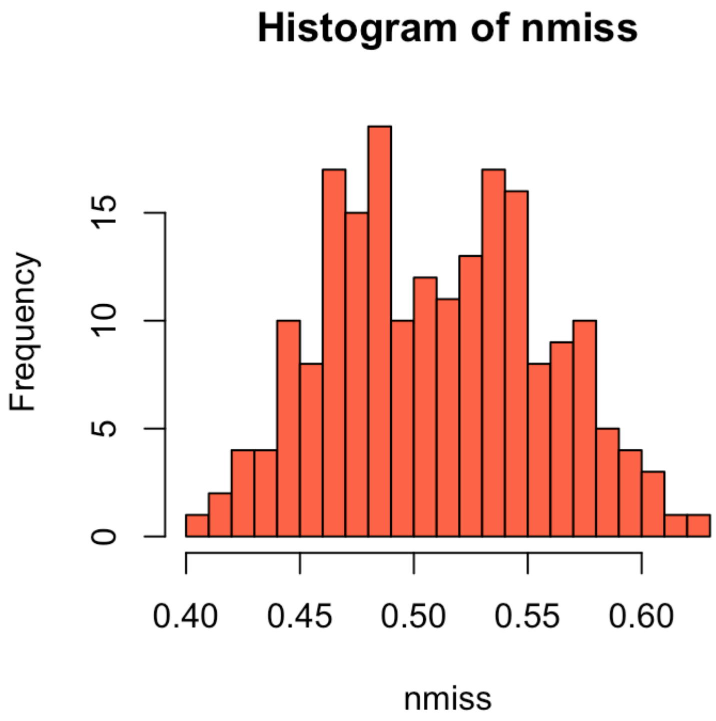
```r
load('MissingData.RData')
table(apply(missdat$founders, 2, function(x) sum(is.na(x))))
```

```
## 
##  0  1  2  3  4  5 
## 15 29 31 19  5  2
```

```r
nmiss <- apply(missdat$finals, 1, function(x) sum(is.na(x))/length(x))
```

50% missing for all lines; 25% missing per founder

```
hist(nmiss, breaks=20, col="tomato")
```



Implementation of imputation

- mpimpute
  - options for founders, finals, or both
  - relies on founder probability calculation

```
impdat <- mpimpute(missdat)
```

```
## [1] "No chromosomes specified, will default to all"
## Using map groupings for groups. Remove map object if you want to regroup.
##  --Read the following data:
##   200  individuals
##   101  markers
##   2  phenotypes
```

How much could we impute?

```r
table(apply(impdat$founders, 2, function(x) sum(!is.na(x))))
```

```
## 
##   8 
## 101 
```

```r
nmissi <- apply(impdat$finals, 1, function(x) sum(is.na(x))/length(x))
sum(nmissi>0)
```

```
## [1] 0
```

How accurate was the imputation?

```
  sum(is.na(impdat$founders))
```

```
## [1] 0
```

```
  sum(impdat$founders!=dat$founders)
```

```
## [1] 0
```

```
  sum(is.na(impdat$finals))/prod(dim(impdat$finals))
```

```
## [1] 0
```

```
  sum(impdat$finals!=dat$finals, na.rm=T)/sum(is.na(missdat$finals))
```

```
## [1] 0.08916084
```

In practice

- May want to test on your known data
  - Mask out some percentage, try imputation and estimate accuracy
- Affected by
  - marker density/spacing
  - sample size
  - type of genotyping platform
  - level of heterozygosity, etc.

Selective Phenotyping

Costs

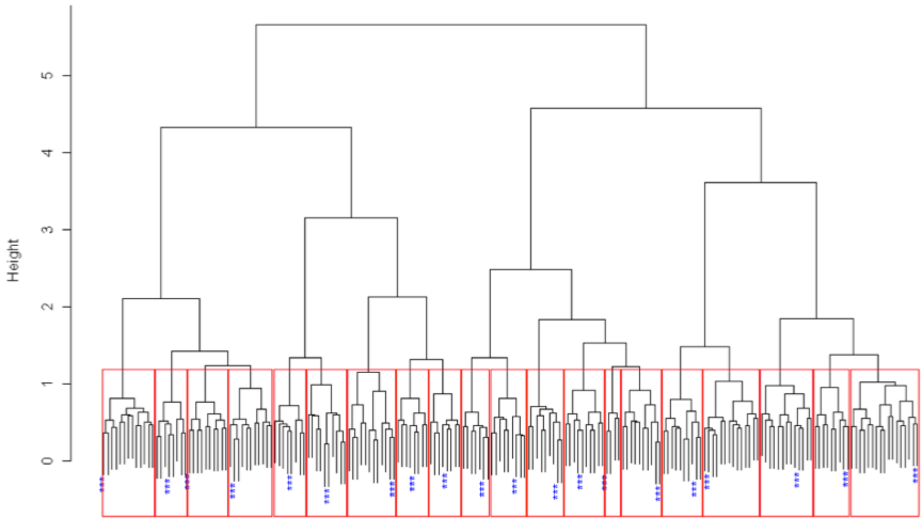
- Phenotyping has overtaken genotyping in cost
  - done many times
  - at multiple scales
  - in multiple environments
  - for multiple traits
  - …
- So how do we best select a sample for phenotyping from a large set of genotyped individuals?

Goal

- A selection method which
  - is general, flexible and robust
  - best captures the genetic information of the population
  - maximized diversity, avoids genetic duplication
- Previous options (Jin et al. 2004, Jannink 2005) focused on specific designs, could not handle missing data

SPCLUST

- Step 1: Compute pairwise distances between all individuals
- Step 2: Cluster distances into k groups
- Step 3: Select representative from each cluster



2-stage SPCLUST

- Multiple stages of selection to refine QTL position
- Genomewide -> candidate gene level
- Selected lines will
    - have higher genetic diversity, so
    - are more likely to have recombination
    - and better resolve QTL location

2-stages

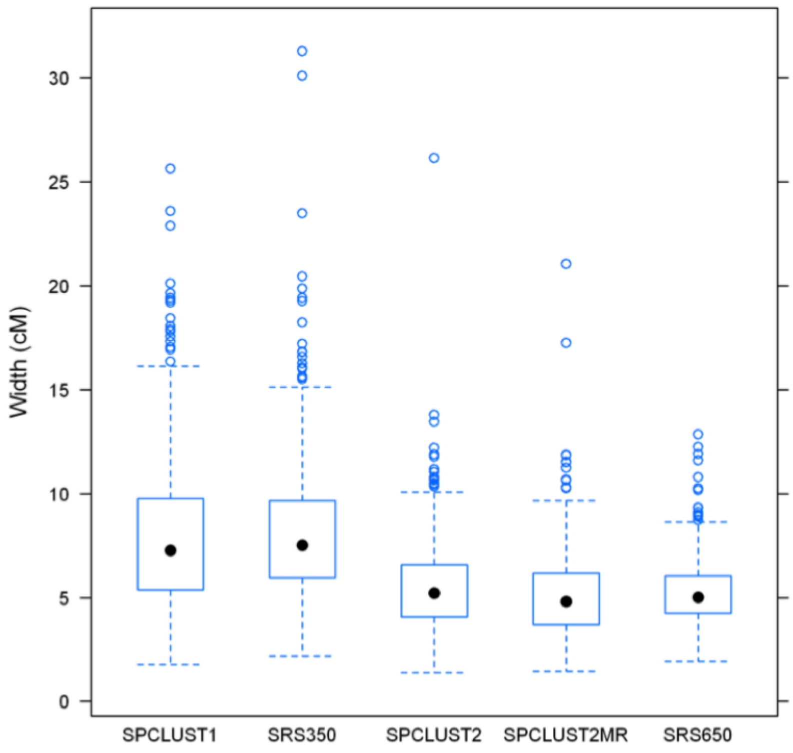
Stage 1:
Selection of lines based on full genome

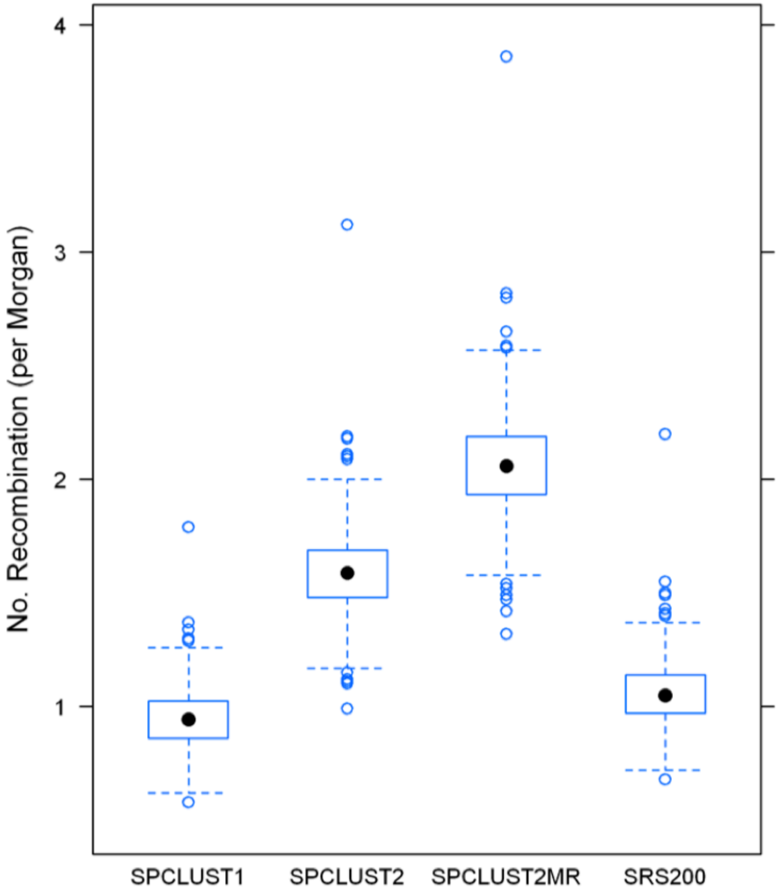
Detection of broad QTL regions

Stage 2:
Selection of lines based on QTL region genotypes

Refine QTL position

QTL Support intervals





Implementation

- R package spclust
- Functions to
    - compute distance (spdist)
    - select lines (spclust, single- or multi-stage)
    - visualize (plot.spclust)

Example

```
library(spclust)
load('SimulatedSP.RData')
selLines <- spclust(dat, nlines=20, method="average")
```

```
## [1] "No chromosomes specified, will default to all"
## Using map groupings for groups. Remove map object if you want to regroup.
##  --Read the following data:
##   200  individuals
##   255  markers
##   2  phenotypes
## No required lines input; will only select a single-stage sample
```

Plot of output

```
plot(selLines, type=2)
```

Simulation

What have we seen so far?

- General MAGIC simulation
  - varying map
  - varying pedigree
  - varying QTL
- Missing data

What can you do with this?

- Assess imputation quality in your data
- Compare different designs
    - number of generations of advanced intercrossing
    - DH vs. RIL
    - MAGIC vs. NAM
- Test power for different approaches
- Generate empirical significance thresholds
- Estimate power for your map/data/founders
- See how theory compares to reality

Comparison of designs

- Generate different pedigrees
- Generate data from them
- Compare number of recombinations, size of haplotype blocks

```r
ped4 <- sim.mpped(4, 3, 200) # MAGIC4RIL
ped8 <- sim.mpped(8, 30, 200) # MAGIC8RIL, 30 funnels
ped8ai2 <- sim.mpped(8, 1, 200, iripgen=2) # MAGIC8AI2RIL
ped26nam <- generateNAMpedigree(26, 100) #NAMRIL
```

Whole genome data

- AlphaMPSim (Hickey et al. 2014)
- Written in Fortran/R
- Faster and more memory efficient than using sim.mpcross
- For very large-scale simulations (>30K markers)

Estimation of power

- Need to set up larger scale scripts
- Generate multiple datasets
  - Could keep observed founder genotypes, map, new progeny
  - Could keep observed progeny genotypes as well - new phenotype
  - Depends on how generalizable results need to be
- Analyze each dataset as you would observed data
- How often are QTL of different sizes detected?
- See Kover et al. (2009) for more details on procedure

Recombination

Counting recombination events

```
load('datfinalPart2.RData')
## Based just on highest probability allele
nrecEst <- lapply(mppEst$estfnd, function(x)
  apply(x, 1, function(y) return(sum(diff(y[!is.na(y)])!=0))))
mean(rowSums(do.call("cbind", nrecEst)))
```

```
## [1] 12.44
```

```
## Errors in the map can cause additional recombination events
load('Part2.RData')
nrecTrue <- lapply(mppTrue$estfnd, function(x)
  apply(x, 1, function(y) return(sum(diff(y[!is.na(y)])!=0))))
mean(rowSums(do.call("cbind", nrecTrue)))
```

```
## [1] 11.954
```

Alternate method of counting

```r
## Based on forward-backward algorithm with penalty
source('nrec.R')
mean(nrec(mppEst, penalty=0)$totrec)
```

```
## [1] 13.797
```

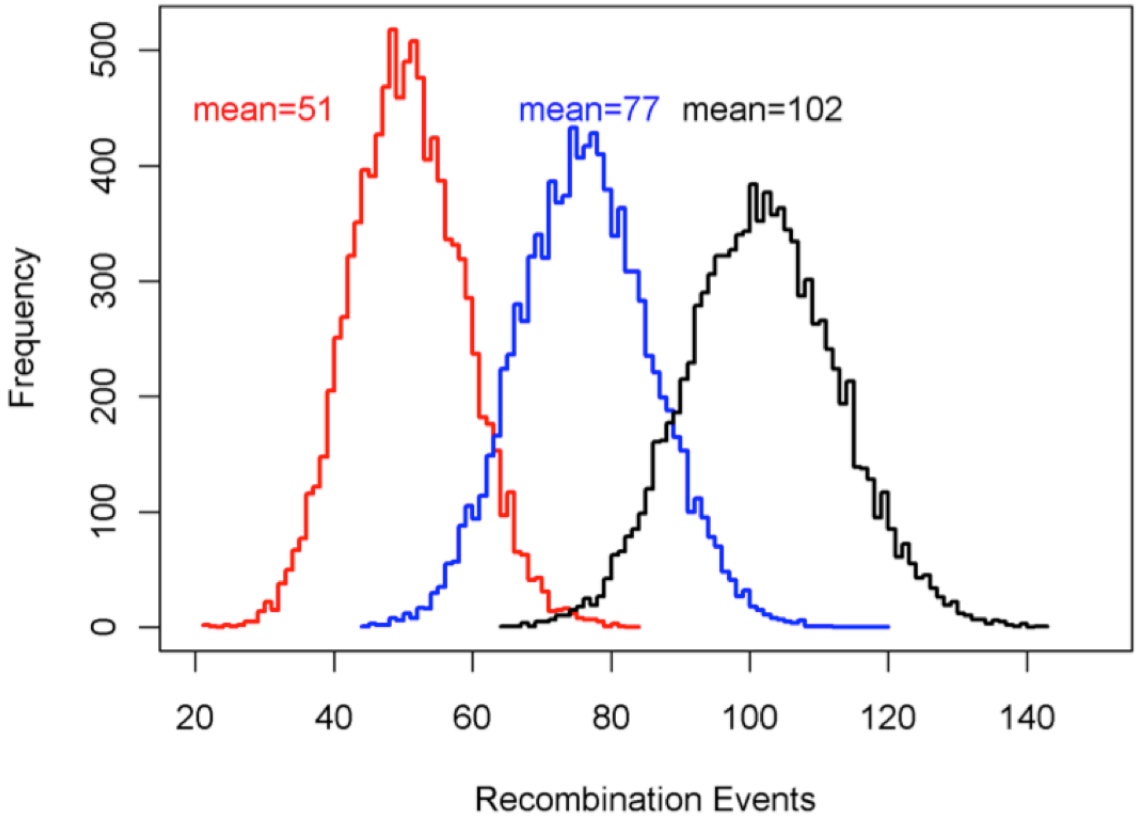
```r
mean(nrec(mppEst, penalty=1)$totrec)
```

```
## [1] 12.055
```

```r
mean(nrec(mppEst, penalty=2)$totrec)
```

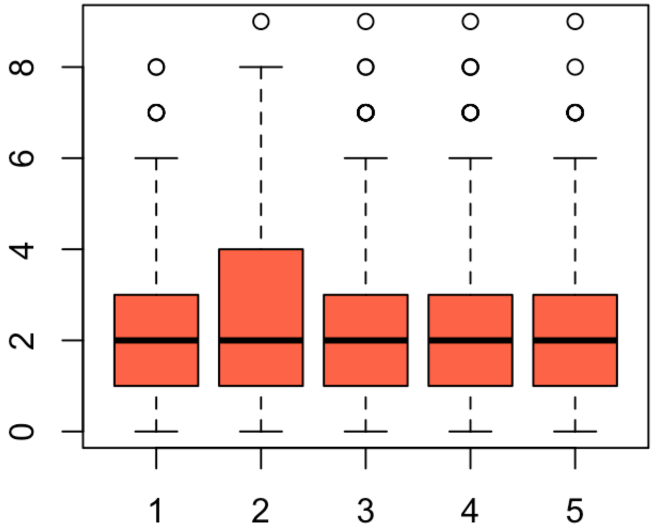
```
## [1] 10.941
```

Simulation of recombination events



Counting per chromosome

```
nr <- nrec(mppEst, penalty=1)
boxplot(do.call("cbind", nr$nrec), col="tomato")
```



QTL mapping with recombination events

```
mppEst$pheno$nrec <- nrec(mppEst, penalty=1)$totrec
mprec <- mpIM(object=mppEst, responsename="nrec", ncov=0)
## No QTL found - but possible in real data
```

Visualization

Additional libraries

- ggplot2
- lattice
- RCircos
- Heatplus
- LDheatmap
- …

mpMap/Interactive

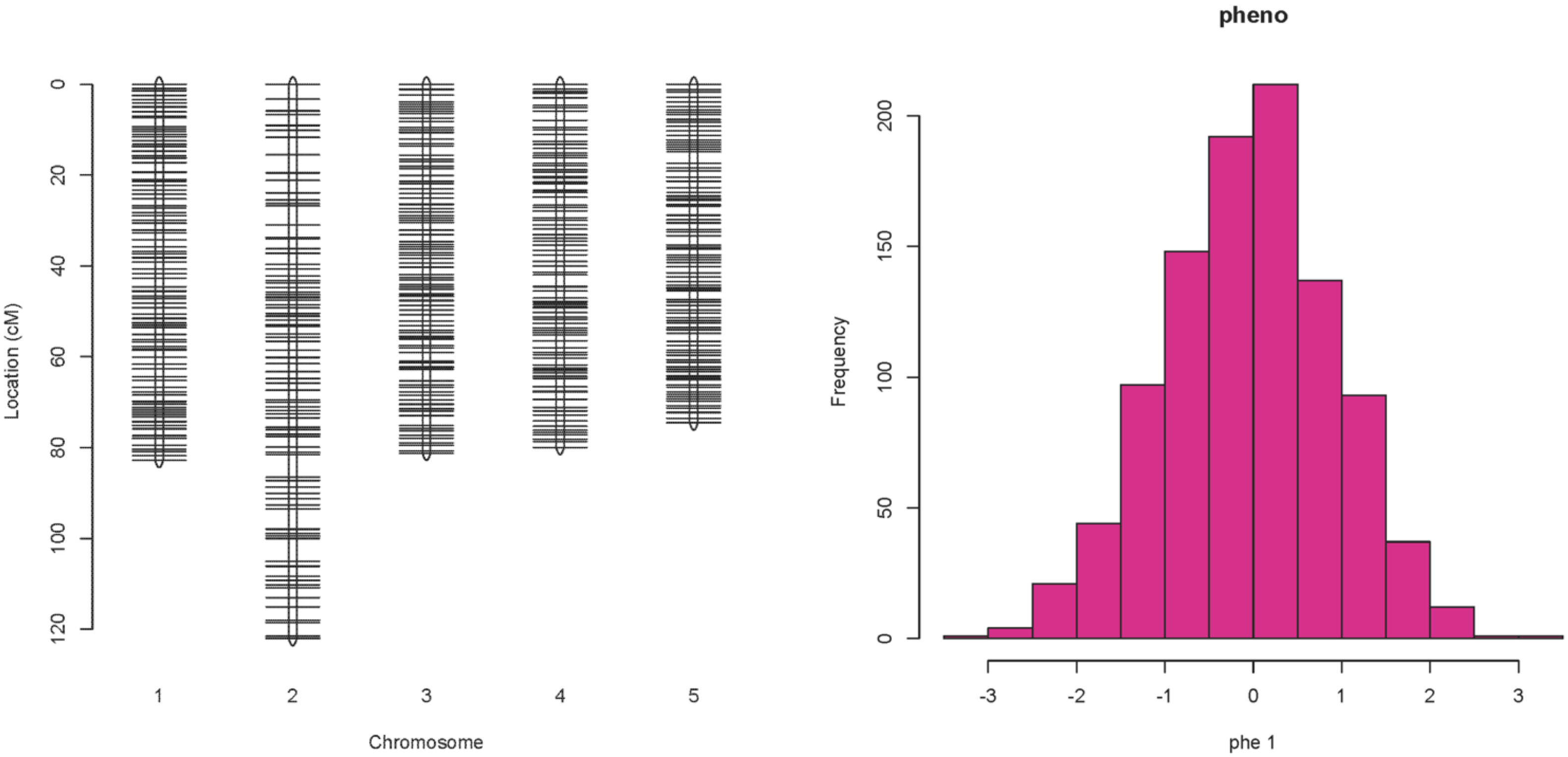
- Works on OS/X and Windows
- Uses:
  - Grouping markers, combining groups
  - Removing markers
  - Flipping blocks of markers
  - Re-ordering chunks of markers
- Note: currently only works for mpcross objects, but could relatively easily be extended to more general crosses

Plot functions for most objects

- plot.mpcross
  - Linkage map
  - Histogram of phenotype(s)
  - RF/LOD heatmap
- plot.mpprob
  - Percent of chromosome inherited from each founder
  - Haplotype mosaics for each chromosome
  - Founder inheritance across genome
- plot.mpqtl
  - QTL profile
  - Support interval

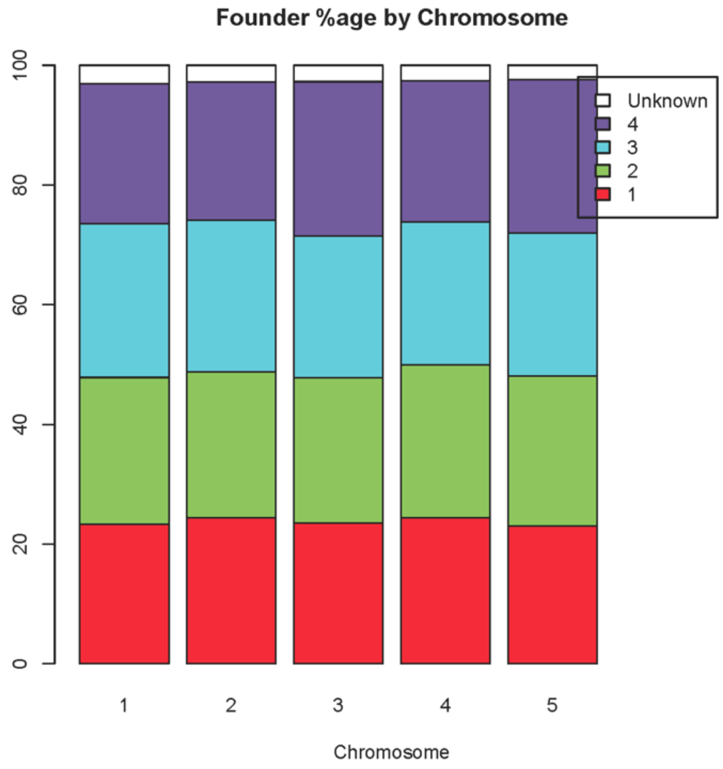
plot.mpcross

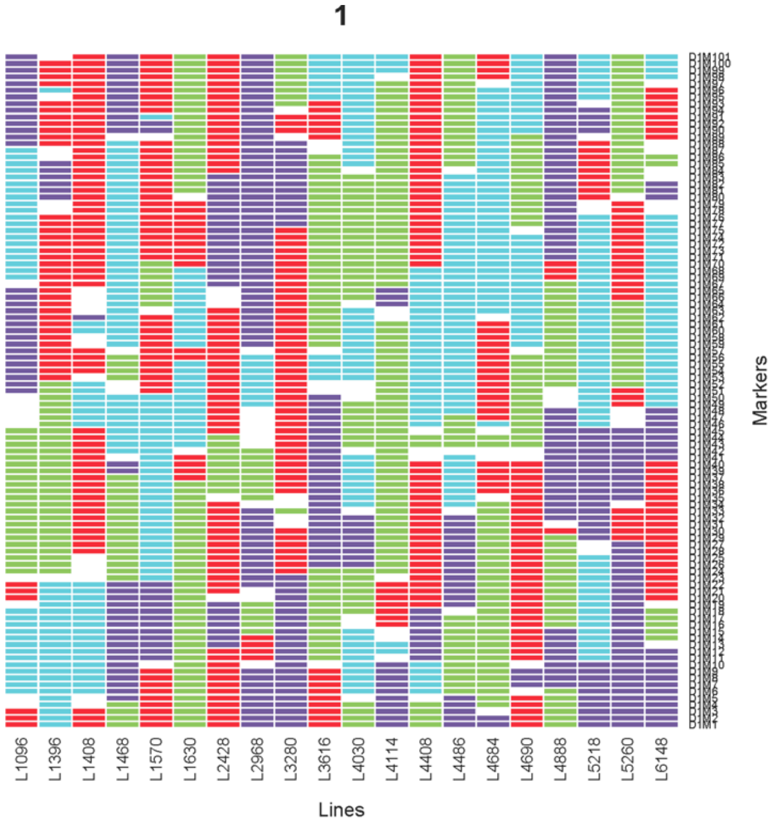
```
load('datfinalPart2.RData')
plot(datfinal)
```



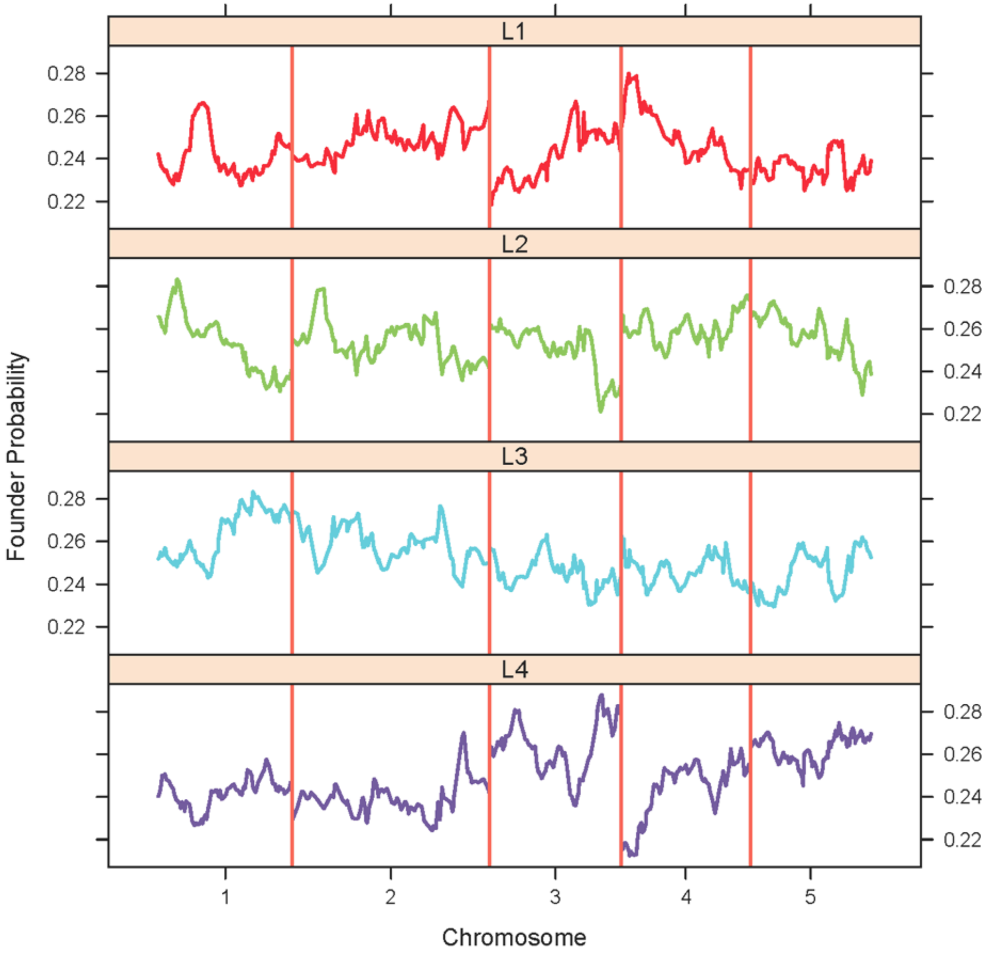
plot.mpprob

```
mpp <- mpprob(datfinal, program="qtl")
plot(mpp)
```





plot.mpprob (cont'd)



Comparisons of maps - validation

- mapcomp function
  - subsets down to common markers/chromosomes
  - compares positions of markers with the same names
  - identifies markers with conflicting chromosomes
- summary.mapcomp
  - summarizes number of markers in each map and in common
  - identifies duplicated markers in each map
  - correlations between chromosomes in each map
- plot.mapcomp
  - scatterplot of markers positions for the two maps

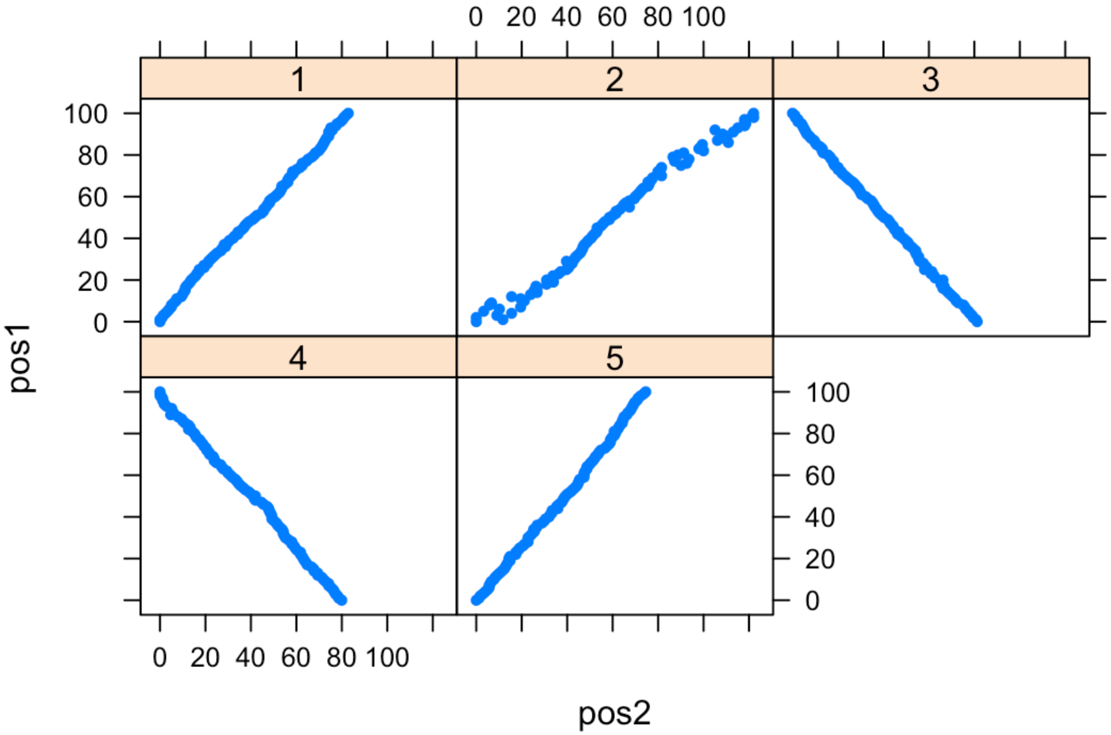
mapcomp

```
load('Part2.RData')
mc <- mapcomp(dat, datfinal)
summary(mc)
```

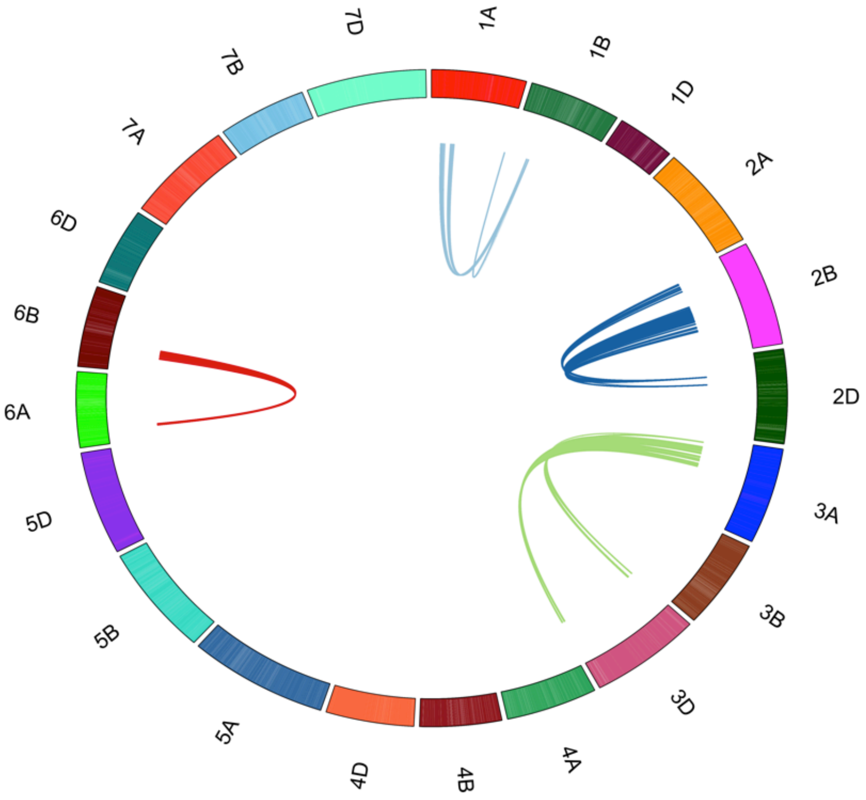
```
## Number of markers in map1 is  505
## Number of markers in map2 is  505
## Number of common markers is  505
## Number of duplicated markers in map1 is  0
## Number of duplicated markers in map2 is  0
## Number of markers with differing chromosomes between maps is  0
## Correlations between chromosomes are:
## 0.999142 0.991306 -0.999554 -0.9988986 0.9987963
```

plot.mapcomp

```
plot(mc)
```



Interactions



Circle plot code

```
library(RCircos)
source('CircularIntx.R')
pmatrix <- matrix(runif(505*505, 0, 1), nrow=505)
plotCircIntx(pmatrix, threshold=1e-4, map=datfinal$map, file="CircEx.png")
```

References (1/2)

Jin et al. 2004, Selective phenotyping for increased efficiency in genetic mapping studies. Genetics 168:2285-2293. doi: 10.1534/genetics.104.027524

Jannink 2005, Selective phenotyping to accurately map quantitative trait loci. Crop Science 45:901-908. doi: 10.2135/cropsci2004.0278

Hickey et al. 2014, AlphaMPSIM: flexible simulation of multi-parent crosses. Bioinformatics 30:2686-2688. doi: 10.1093/bioinformatics/btu206

Huang et al. 2013, Selecting subsets of genotyped experimental populations for phenotyping to maximize genetic diversity. TAG 126: 379-388. doi: 10.1007/s00122-012-1986-4

References (2/2)

Huang et al. 2014, Efficient imputation of missing markers in low-coverage genotyping-by-sequencing data from multiparental crosses. Genetics 197:401-404. doi: 10.1534/genetics.113.158014

Kover et al. 2009, A multiparent advanced generation inter-cross to fine-map quantitative traits in Arabidopsis thaliana. PLoS Genetics doi: 10.1371/journal.pgen.1000551

Questions

Contact me: b.emma.huang@gmail.com

github.com/behuang/mpMap for updates

The End!