NC STATE UNIVERSITY

MAPpoly

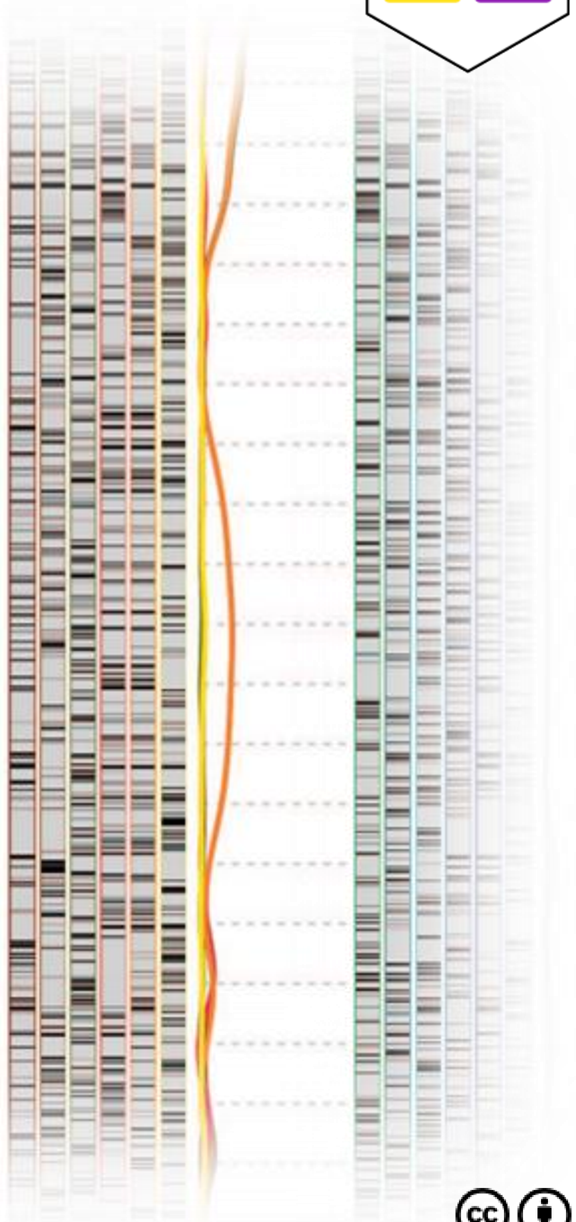
MAPpoly training section

Marcelo Mollinari
mmollin@ncsu.edu

April 22, 2021

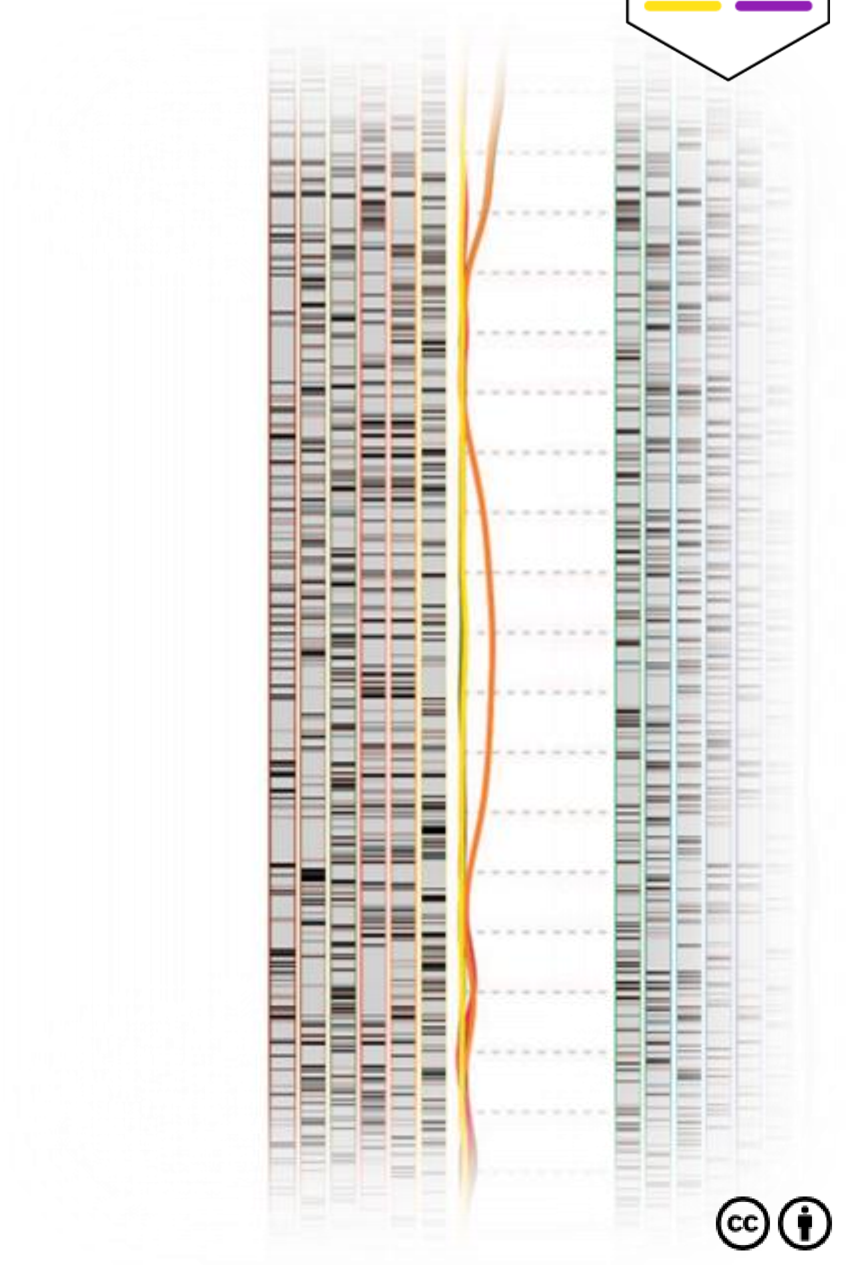
Outline

- Installation
- Reading/import datasets
- Filtering
- Two-point analysis
- Grouping
- Ordering
- Phasing
- Modeling errors/genotype probabilities
- Final checking
- Map summary/results
- Genotype probabilities
- Preferential pairing/haplotype probabilities
- Hexaploid scripts/results



Acknowledgments

- Gabriel Gesteira – USP
- Guilherme Pereira – CIP/UVF
- Augusto Garcia – USP
- Craig Yencho – NCSU
- Zhao-Bang Zeng – NCSU



Initial notes about MAPpoly

- Primary intention was implementing a program to construct genetic maps in polyploids with high ploidy levels (i.e. > 4: sugarcane, sweetpotato, etc.)
- Linkage Analysis and Haplotype Phasing in Experimental Autopolyploid Populations with High Ploidy Level Using Hidden Markov Models (Mollinari and Garcia, 2019)
- Obtain conditional probabilities for QTL analysis
- It does not model double reduction: double reduced markers/individuals are treated as missing data

NC STATE UNIVERSITY BILL & MELINDA GATES foundation GT4SP GENOMIC TOOLS FOR SWEETPOTATO IMPROVEMENT SweetGAINS TOOLS FOR POLYPLOIDS USDA NIFA

Installation

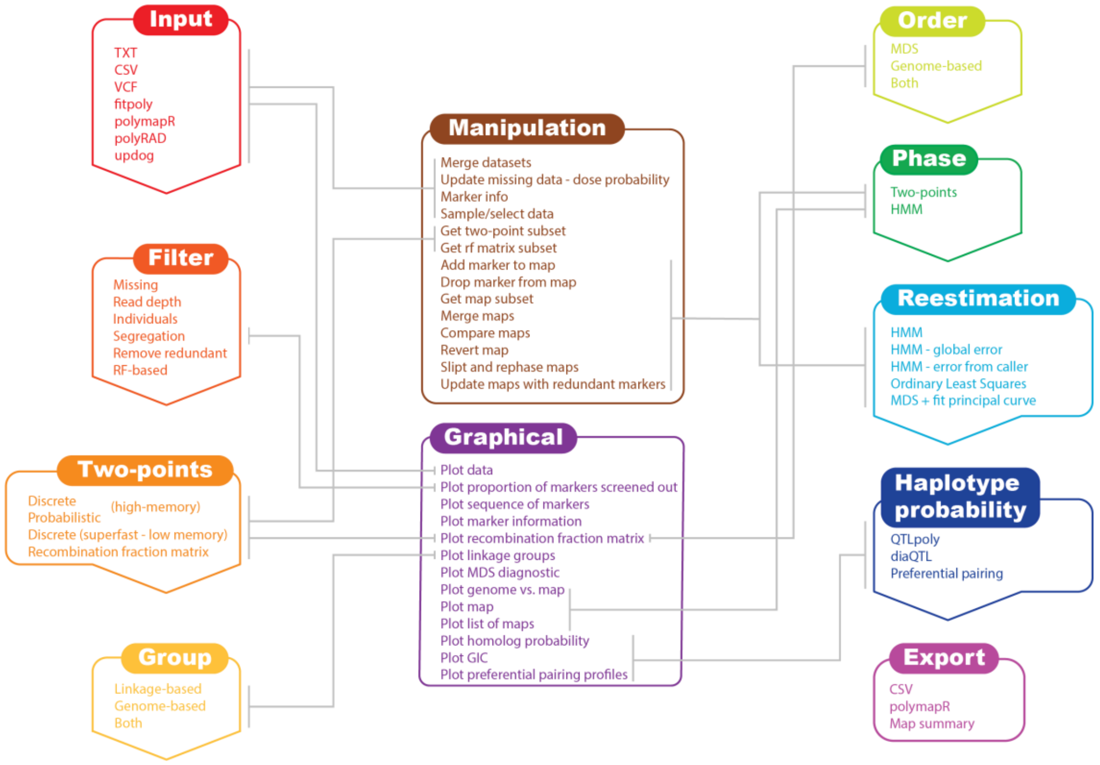
- From CRAN (stable version)

```
install.packages("mappoly")
```

- From GitHub (development version)

```
install.packages("devtools")
devtools::install_github("mmollina/mappoly", dependencies=TRUE)
```

MAPpoly workflow



This project is funded by USDA NIFA Specialty Crop Research Initiative Award # 2020-51181-32156 (09/01/20 - 08/31/24)

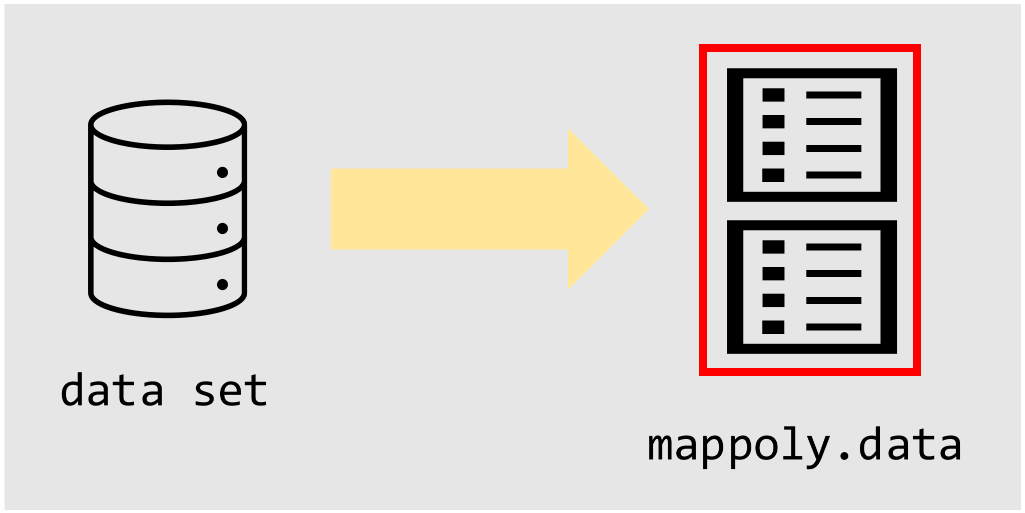
Reading/importing data sets

Supported datasets

- MAPpoly files
  - Discrete dosage
  - Dosage probability
- CSV files
- fitPoly files
  - Dosage Probability
- VCF files
  - Discrete dosage
  - Dosage probability

Supported R objects

- updog objects
- polyRAD objects
- polymapR
  - datasets
  - maps



Reading/importing datasets

- MAPpoly data reading functions

1. MAPpoly

```
dat.mappoly <- read_geno(file.in = 'path_to_input_file')
```

2. MAPpoly with genotype probabilities

```
dat.mappoly.prob <- read_geno_prob(file.in = 'path_to_input_file)
```

3. CSV

```
dat.csv <- read_geno_csv(file.in = 'path_to_input_file', ploidy = 6)
```

4. fitPoly, supports genotype probabilities

```
dat.fitpoly <- read_fitpoly(file.in = 'path_to_input_file', ploidy = 6,
                            parent1 = 'P1', parent1 = 'P2')
```

5. VCF files, supports genotype probabilities (PL field)

```
dat.vcf <- read_vcf(file.in = 'path_to_input_file', ploidy = 6,
                    parent1 = 'P1', parent1 = 'P2',
                    min.gt.depth = 0, min.av.depth = 0)
```

Reading/importing datasets

- MAPpoly data importing functions

1. updog

```
dat.updog <- import_from_updog(object = input_multidog_object)
```

2. polyRAD

```
dat.polyRAD <- polyRAD::Export_MAPpoly(object = 'polyrad_object')
```

3. polymapR

Importing data

```
dat.polymapr <- import_data_from_polymapR(input.data = polymapR_screened_data,
                                          ploidy = 6, parent1 = 'P1',
                                          parent1 = 'P2')
```

Importing maps

```
map.polymapr <- import_phased_maplist_from_polymapR(maplist = phased_maplist,
                                                    mappoly.data = mappoly.data)
```

Reading/importing datasets

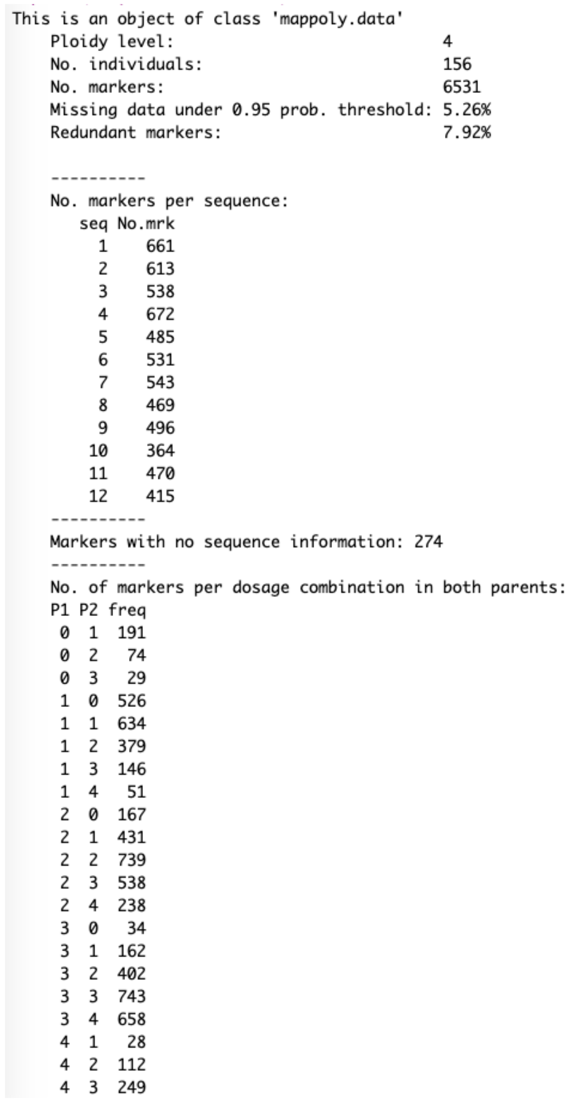
- Example: reading fitPoly files

```
## Downloading data from GitHub
library("mappoly")
setwd("SCRI/MAPpoly/tetra")
address <- "https://github.com/mmollina/SCRI/raw/main/data/fitpoly_tetra_call/B2721_scores.zip"
download.file(url = address, destfile = "B2721_scores.zip")
unzip(tempfl, files = "B2721_scores.dat")
## Reading fitPoly data
dat <- read_fitpoly(file.in = "B2721_scores.dat",
                    ploidy = 4,
                    parent1 = "Atlantic",
                    parent2 = "B1829",
                    verbose = TRUE)
source("get_solcap_snp_pos.R")
plot(dat)
print(dat, detailed = TRUE)
```

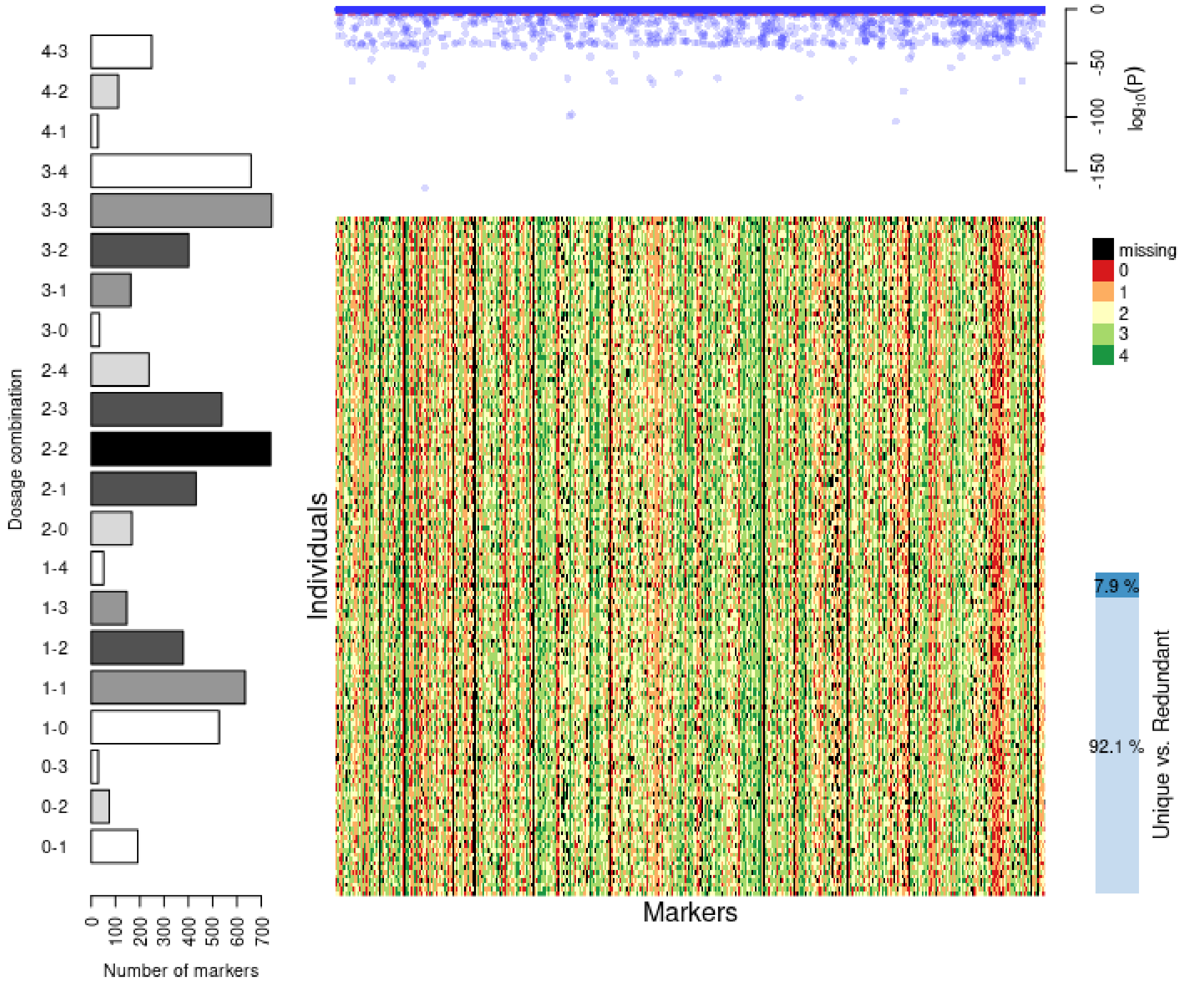
Genotype calling using fitPoly:
https://github.com/mmollina/SCRI/blob/main/data/fitpoly_tetra_call/B2721_fitpoly_call.R
Adding *Solanum tuberosum* genome v4.03 information:
https://github.com/mmollina/SCRI/blob/main/MAPpoly/get_solcap_snp_pos.R

NC STATE UNIVERSITY

Inspecting data loaded from fitPoly files



```
This is an object of class 'mappoly.data'
    Ploidy level:                       4
    No. individuals:                    156
    No. markers:                        6531
    Missing data under 0.95 prob. threshold: 5.26%
    Redundant markers:                  7.92%

    ----------
    No. markers per sequence:
       seq No.mrk
         1    661
         2    613
         3    538
         4    672
         5    485
         6    531
         7    543
         8    469
         9    496
        10    364
        11    470
        12    415
    ----------
    Markers with no sequence information: 274
    ----------
    No. of markers per dosage combination in both parents:
    P1 P2 freq
     0  1  191
     0  2   74
     0  3   29
     1  0  526
     1  1  634
     1  2  379
     1  3  146
     1  4   51
     2  0  167
     2  1  431
     2  2  739
     2  3  538
     2  4  238
     3  0   34
     3  1  162
     3  2  402
     3  3  743
     3  4  658
     4  1   28
     4  2  112
     4  3  249
```

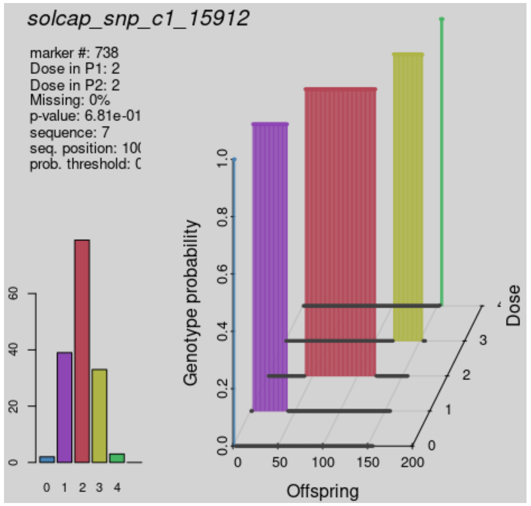


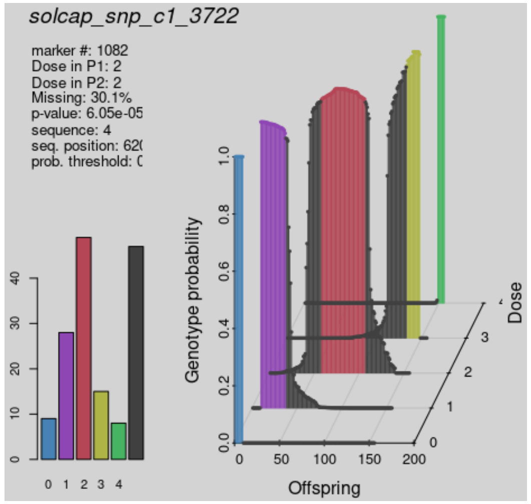
MAPpoly

Inspecting specific marker

```
plot_mrk_info(dat, 738)
print_mrk(dat, 738)
```

```
plot_mrk_info(dat, "solcap_snp_c1_3722")
print_mrk(dat, "solcap_snp_c1_3722")
```





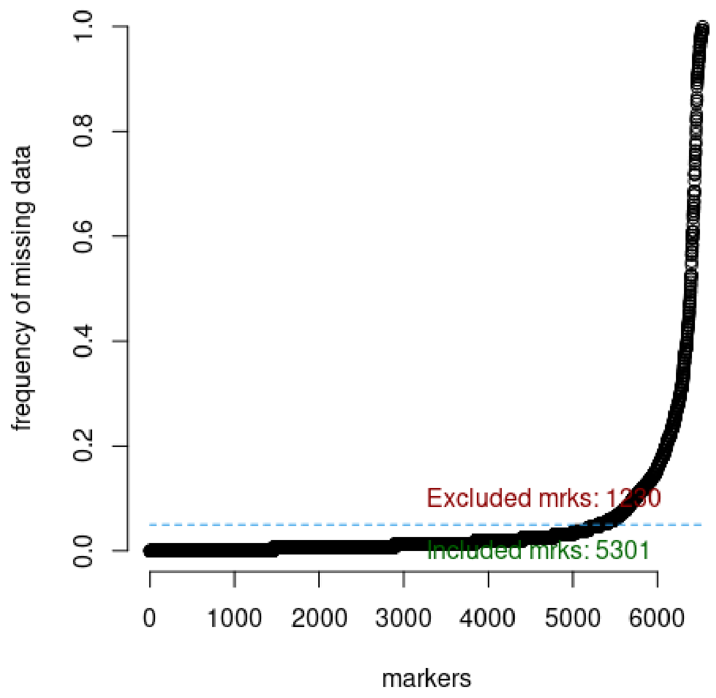
```
 solcap_snp_c1_15912
----------------------------------
 dosage P1:  2
 dosage P2:  2
----
 dosage distribution
  0   1   2   3   4 mis
  2  39  79  33   3   0
----
 expected polysomic segregation
          0          1          2          3          4
0.02777778 0.22222222 0.50000000 0.22222222 0.02777778
----------------------------------
```

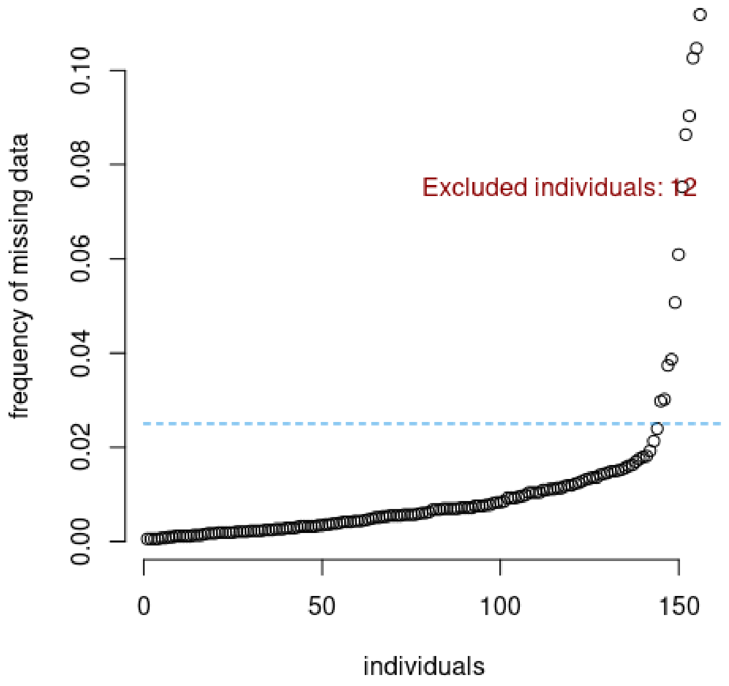
```
 solcap_snp_c1_3722
----------------------------------
 dosage P1:  2
 dosage P2:  2
----
 dosage distribution
  0   1   2   3   4 mis
  9  28  49  15   8  47
----
 expected polysomic segregation
          0          1          2          3          4
0.02777778 0.22222222 0.50000000 0.22222222 0.02777778
----------------------------------
```

Filtering by missing data

```r
dat <- filter_missing(input.data = dat,
                      type = "marker",
                      filter.thres = 0.05)
```

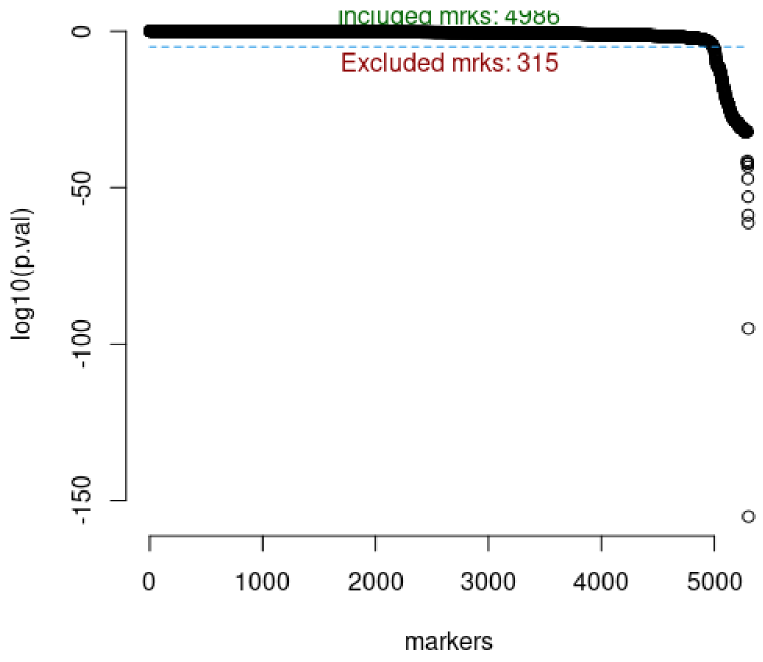
```r
dat <- filter_missing(input.data = dat,
                      type = "individual",
                      filter.thres = 0.025)
```

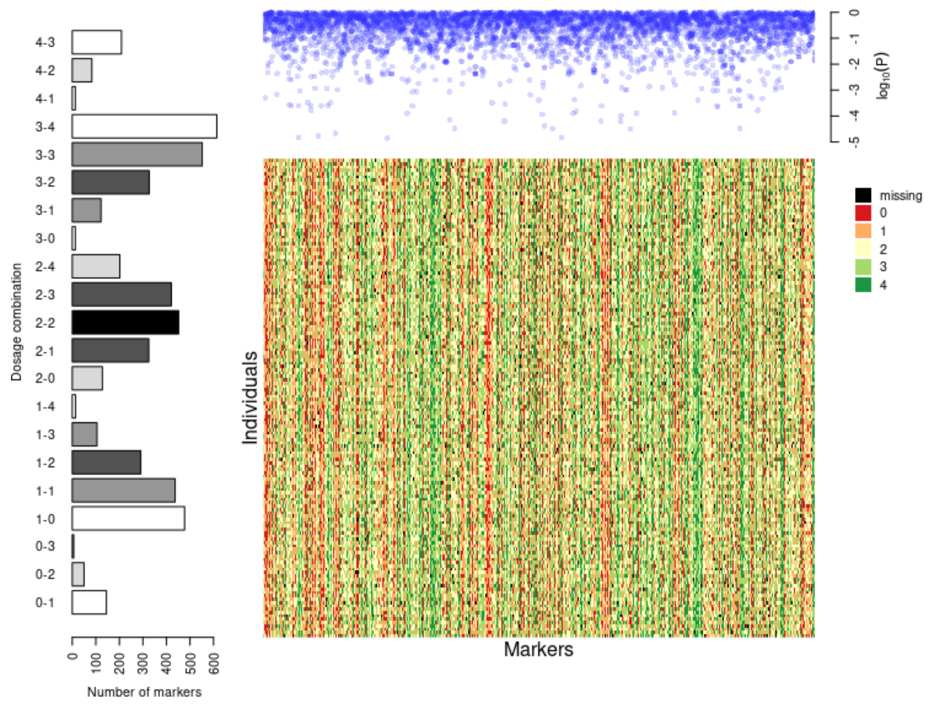




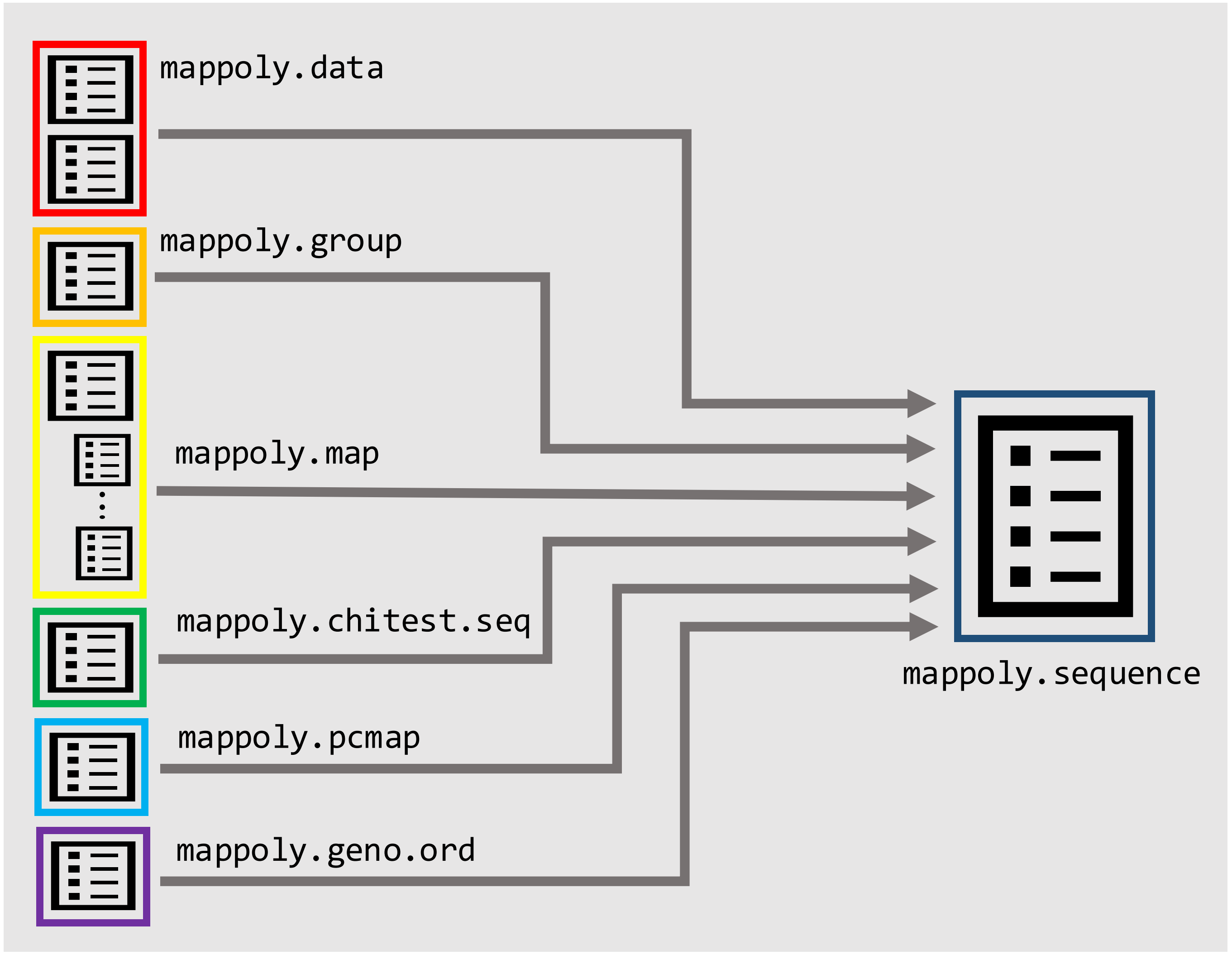
Filtering by segregation distortion

```
s <- filter_segregation(input.data = dat, chisq.pval.thres = 0.05/dat$n.mrk)
s <- make_seq_mappoly(s)
plot(s)
```





Making a sequence of markers

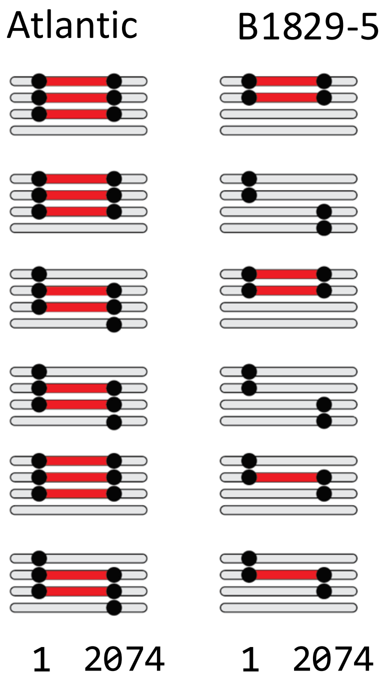


NC STATE UNIVERSITY

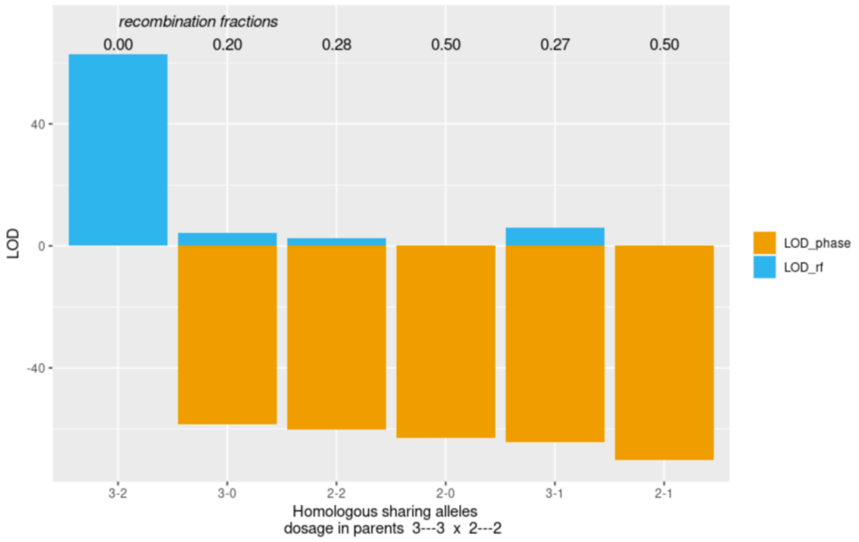
MAPpoly

Two-point analysis

```r
## Two-point analysis
nc <- parallel::detectCores() - 1
tpt10 <- est_pairwise_rf(s10, ncpus = nc, est.type = "disc")
tpt10$pairwise[90:92]
round(tpt10$pairwise[[91]], 2)
plot(tpt10, first.mrk = 1, second.mrk = 2074)
```



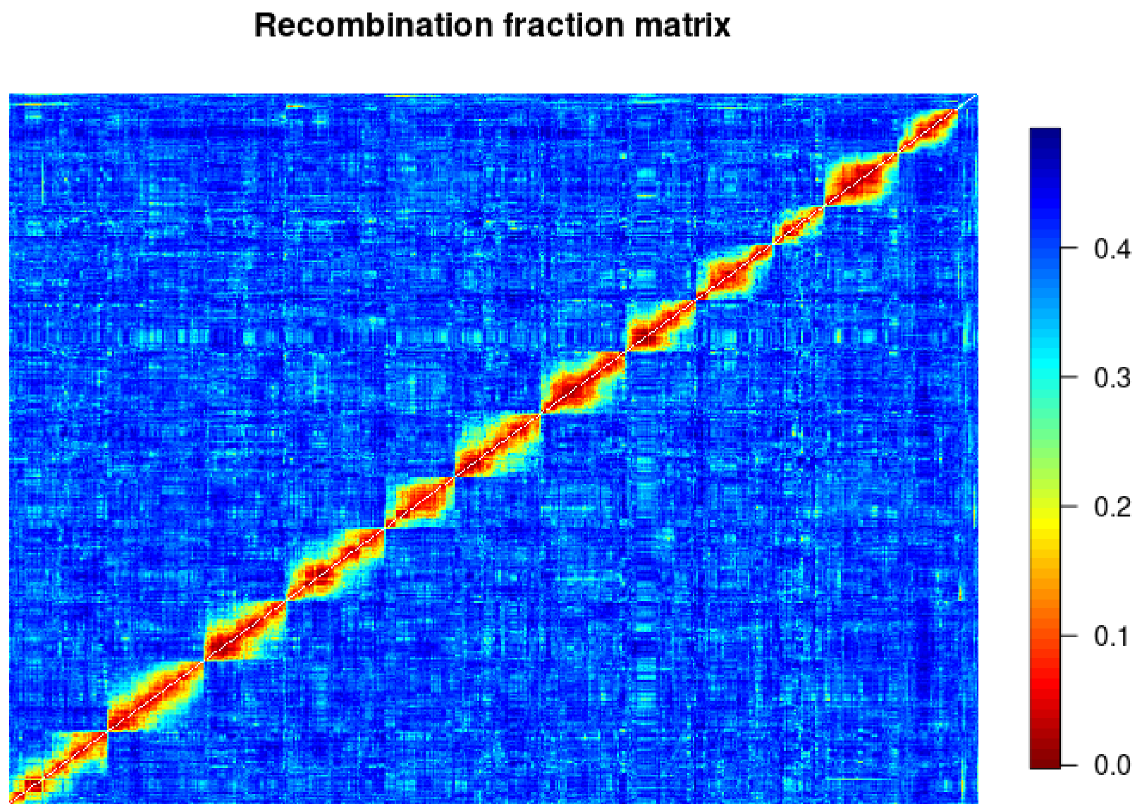
```
      LOD_ph   rf LOD_rf
3-2     0.00 0.00  62.86
3-0   -58.46 0.20   4.40
2-2   -60.17 0.28   2.69
2-0   -62.86 0.50   0.00
3-1   -64.29 0.27   5.89
2-1   -70.18 0.50   0.00
```



MAPpoly

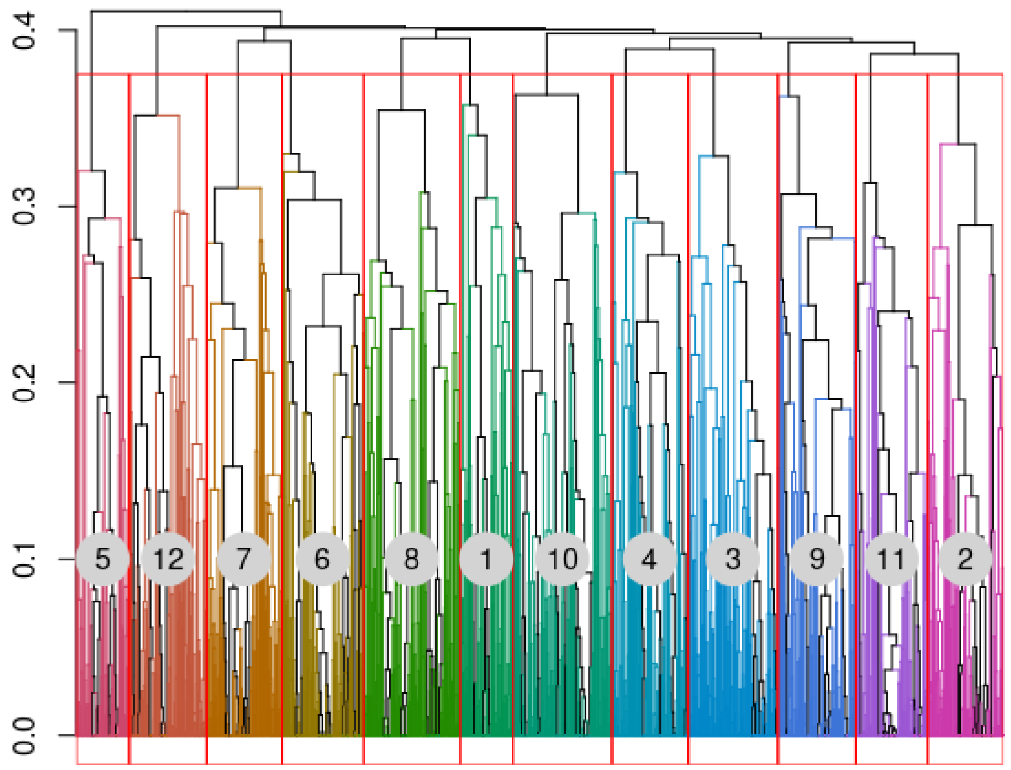
Recombination fraction matrix

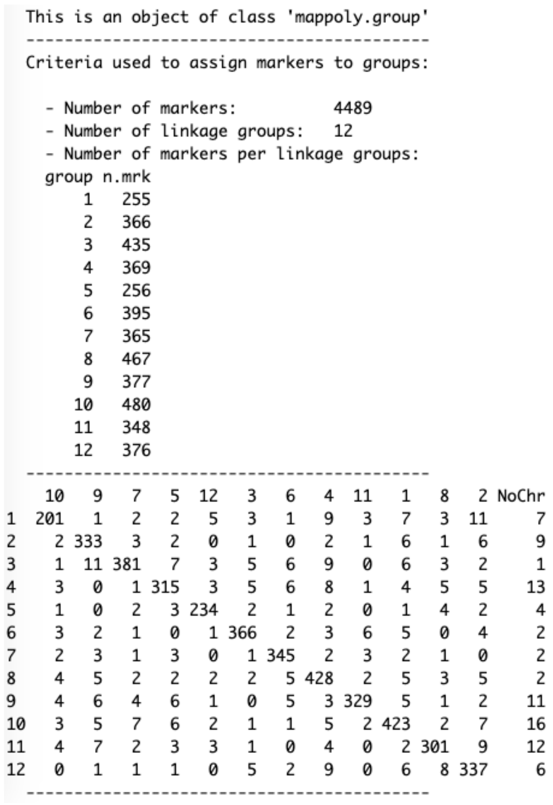
```
m <- rf_list_to_matrix(tpt, ncpus = nc)
gen.ord <- get_genomic_order(s)
s.gen.ord <- make_seq_mappoly(gen.ord)
plot(m, ord = s.gen.ord$seq.mrk.names, fact = 10)
```



Filtering by pairwise recombination fraction

```
gr <- group_mappoly(m, expected.groups = 12,
                    comp.mat = TRUE)
gr
```





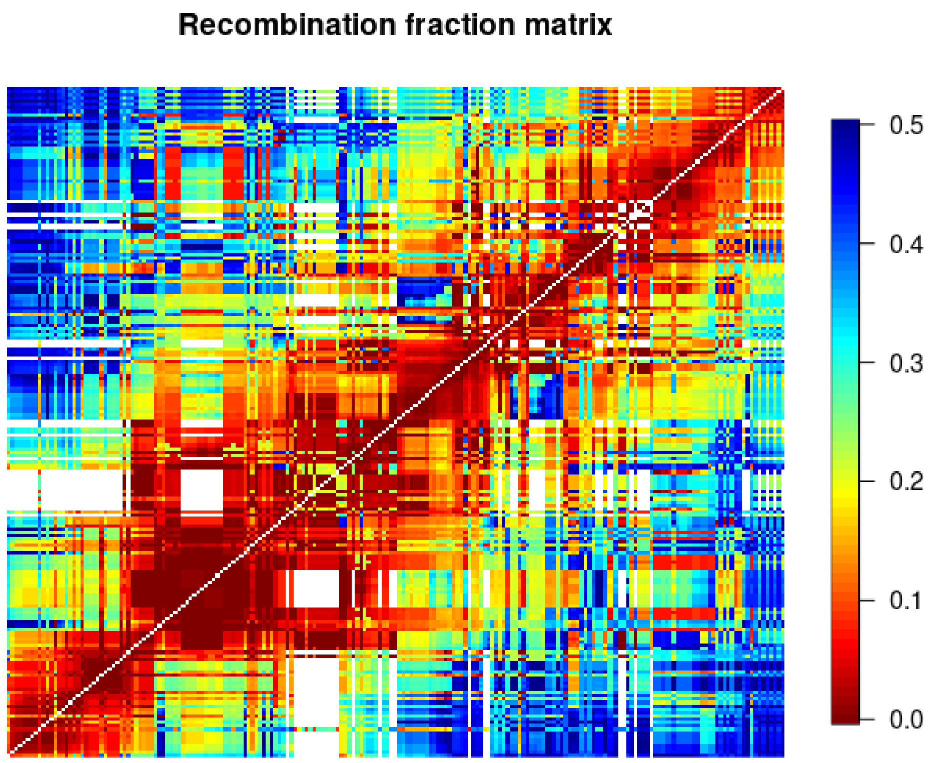
```
This is an object of class 'mappoly.group'
  ------------------------------------------
  Criteria used to assign markers to groups:

    - Number of markers:          4489
    - Number of linkage groups:   12
    - Number of markers per linkage groups:
    group n.mrk
        1   255
        2   366
        3   435
        4   369
        5   256
        6   395
        7   365
        8   467
        9   377
       10   480
       11   348
       12   376
  ------------------------------------------
    10   9   7   5  12   3   6   4  11   1   8   2 NoChr
1  201   1   2   2   5   3   1   9   3   7   3  11     7
2    2 333   3   2   0   1   0   2   1   6   1   6     9
3    1  11 381   7   3   5   6   9   0   6   3   2     1
4    3   0   1 315   3   5   6   8   1   4   5   5    13
5    1   0   2   3 234   2   1   2   0   1   4   2     4
6    3   2   1   0   1 366   2   3   6   5   0   4     2
7    2   3   1   3   0   1 345   2   3   2   1   0     2
8    4   5   2   2   2   2   5 428   2   5   3   5     2
9    4   6   4   6   1   0   5   3 329   5   1   2    11
10   3   5   7   6   2   1   1   5   2 423   2   7    16
11   4   7   2   3   3   1   0   4   0   2 301   9    12
12   0   1   1   1   0   5   2   9   0   6   8 337     6
  ------------------------------------------
```

MAPpoly

Ordering within a specific linkage group

Genome order

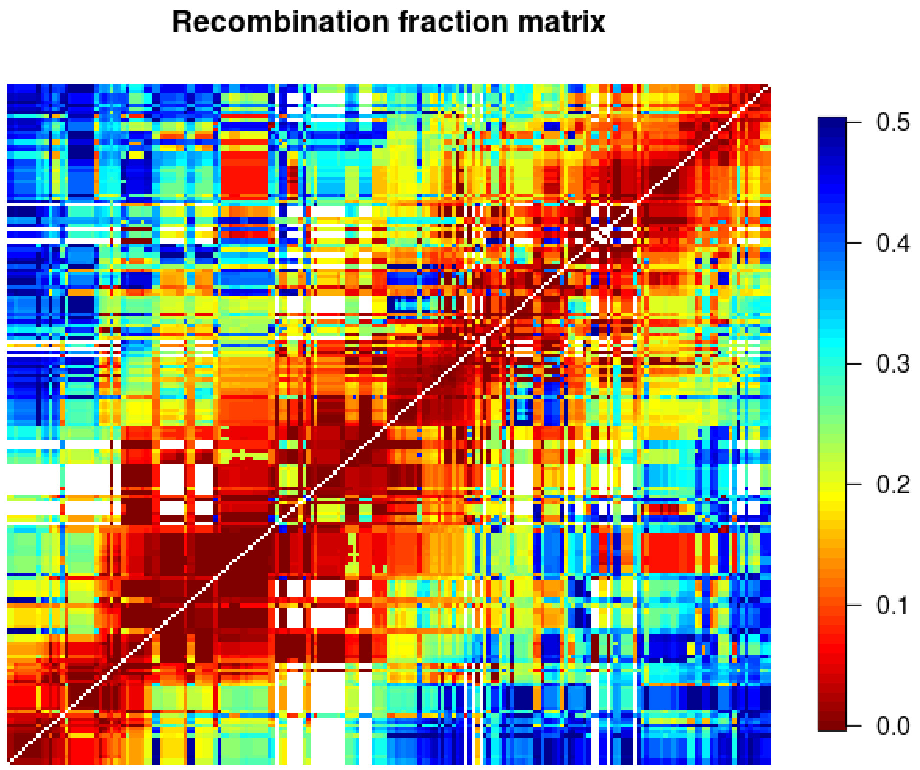
```
s1 <- make_seq_mappoly(gr, arg = 1,
                       genomic.info = 1)
tpt1 <- make_pairs_mappoly(tpt, s1)
m1 <- make_mat_mappoly(m, s1)
##Genomic order
g.o1 <- get_genomic_order(s1)
s1.g <- make_seq_mappoly(g.o1)
plot(g.o1)
plot(m1, ord = s1.g$seq.mrk.names,
     fact = 3)
```



Ordering within a specific linkage group

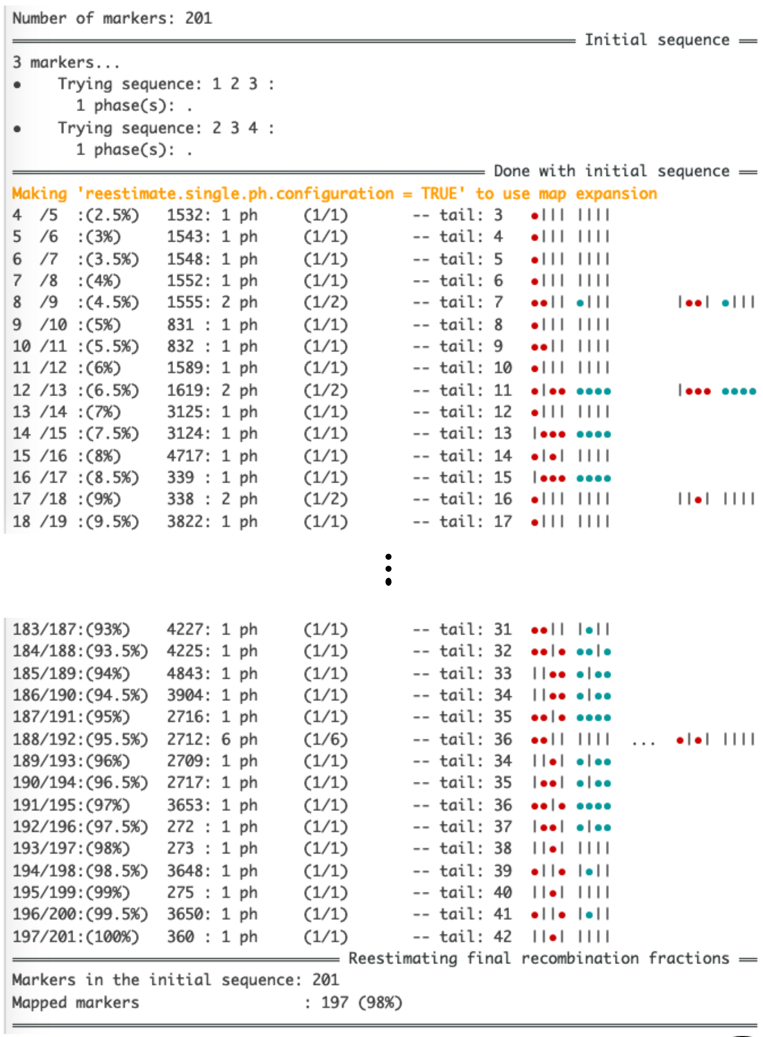
MDS order

```r
mds.o1 <- mds_mappoly(m1, n = c(201,47))
plot(mds.o1)
s1.mds <- make_seq_mappoly(mds.o1)
plot(m1, ord = s1.mds$seq.mrk.names)
```



Phasing and HMM estimation of distance

```r
lg1.geno.map<-est_rf_hmm_sequential(input.seq = s1.g,
                                    start.set = 3,
                                    thres.twopt = 10,
                                    thres.hmm = 50,
                                    extend.tail = 30,
                                    twopt = tpt1,
                                    verbose = TRUE,
                                    tol = 10e-2,
                                    tol.final = 10e-4,
                                    phase.number.limit = 20,
                                    sub.map.size.diff.limit = 5,
                                    info.tail = TRUE)
```



```
Number of markers: 201
═══════════════════════════════════════════════════════ Initial sequence ══
3 markers...
•    Trying sequence: 1 2 3 :
       1 phase(s): .
•    Trying sequence: 2 3 4 :
       1 phase(s): .
════════════════════════════════════════════════ Done with initial sequence ══
Making 'reestimate.single.ph.configuration = TRUE' to use map expansion
4  /5  :(2.5%)   1532: 1 ph     (1/1)       -- tail: 3  ●||| ||||
5  /6  :(3%)     1543: 1 ph     (1/1)       -- tail: 4  ●||| ||||
6  /7  :(3.5%)   1548: 1 ph     (1/1)       -- tail: 5  ●||| ||||
7  /8  :(4%)     1552: 1 ph     (1/1)       -- tail: 6  ●||| ||||
8  /9  :(4.5%)   1555: 2 ph     (1/2)       -- tail: 7  ●●|| ●|||       |●●| ●|||
9  /10 :(5%)     831 : 1 ph     (1/1)       -- tail: 8  ●||| ||||
10 /11 :(5.5%)   832 : 1 ph     (1/1)       -- tail: 9  ●●|| ||||
11 /12 :(6%)     1589: 1 ph     (1/1)       -- tail: 10 ●||| ||||
12 /13 :(6.5%)   1619: 2 ph     (1/2)       -- tail: 11 ●|●● ●●●●       |●●● ●●●●
13 /14 :(7%)     3125: 1 ph     (1/1)       -- tail: 12 ●||| ||||
14 /15 :(7.5%)   3124: 1 ph     (1/1)       -- tail: 13 |●●● ●●●●
15 /16 :(8%)     4717: 1 ph     (1/1)       -- tail: 14 ●|●| ||||
16 /17 :(8.5%)   339 : 1 ph     (1/1)       -- tail: 15 |●●● ●●●●
17 /18 :(9%)     338 : 2 ph     (1/2)       -- tail: 16 ●||| ||||       ||●| ||||
18 /19 :(9.5%)   3822: 1 ph     (1/1)       -- tail: 17 ●||| ||||
```

⋮

```
183/187:(93%)    4227: 1 ph     (1/1)       -- tail: 31 ●●|| |●||
184/188:(93.5%)  4225: 1 ph     (1/1)       -- tail: 32 ●●|● ●●|●
185/189:(94%)    4843: 1 ph     (1/1)       -- tail: 33 ||●● ●|●●
186/190:(94.5%)  3904: 1 ph     (1/1)       -- tail: 34 ||●● ●|●●
187/191:(95%)    2716: 1 ph     (1/1)       -- tail: 35 ●●|● ●●●●
188/192:(95.5%)  2712: 6 ph     (1/6)       -- tail: 36 ●●|| ||||  ...  ●|●| ||||
189/193:(96%)    2709: 1 ph     (1/1)       -- tail: 34 ||●| ●|●●
190/194:(96.5%)  2717: 1 ph     (1/1)       -- tail: 35 |●●| ●|●●
191/195:(97%)    3653: 1 ph     (1/1)       -- tail: 36 ●●|● ●●●●
192/196:(97.5%)  272 : 1 ph     (1/1)       -- tail: 37 |●●| ●|●●
193/197:(98%)    273 : 1 ph     (1/1)       -- tail: 38 ||●| ||||
194/198:(98.5%)  3648: 1 ph     (1/1)       -- tail: 39 ●||● |●||
195/199:(99%)    275 : 1 ph     (1/1)       -- tail: 40 ||●| ||||
196/200:(99.5%)  3650: 1 ph     (1/1)       -- tail: 41 ●||● |●||
197/201:(100%)   360 : 1 ph     (1/1)       -- tail: 42 ||●| ||||
═════════════════════════════════════ Reestimating final recombination fractions ══
Markers in the initial sequence: 201
Mapped markers                  : 197 (98%)
══════════════════════════════════════════════════════════════════════════════
```

Phasing and HMM estimation of distance

Tetraploid

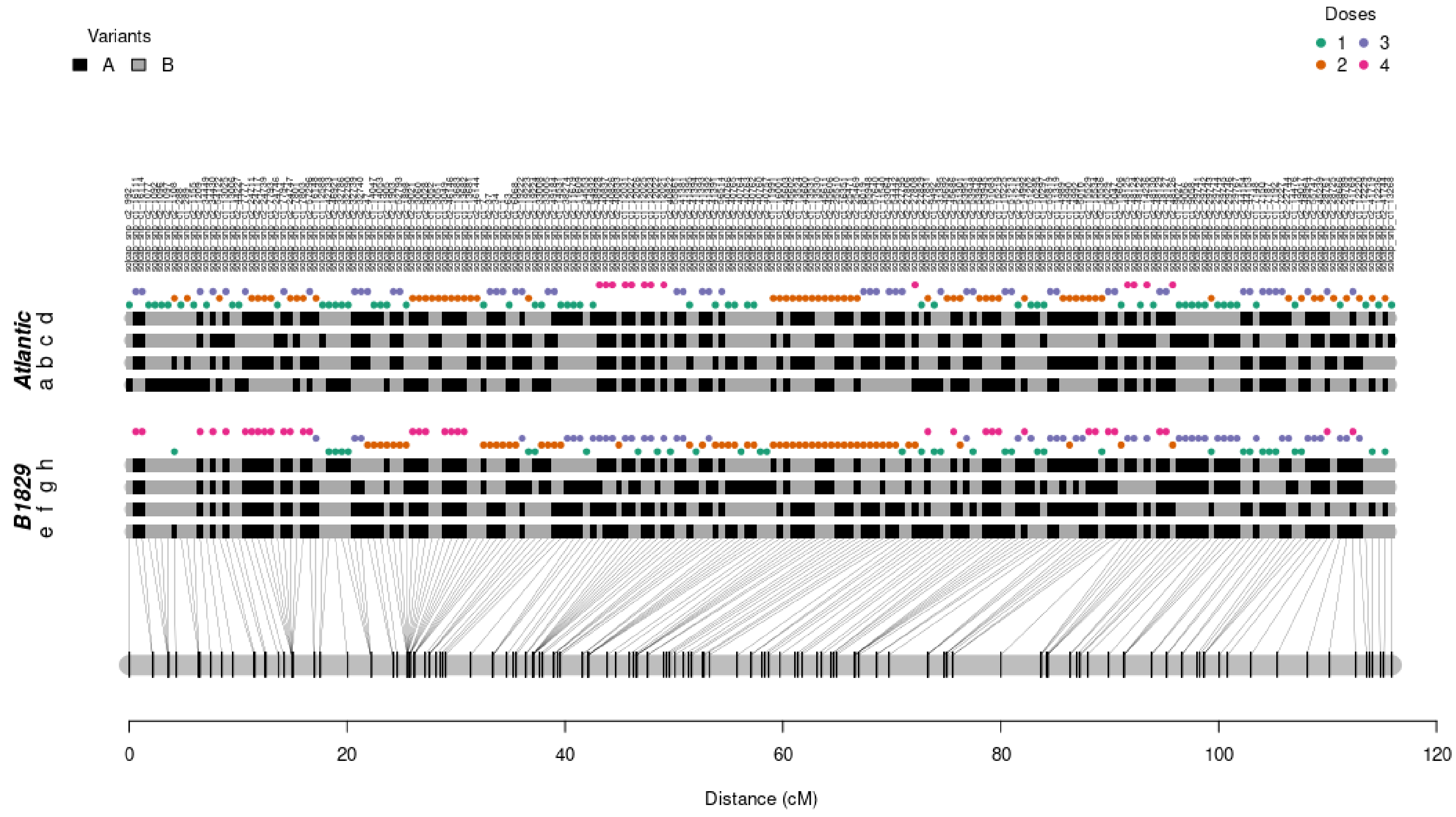
Argument	Value
`start.set`	3 – 10
`thres.twopt`	10
`thres.hmm`	10 – 50
`extend.tail`	30 – 100
`phase.number.limit`	20
`sub.map.size.diff.limit`	2 (thousands per LG) … 10 (hundred(s) per LG)

Hexaploid

Argument	Value
`start.set`	3 – 5
`thres.twopt`	10
`thres.hmm`	10 – 50
`extend.tail`	50 – 200
`phase.number.limit`	20
`sub.map.size.diff.limit`	2 (thousands per LG) … 10 (hundred(s) per LG)

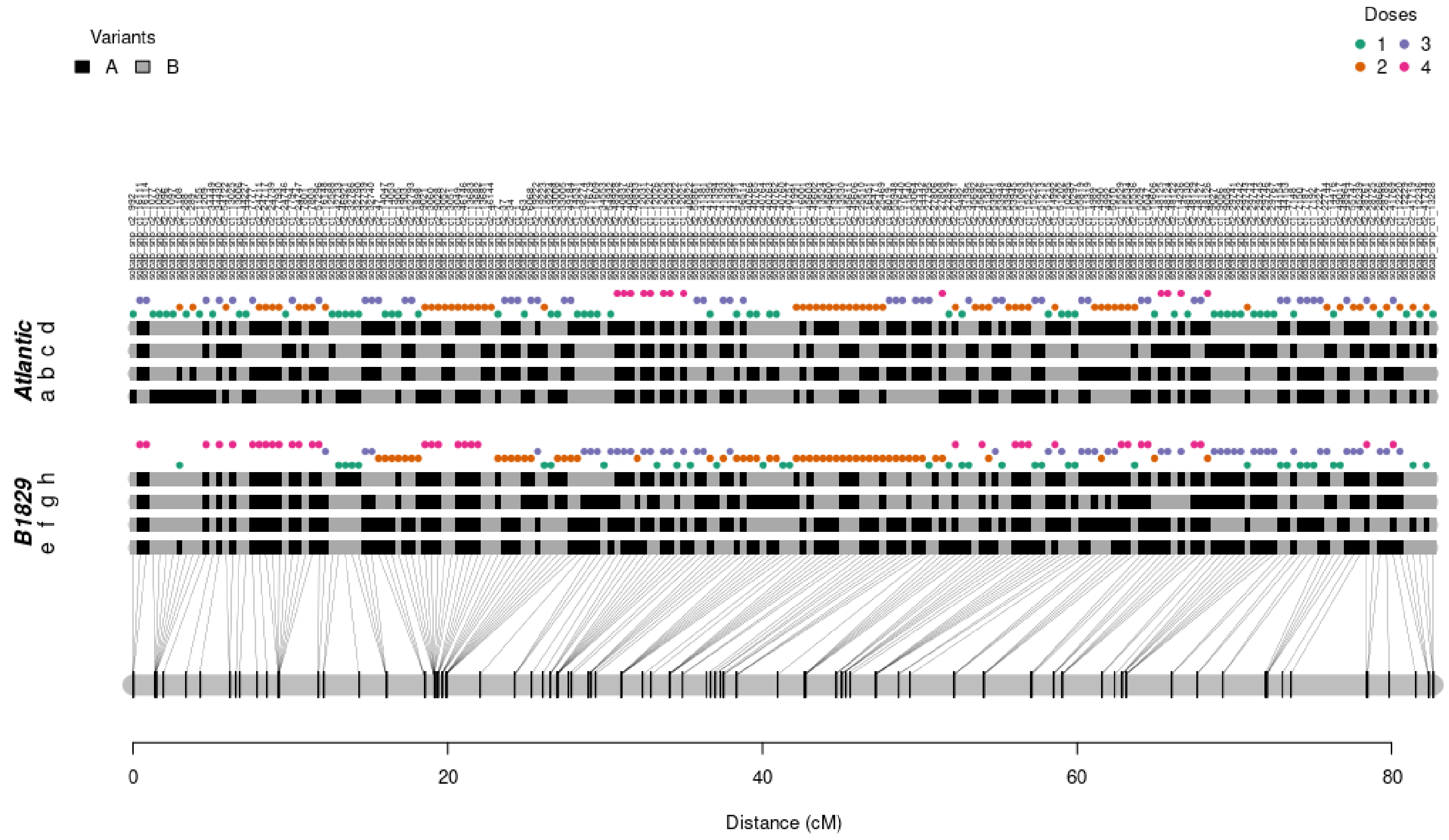
Resulting map

```
plot(lg1.geno.map, mrk.names = TRUE, cex = 0.5, P = "Atlantic", Q = "B1829")
```



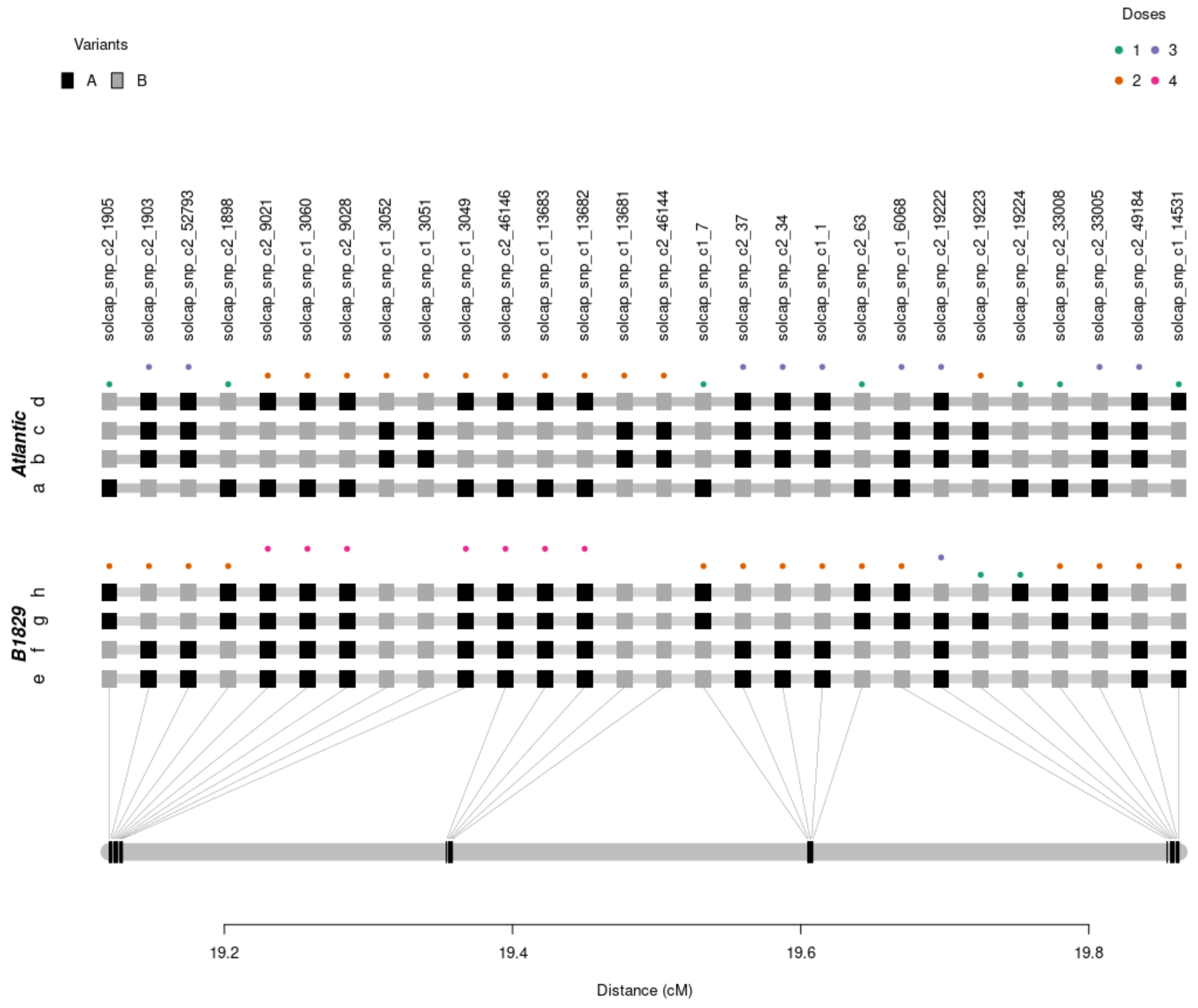
Resulting map – error modeling

```
lg1.geno.map.err <- est_full_hmm_with_global_error(lg1.geno.map, error = 0.05, tol = 10e-4)
plot(lg1.geno.map.err, mrk.names = TRUE, cex = 0.5, P = "Atlantic", Q = "B1829")
```



Resulting map: 20-30 cM segment

```
plot(lg1.geno.map.err,
     mrk.names = TRUE,
     left.lim = 19,
     right.lim = 20,
     P = "Atlantic",
     Q = "B1829")
```

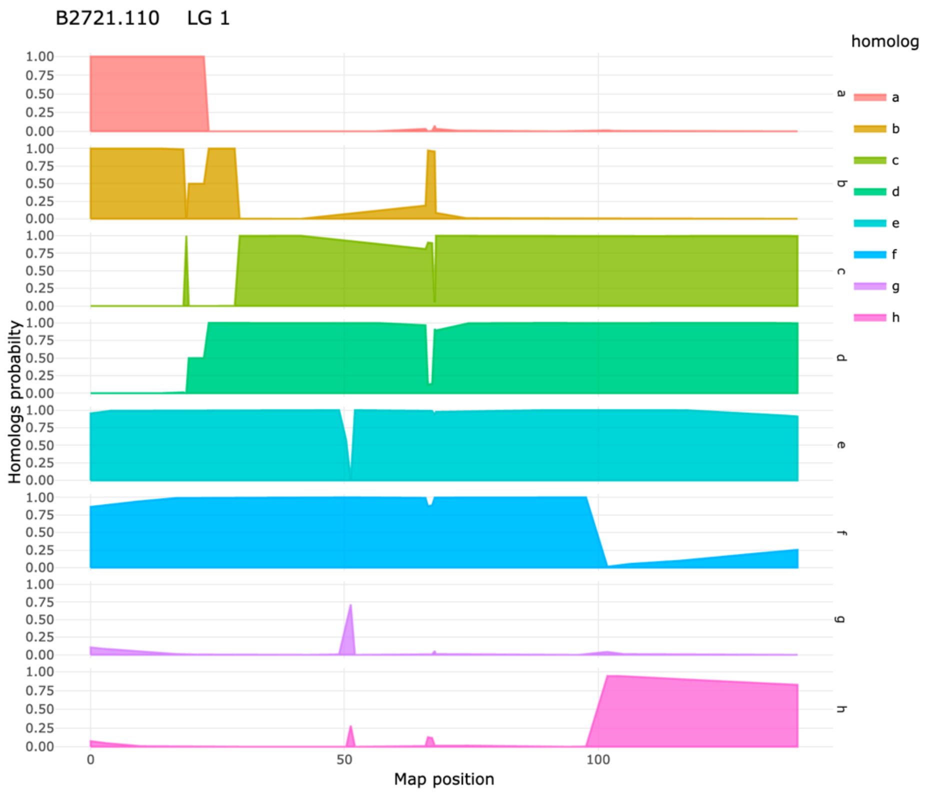


NC STATE UNIVERSITY

MAPpoly

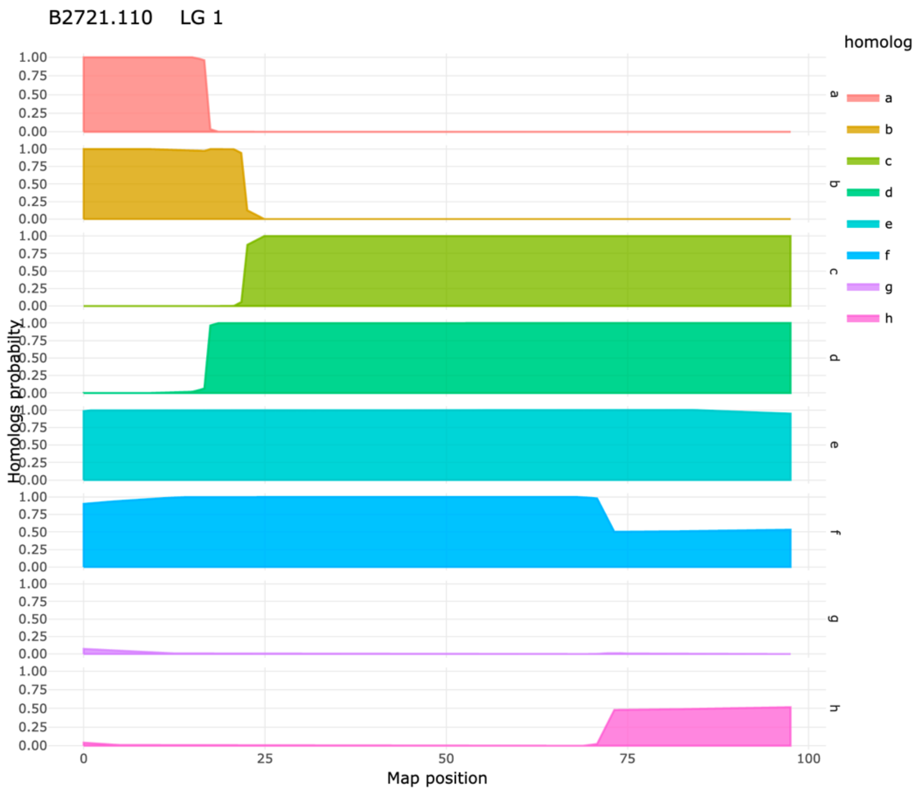
Probabilistic haplotype reconstruction

```
g.lg10 <- calc_genoprob(lg10.geno.map, step = 1)
h.lg10 <- calc_homoprob(g.lg10)
plot(h.lg10, ind = "B2721.110")
```



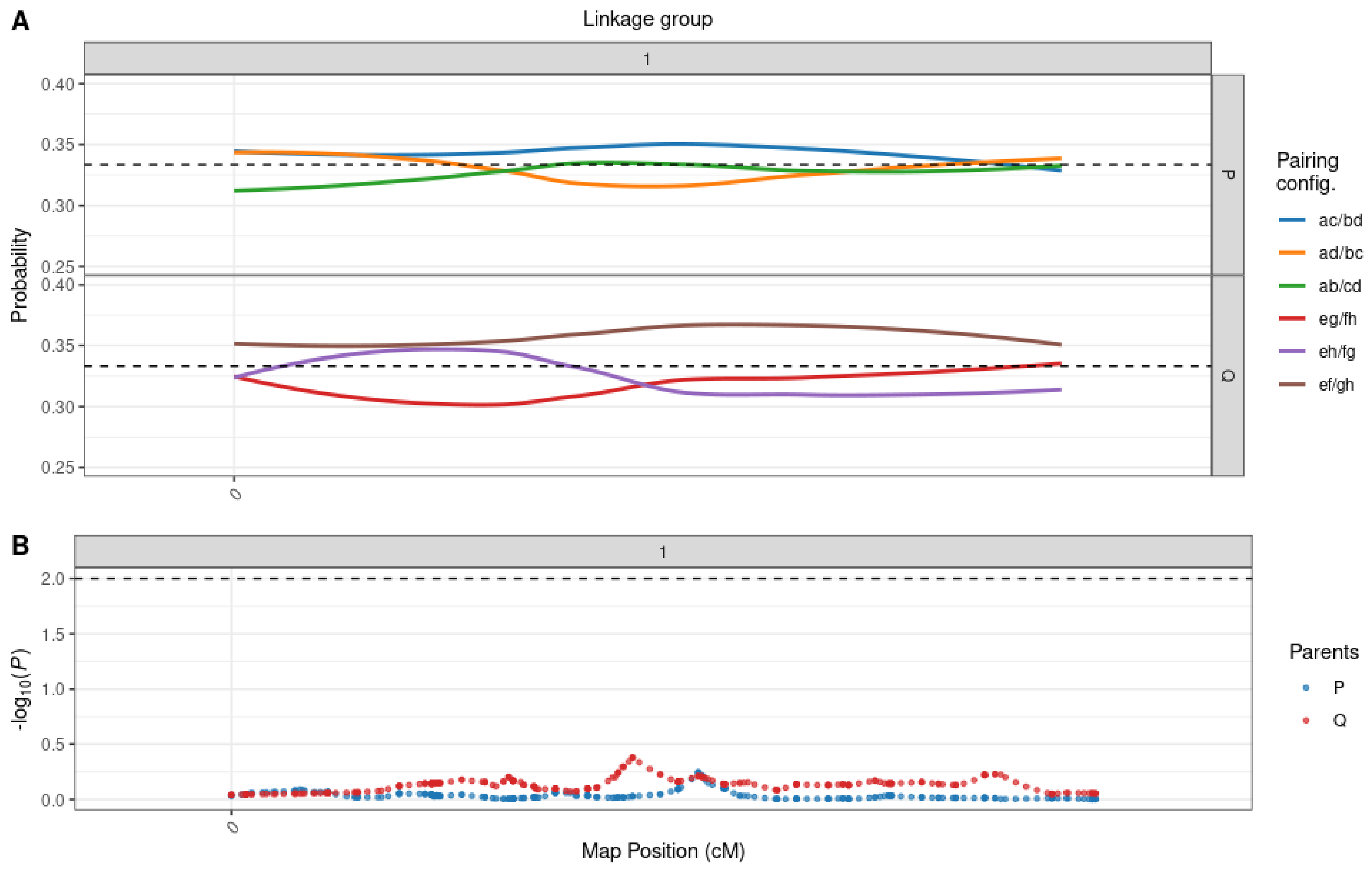
Probabilistic haplotype reconstruction

```
g.lg10.err <- calc_genoprob_error(lg10.geno.map.err, step = 1, error = 0.05)
h.lg10.err <- calc_homoprob(g.lg10.err)
plot(h.lg10.err, ind = "B2721.110")
```



Preferential pairing profiles

```
p1 <- calc_prefpair_profiles(g1)
plot(p1, min.y.prof = 0.25, max.y.prof = 0.40)
```



MAPpoly

Parallel map construction – genomic order

```r
#### Functions ####
phasing_and_hmm_rf <- function(X){
  dir.create("map_output", showWarnings = FALSE)
  fl <- paste0("output_map_ch_", X$seq$sequence[1], ".txt")
  fl <- file.path("map_output", fl)
  sink(fl)
  map <- est_rf_hmm_sequential(input.seq = X$seq,
                               start.set = 3,
                               thres.twopt = 10,
                               thres.hmm = 50,
                               extend.tail = 30,
                               twopt = X$tpt,
                               verbose = TRUE,
                               phase.number.limit = 20,
                               sub.map.size.diff.limit = 5)
  sink()
  return(map)
}
error_model <- function(X, error = 0.05, tol = 10e-4){
x <- est_full_hmm_with_global_error(input.map = X,
                                    error = error,
                                    tol = tol,
                                    verbose = FALSE)
  return(x)
}
```

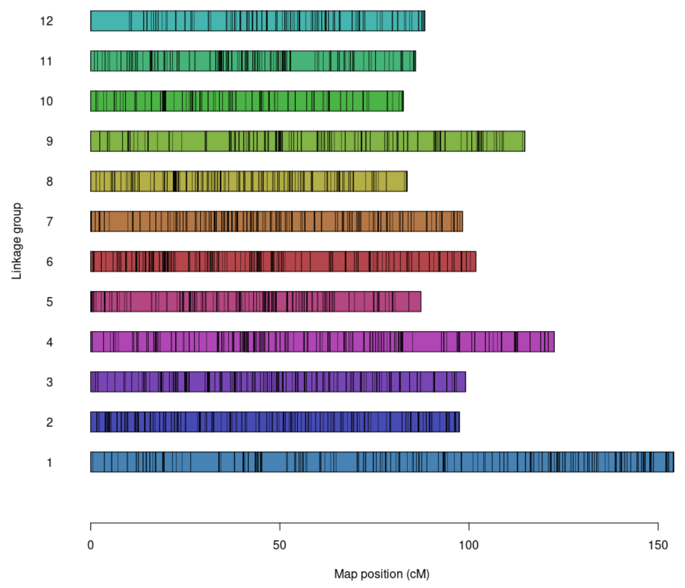
```r
#### Correspondence with genome
z<-as.numeric(colnames(gr$seq.vs.grouped.snp)[1:12])

#### Assembling linkage groups (order based on genome)
LGS<-vector("list", 12)
for(ch in 1:12){
  cat("\n ~~~~~~ ch:", ch, "...\n")
  lg <- which(z==ch)
  s.temp<-make_seq_mappoly(gr, lg, genomic.info = 1)
  tpt.temp <- make_pairs_mappoly(tpt, s.temp)
  s.temp.filt <- rf_snp_filter(tpt.temp, 5, 5, 0.15, c(0.05, 1))
  m.temp <- make_mat_mappoly(m, s.temp.filt)
  g.o <- get_genomic_order(s.temp)
  s.g <- make_seq_mappoly(g.o)
  tpt.temp <- make_pairs_mappoly(tpt, input.seq = s.g)
  LGS[[ch]] <- list(seq = s.g, tpt = tpt.temp)
}
```

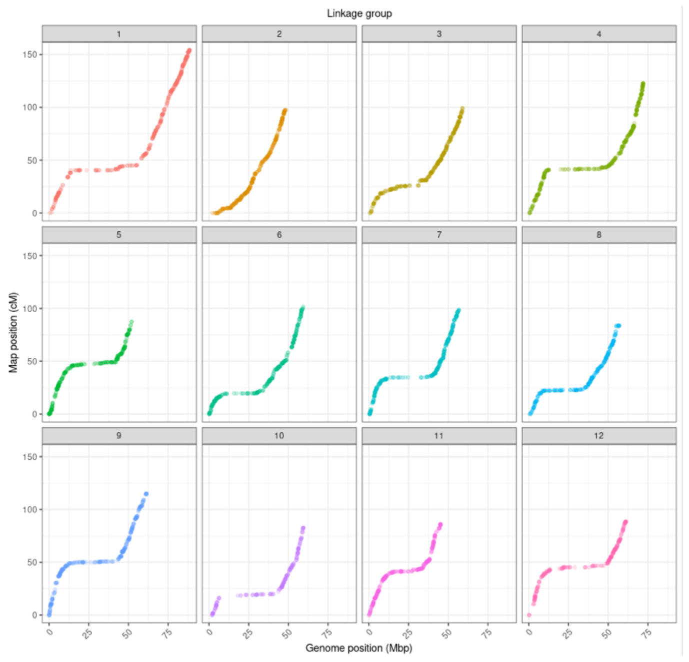
```r
#### Parallel map construction
cl <- parallel::makeCluster(12)
parallel::clusterEvalQ(cl, require(mappoly))
parallel::clusterExport(cl, "dat")
# ~12.5 minutes
MAPs.geno <- parallel::parLapply(cl, LGS, phasing_and_hmm_rf)
# ~2.5 minutes
MAPs.geno.err <- parallel::parLapply(cl, MAPs.geno,
                                     error_model)
# ~22 seconds
genoprob <- parallel::parLapply(cl,
                                MAPs.geno.err,
                                calc_genoprob_error,
                                step = 1,
                                error = 0.05)
parallel::stopCluster(cl)
```

```r
#### Map results
map.out <- plot_map_list(MAPs.geno.err, col = "ggstyle")
map.out
plot_genome_vs_map(MAPs.geno.err, same.ch.lg = TRUE)
summary_maps(MAPs.geno.err)
export_map_list(MAPs.geno.err, file = "output_map.csv")
```

```r
#### Preferential pairing
pp <- calc_prefpair_profiles(genoprob)
print(pp)
head(pp$prefpair.psi)
plot(pp, P = "Atlantic", Q = "B1829")

#### Haplotype probabilities
hp <- calc_homoprob(genoprob)
print(hp)
plot(hp, ind = 2, lg = 1)
plot(hp, ind = 2, lg = 1:12, use.plotly = FALSE)
```

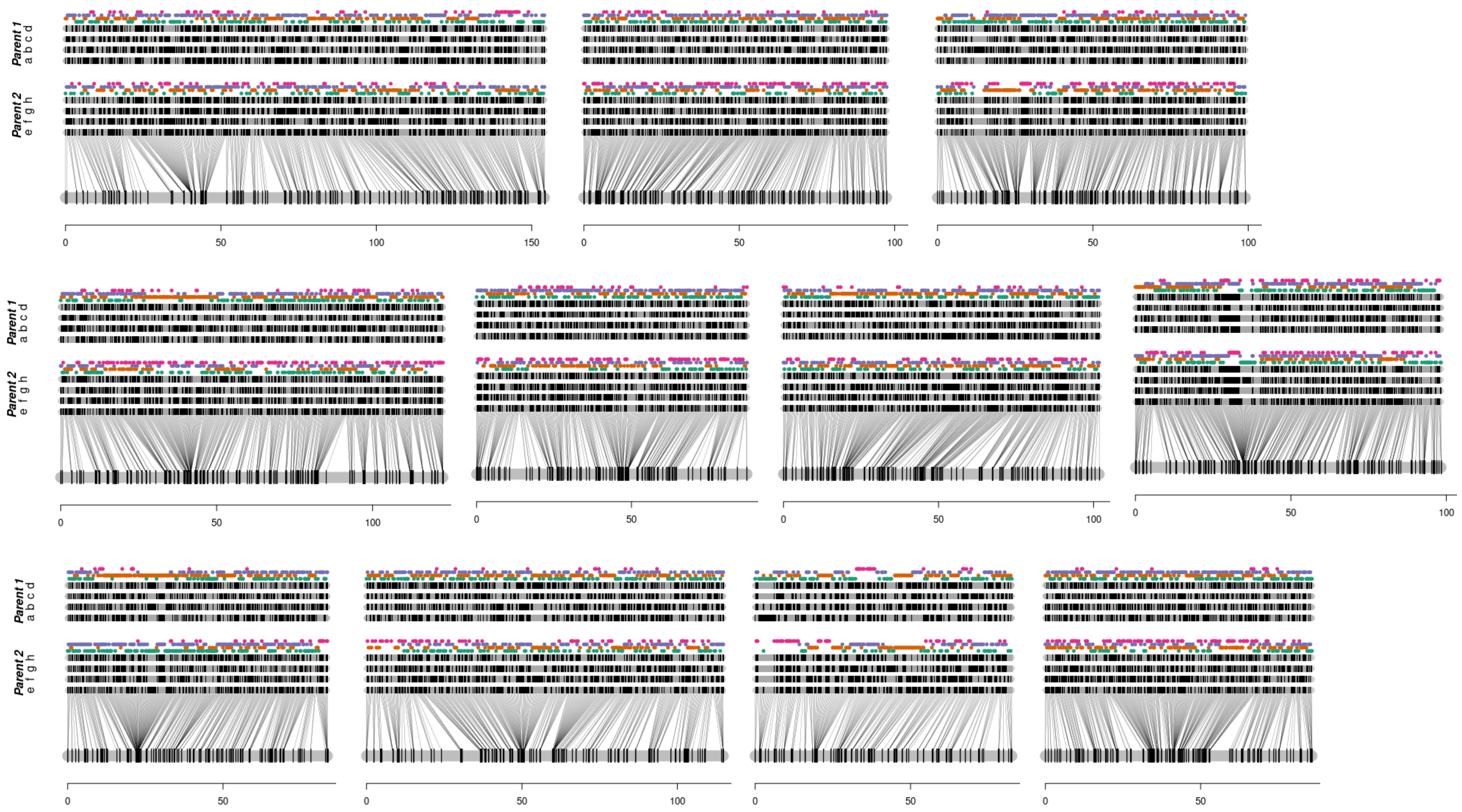
Results





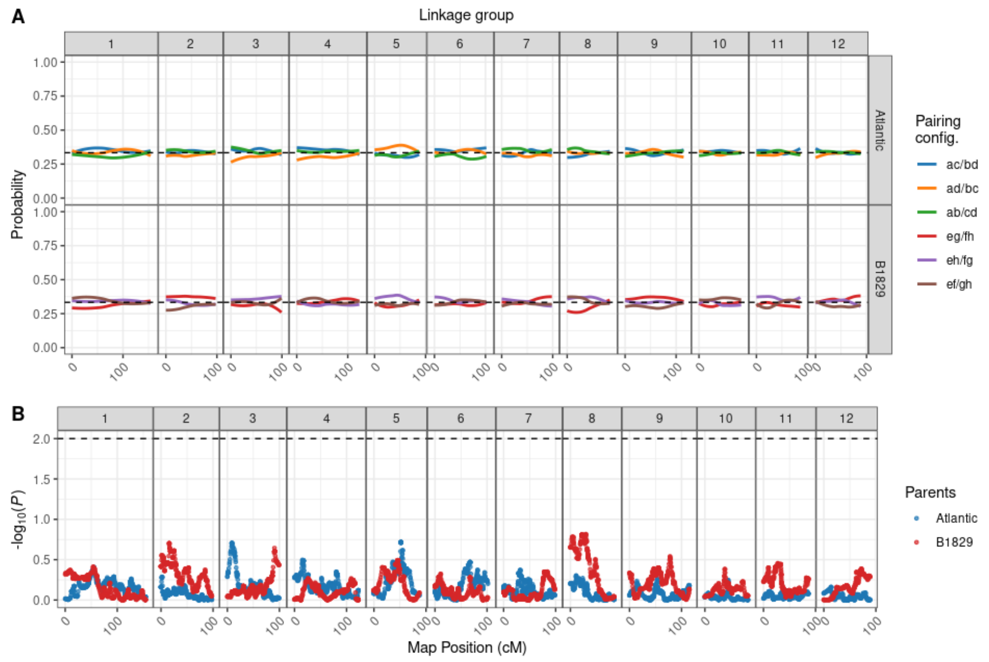
```
   LG Genomic sequence Map length (cM) Markers/cM Simplex Double-simplex Multiplex Total Max gap
1   1             1-NA           155.5       2.89     103            107       202   449   12.84
2   2                2           97.25       4.77     142            132       155   464    2.99
3   3                3          113.91       3.92     160             26       212   446    8.38
4   4                4           121.8       3.92     123             81       235   477    7.71
5   5             5-NA           90.15       3.87     127             56       140   349    5.69
6   6                6          106.18       4.15      72             83       247   441    3.55
7   7           7-NA-4           97.43       4.95     141             74       181   482    4.96
8   8           8-NA-1           93.83       3.88      59            102       165   364    3.32
9   9           9-7-NA          117.73       3.16      81             95       161   372    6.51
10 10               10           97.52       2.47      56             34       131   241    3.68
11 11               11            85.8       4.16      97             52       192   357    5.97
12 12               12           90.81       3.28     121             27       114   298    3.62
13 Total          <NA>         1267.91       3.79    1282            869      2135  4740    5.77
```

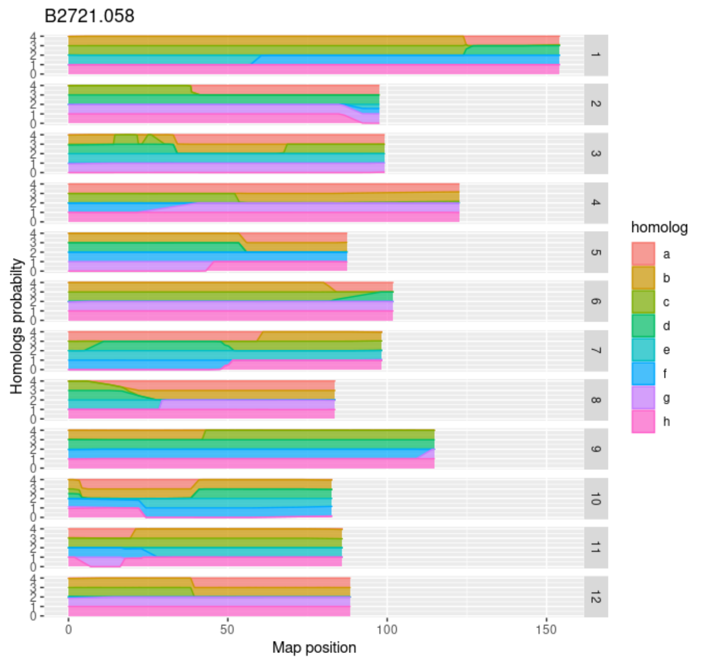
Results – phased map

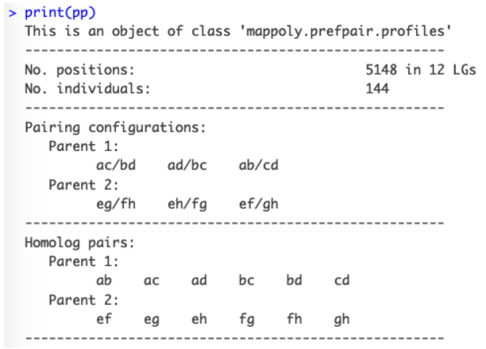


NC STATE UNIVERSITY

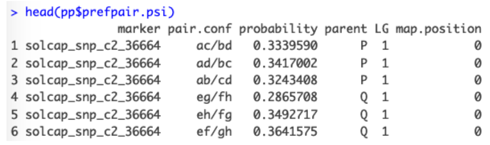
Results



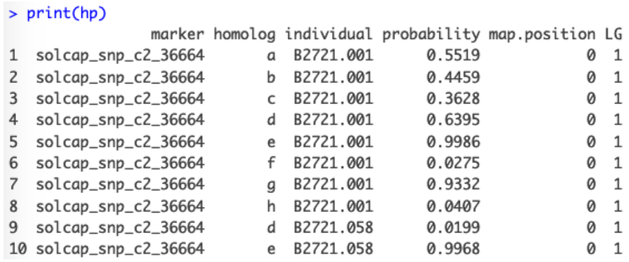




```
> print(pp)
  This is an object of class 'mappoly.prefpair.profiles'
  ----------------------------------------------------
  No. positions:                              5148 in 12 LGs
  No. individuals:                            144
  ----------------------------------------------------
  Pairing configurations:
     Parent 1:
          ac/bd     ad/bc     ab/cd
     Parent 2:
          eg/fh     eh/fg     ef/gh
  ----------------------------------------------------
  Homolog pairs:
     Parent 1:
          ab    ac    ad    bc    bd    cd
     Parent 2:
          ef    eg    eh    fg    fh    gh
  ----------------------------------------------------
```



```
> head(pp$prefpair.psi)
              marker pair.conf probability parent LG map.position
1 solcap_snp_c2_36664     ac/bd   0.3339590      P  1            0
2 solcap_snp_c2_36664     ad/bc   0.3417002      P  1            0
3 solcap_snp_c2_36664     ab/cd   0.3243408      P  1            0
4 solcap_snp_c2_36664     eg/fh   0.2865708      Q  1            0
5 solcap_snp_c2_36664     eh/fg   0.3492717      Q  1            0
6 solcap_snp_c2_36664     ef/gh   0.3641575      Q  1            0
```



```
> print(hp)
                marker homolog individual probability map.position LG
1  solcap_snp_c2_36664       a  B2721.001      0.5519            0  1
2  solcap_snp_c2_36664       b  B2721.001      0.4459            0  1
3  solcap_snp_c2_36664       c  B2721.001      0.3628            0  1
4  solcap_snp_c2_36664       d  B2721.001      0.6395            0  1
5  solcap_snp_c2_36664       e  B2721.001      0.9986            0  1
6  solcap_snp_c2_36664       f  B2721.001      0.0275            0  1
7  solcap_snp_c2_36664       g  B2721.001      0.9332            0  1
8  solcap_snp_c2_36664       h  B2721.001      0.0407            0  1
9  solcap_snp_c2_36664       d  B2721.058      0.0199            0  1
10 solcap_snp_c2_36664       e  B2721.058      0.9968            0  1
```

On-line resources

- MAPpoly web page
- GitHub page for the topics I presented in this workshop
- Tetraploid: B2721 potato biparental population (Atlantic x B1829-5)
- Hexaploid: BT sweetpotato population (Beauregard x Tanzania)
- Detailed analytic procedures for Pereira et al. (2020)

MAPpoly

APPENDIX

Hexaploid map – Results

Hexaploid map – Results

- Script for hexaploid
  - https://github.com/mmollina/SCRI/tree/main/MAPpoly/hexa
- Complete map
  - Unraveling the Hexaploid Sweetpotato Inheritance Using Ultra-Dense Multilocus Mapping (Mollinari et al., 2020).
- Shiny applications
  - Map: https://gt4sp-genetic-map.shinyapps.io/bt_map/
  - Haplotypes: https://gt4sp-genetic-map.shinyapps.io/offspring_haplotype_BT_population/

Biparental Population - BT

- Beauregard x Tanzania
- 315 individuals
- GBS – GBSpoly protocol (Bode Olukolu – U Tennessee)
- Two reference genomes *I. trifida* and *I. triloba* (Zhangjun Fei's group – BTI Cornell). In the available dataset we used *I. trifida.*
- In this example we used three chromosomes: ch3, ch9 and ch12

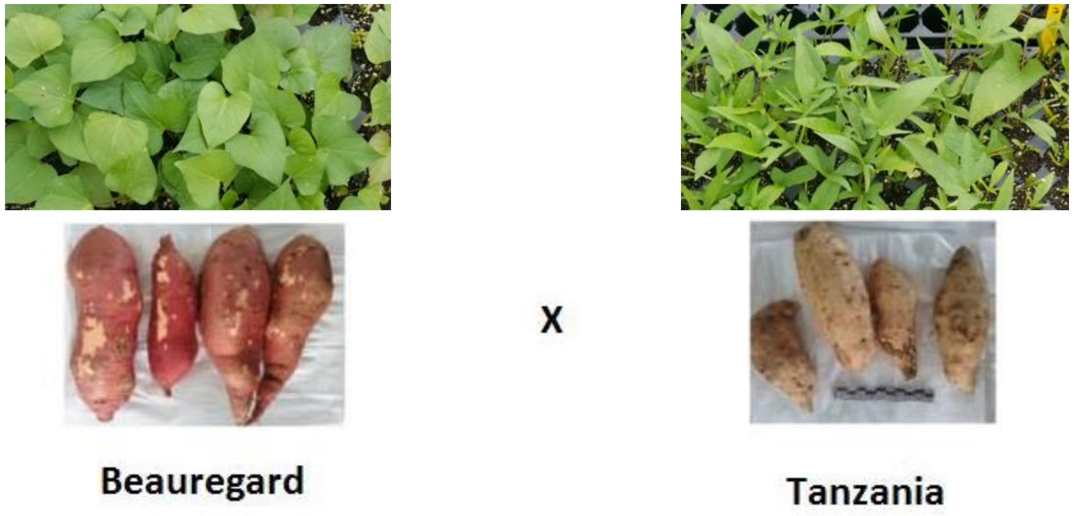
x

Beauregard

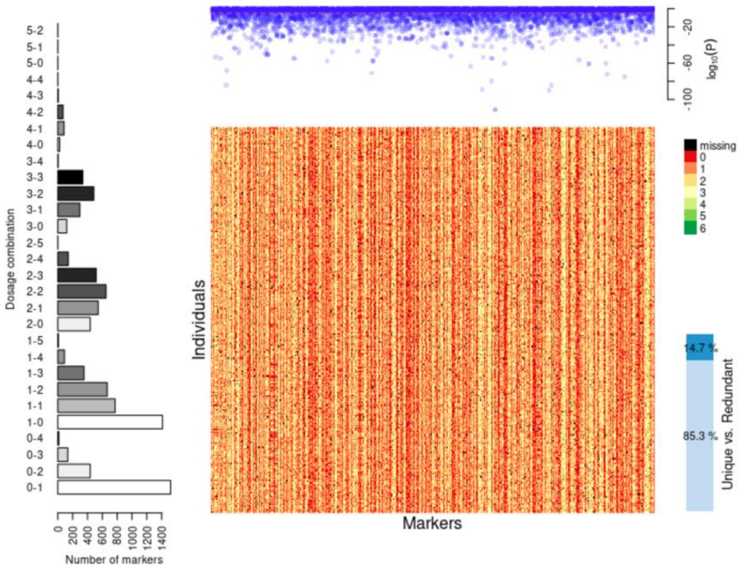
Tanzania

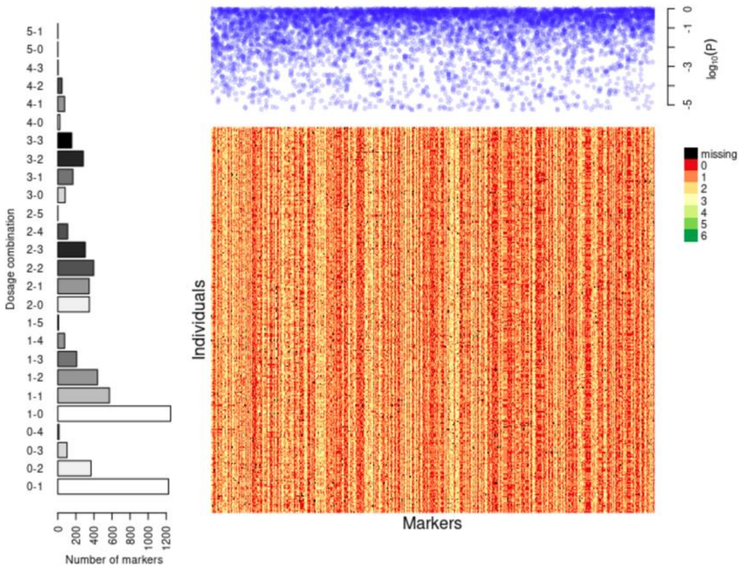
Phenotyping Beauregard x
Tanzania Mapping Population
Design: Alpha lattice (80 x 4)
Genotypes: 320
Replications: 3
Plot: 8 plants/row, 2 rows

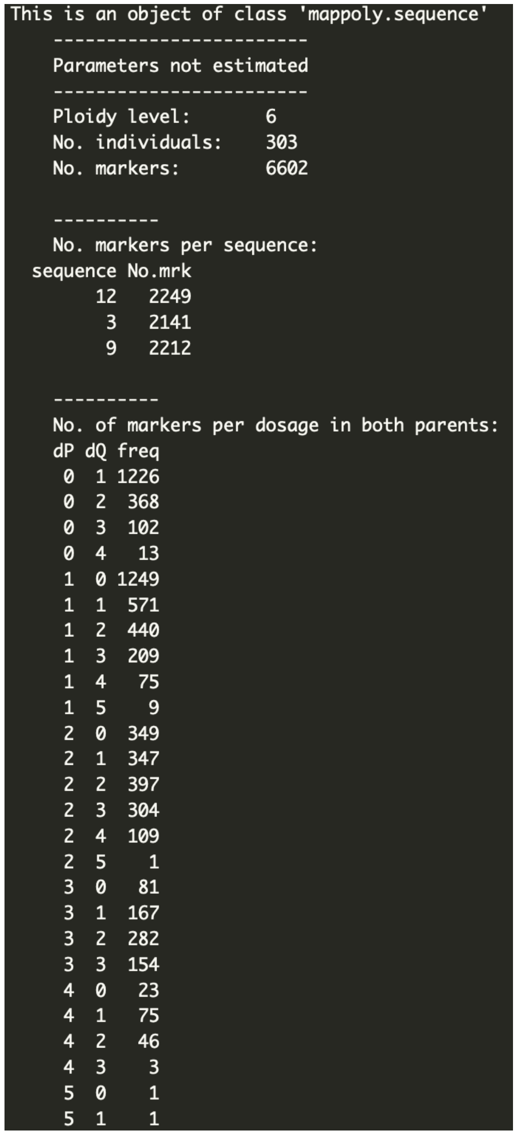
Biparental Population - BT



Results for hexaploid sweetpotato BT population

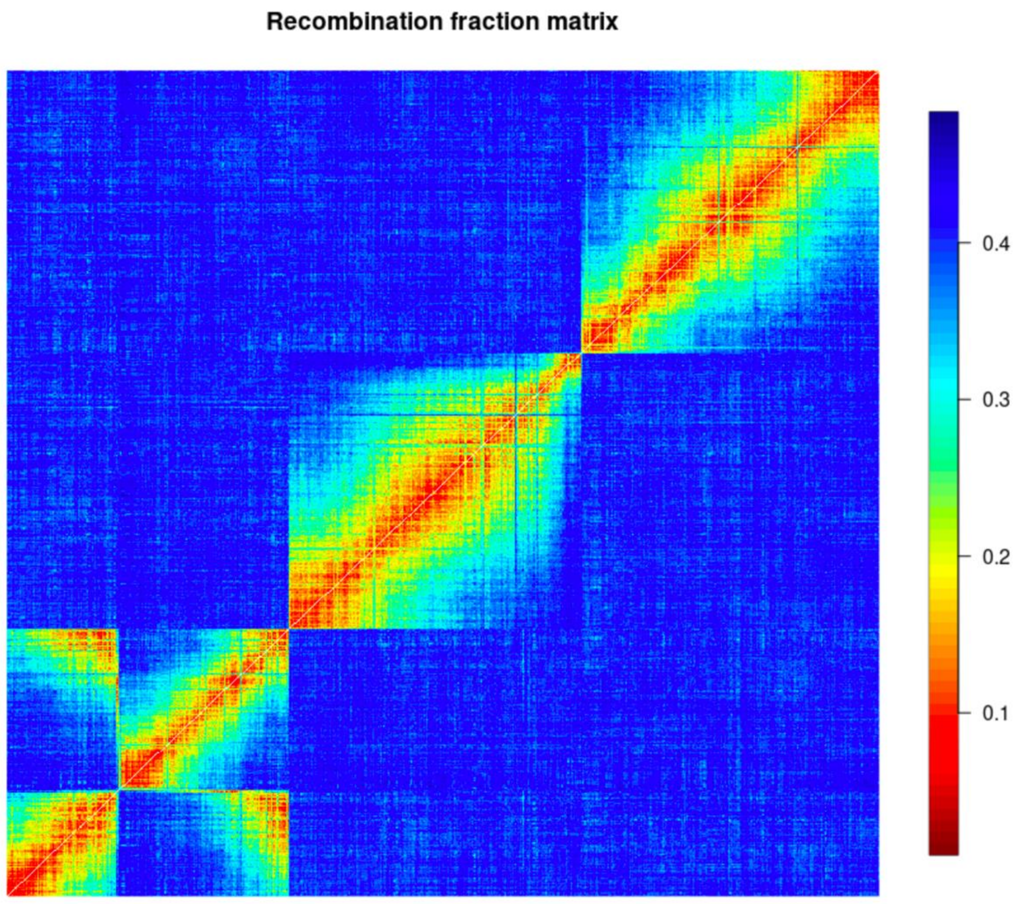


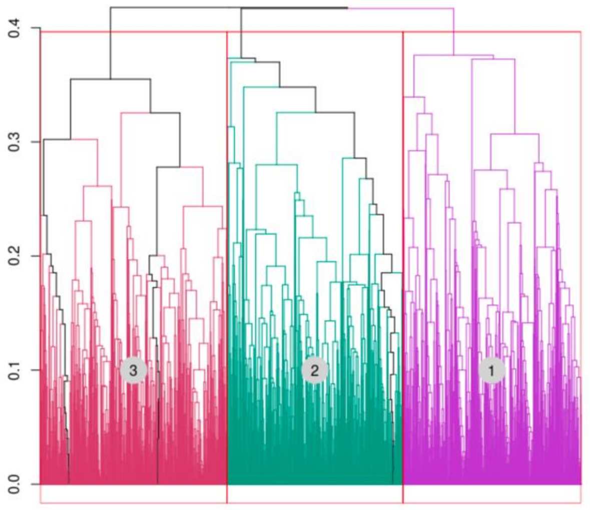


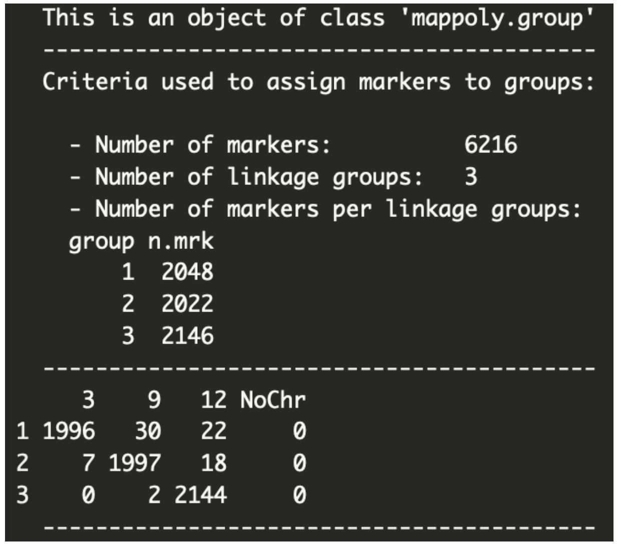


```
This is an object of class 'mappoly.sequence'
    ------------------------
    Parameters not estimated
    ------------------------
    Ploidy level:       6
    No. individuals:    303
    No. markers:        6602

    ----------
    No. markers per sequence:
  sequence No.mrk
        12   2249
         3   2141
         9   2212

    ----------
    No. of markers per dosage in both parents:
    dP dQ freq
     0  1 1226
     0  2  368
     0  3  102
     0  4   13
     1  0 1249
     1  1  571
     1  2  440
     1  3  209
     1  4   75
     1  5    9
     2  0  349
     2  1  347
     2  2  397
     2  3  304
     2  4  109
     2  5    1
     3  0   81
     3  1  167
     3  2  282
     3  3  154
     4  0   23
     4  1   75
     4  2   46
     4  3    3
     5  0    1
     5  1    1
```

Recombination fraction matrix and grouping

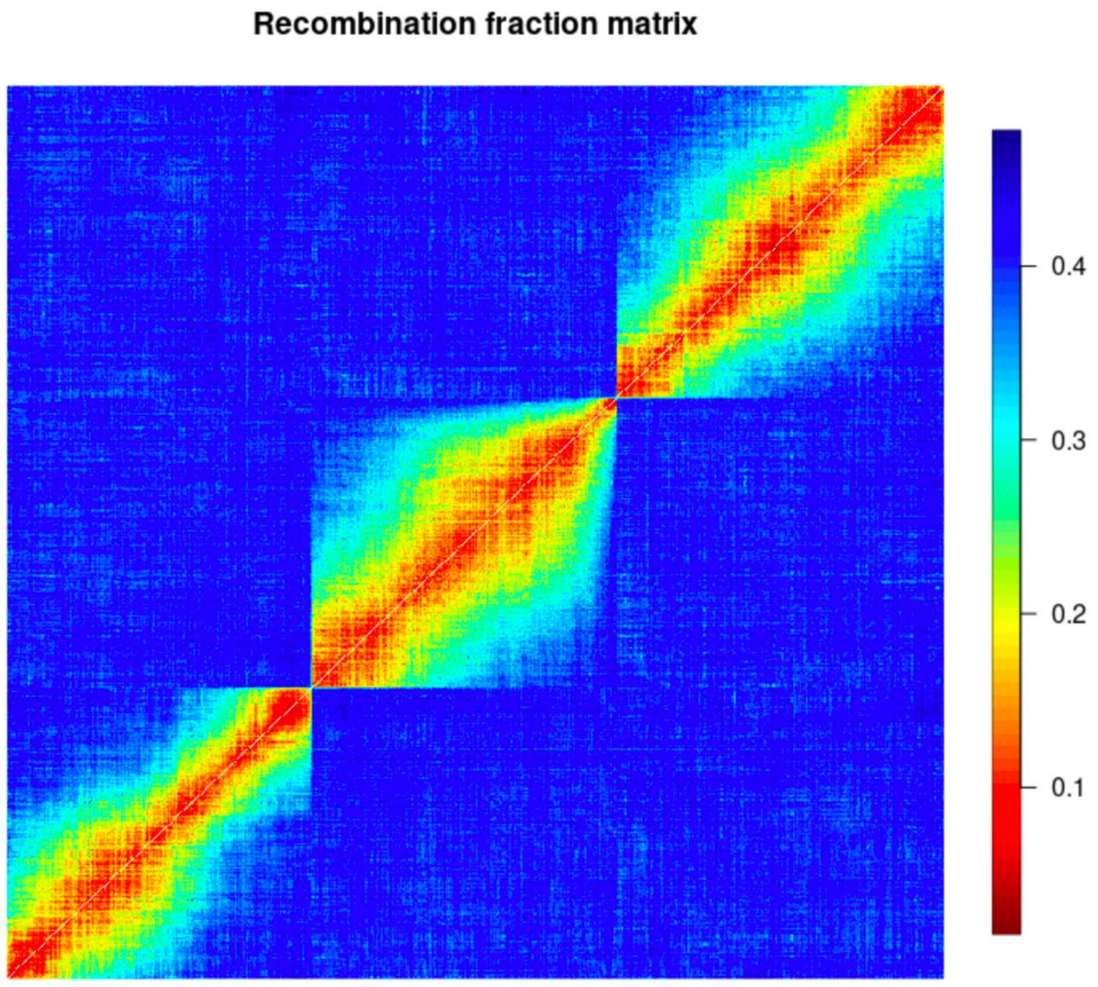




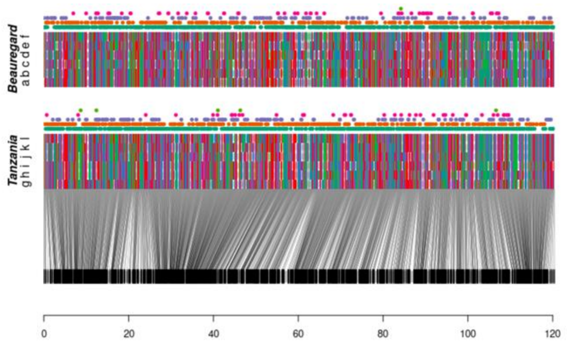


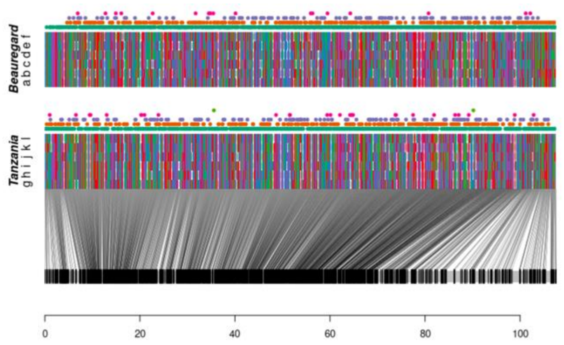
```
  This is an object of class 'mappoly.group'
  ------------------------------------------
  Criteria used to assign markers to groups:

    - Number of markers:          6216
    - Number of linkage groups:   3
    - Number of markers per linkage groups:
    group n.mrk
        1  2048
        2  2022
        3  2146
  ------------------------------------------
     3    9   12 NoChr
1 1996   30   22     0
2    7 1997   18     0
3    0    2 2144     0
  ------------------------------------------
```

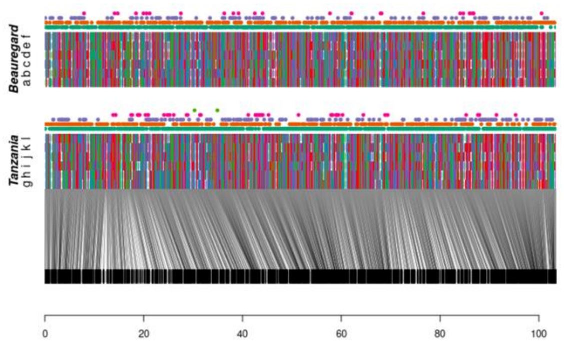
Ordered recombination fraction matrix

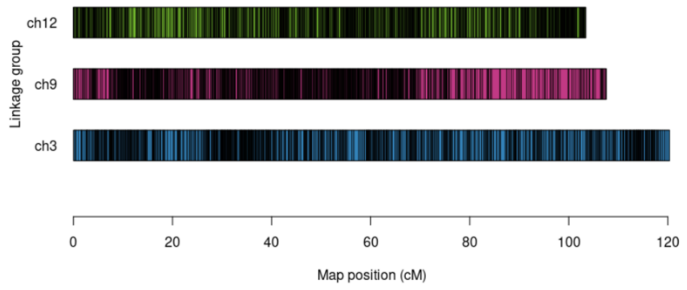


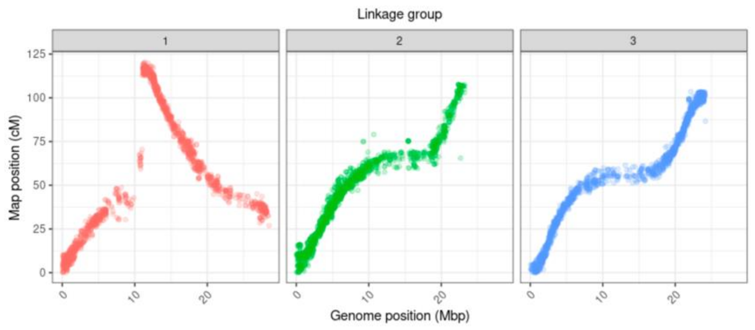
Map results

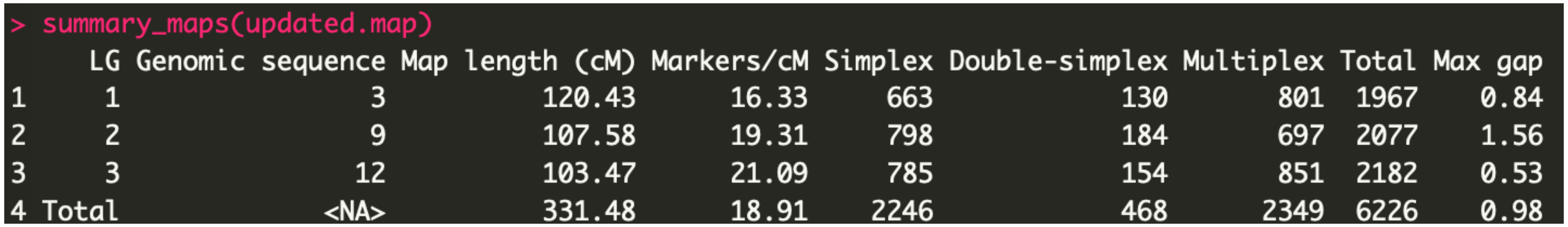








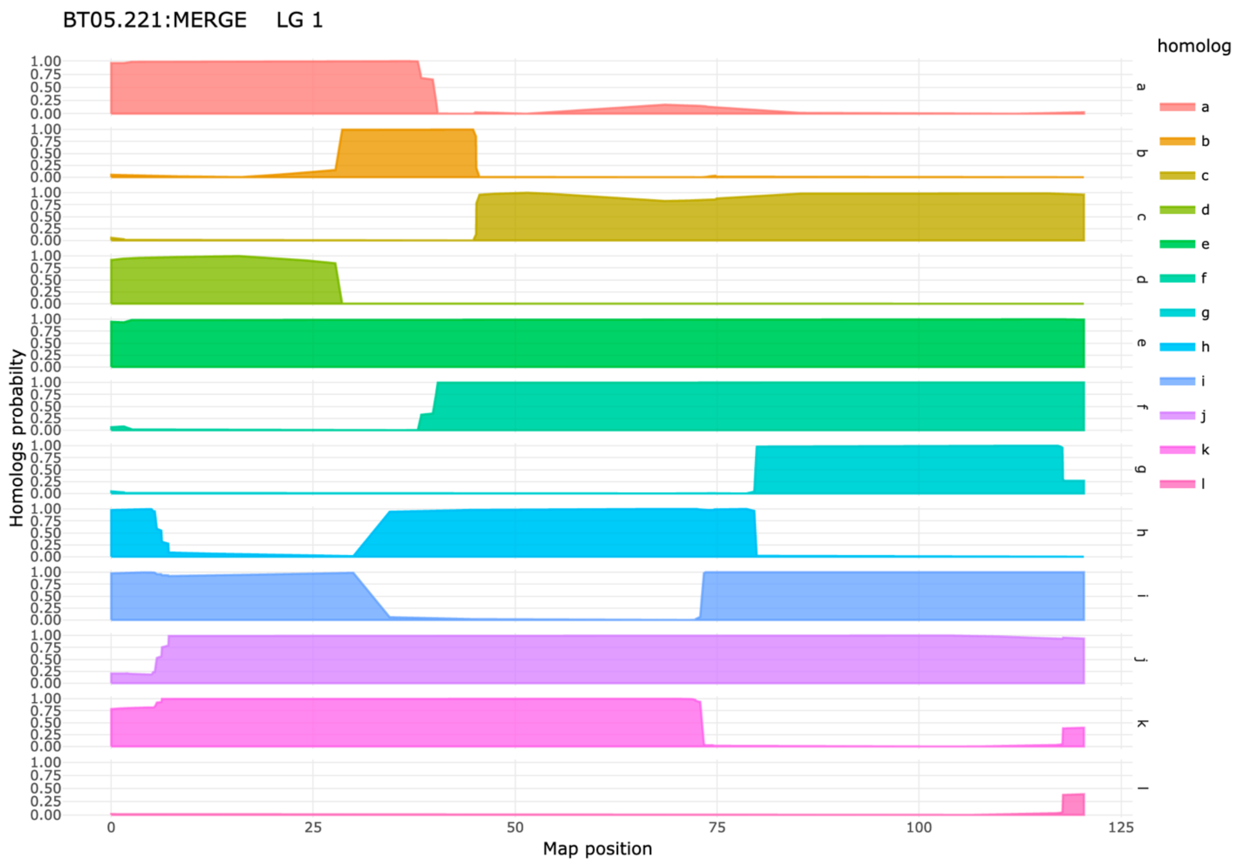




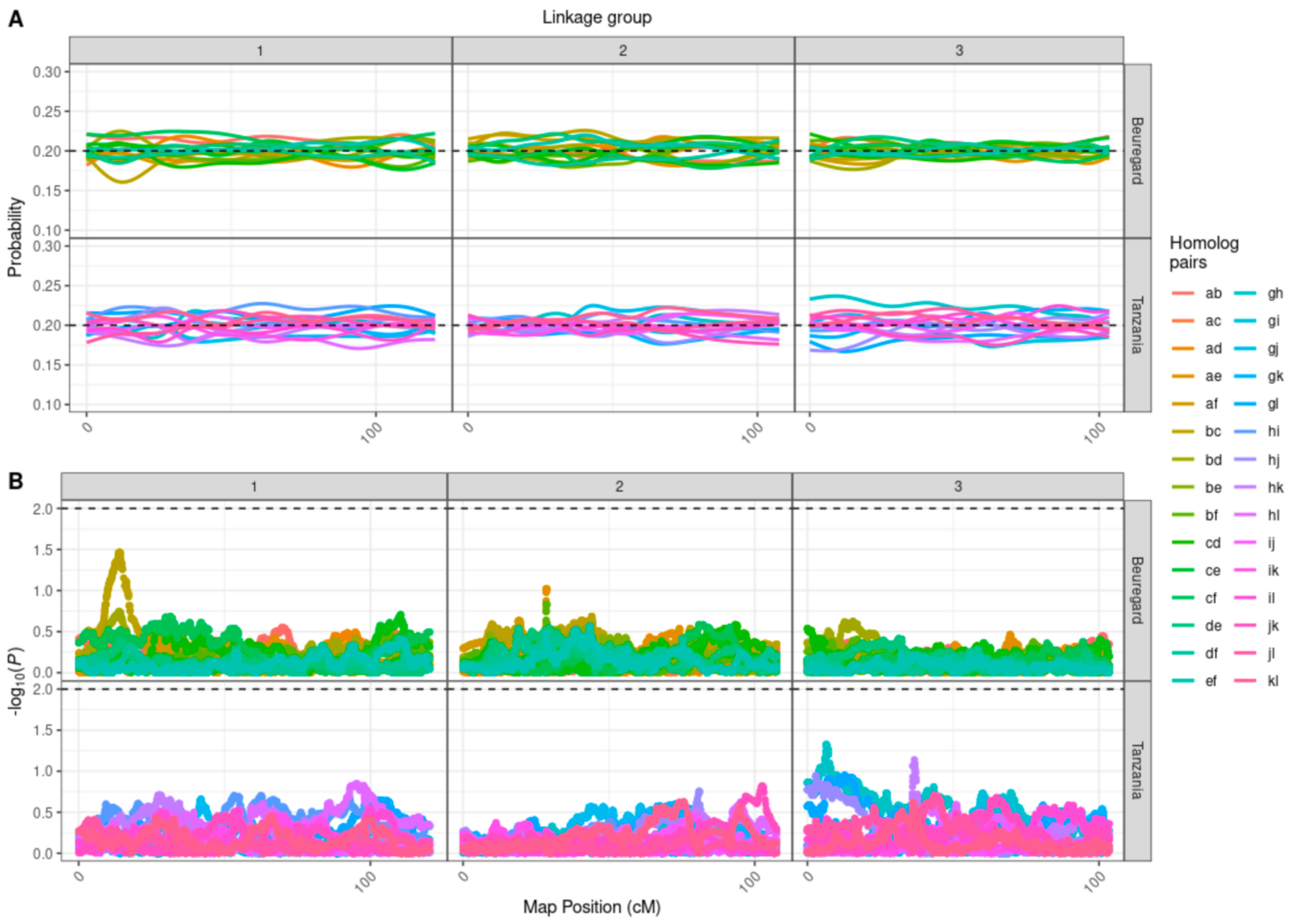
```
> summary_maps(updated.map)
     LG Genomic sequence Map length (cM) Markers/cM Simplex Double-simplex Multiplex Total Max gap
1     1                3          120.43      16.33     663            130       801  1967    0.84
2     2                9          107.58      19.31     798            184       697  2077    1.56
3     3               12          103.47      21.09     785            154       851  2182    0.53
4 Total             <NA>          331.48      18.91    2246            468      2349  6226    0.98
```

MAPpoly

Homolog probabilities



Preferential pairing profiles


A
Linkage group
Probability
Beuregard
Tanzania
B
-log10(P)
Map Position (cM)
Homolog pairs
ab ac ad ae af bc bd be bf cd ce cf de df ef
gh gi gj gk gl hi hj hk hl ij ik il jk jl kl