HOS 6236 Molecular Marker Assisted Plant Breeding

Create a map using software OneMap

Instructors:


```
Dr. Patricio R. Munoz
2211 Fifield Hall
352-273-4837
p.munoz@ufl.edu
Office Hours by appointment

Dr. Marcio F. Resende
2135 Fifield Hall
352-273-4772
mresende@ufl.edu
Office Hours by appointment

Dr. Kevin Folta
2239 Fifield Hall
352-273-4812
kfolta@ufl.edu
Office Hours by appointment
```


Teaching Assistant:


```
MSc. Ivone de Bem Oliveira
idebem.oliveira@ufl.edu
352-273-4836
Office Hours by appointment
```


Online resources for this tutorial

- https://github.com/augusto-garcia/onemap
- http://augustogarcia.me/onemap/vignettes_highres/Inbred_Based_Populations.html
- http://augustogarcia.me/onemap/vignettes_highres/Outcrossing_Populations.html

CONTENTS

TUTORIAL

Adapted from: Tutorial_Onemap_reduced_version.pdf Created by Ivone de Bem Oliveira

- OneMap is an environment for constructing linkage maps in several experimental crosses, including outcrossing (full-sib families derived from two non-homozygous parents), RILs, F2 and backcrosses. It is designed to be fully integrated with R/qtl package (Broman et al., 2008) and Windows QTL Cartographer (Wang et al., 2010), software to perform QTL analysis.
- Makes use of Wu et al. (2002) segregation patterns
- Allows the simultaneous estimation of linkage and linkage phases between markers, and was successfully applied in the analysis of sugarcane (Garcia et al., 2006; Oliveira et al., 2007) and Passiflora (Oliveira et al., 2008) data sets.
- Although no advanced knowledge in R is required to use OneMap

Installing and calling package

```
install.packages('devtools')
require(devtools)
install_github("augusto-garcia/onemap")
require(onemap)
```

Citation

```
citation('onemap')
```

```
## To cite OneMap in publications use:
##
##   Margarido, G. R. A., Souza, A. P. and Garcia, A. A. F. (2007)
##   OneMap: software for genetic mapping in outcrossing species.
##   Hereditas 144: 78-79.
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {OneMap: software for genetic mapping in outcrossing
species},
##     author = {Gabriel Rodrigues Alves Margarido and Anete Pereira {de
Souza} and Antonio Augusto Franco Garcia},
##     journal = {Hereditas},
##     volume = {144},
##     pages = {78-79},
##     year = {2007},
##   }
```

Genetic Map for Outcrossing (non-inbred) species with OneMap

- Same input that MAPMAKER/EXP uses - txt file, where the first line indicates the number of individuals and the number of markers. Then, the genotype information is included separately for each marker. The character "*" indicates the beginning of information input for a new marker, followed by the marker name. Next, there is a code indicating the marker type, according to Wu's et al. (2002) notation (pg 12). After each marker name, comes the genotype data for the segregating population (Fig. 1).

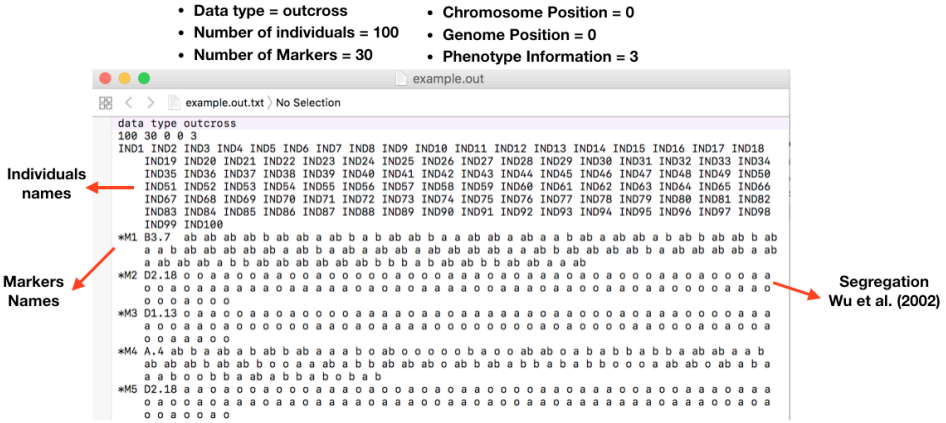


Fig. 1 Example of data entry in OneMap.

- The coding for marker genotypes used by OneMap is also the same one proposed by Wu et al. (2002) Marker types must be one of the following: A.1, A.2, A.3, A.4, B1.5, B2.6, B3.7, C.8, D1.9, D1.10, D1.11, D1.12, D1.13, D2.14, D2.15, D2.16, D2.17 or D2.18, each one corresponding to a row of the table 1. The letter and the number before the dot indicate the segregation type (i.e., 1:1:1:1, 1:2:1, 3:1 or 1:1), while the number after the dot indicates the observed bands in the offspring.

Table 1. Cross types that OneMap can handle including the expected segregation patterns (adapted from Wu et al., 2002).

			Parent			Offspring	
		crosstype	Cross	Observed bands		Observed bands	Segregation
A		1	$ab \times cd$	$ab \times cd$		ac, ad, bc, bd	1:1:1:1
		2	$ab \times ac$	$ab \times ac$		a, ac, ba, bc	1:1:1:1
		3	$ab \times co$	$ab \times c$		ac, a, bc, b	1:1:1:1
		4	$ao \times bo$	$a \times b$		ab, a, b, o	1:1:1:1
B	B_1	5	$ab \times ao$	$ab \times a$		$ab, 2a, b$	1:2:1
	B_2	6	$ao \times ab$	$a \times ab$		$ab, 2a, b$	1:2:1
	B_3	7	$ab \times ab$	$ab \times ab$		$a, 2ab, b$	1:2:1
C		8	$ao \times ao$	$a \times a$		$3a, o$	3:1
D	D_1	9	$ab \times cc$	$ab \times c$		ac, bc	1:1
		10	$ab \times aa$	$ab \times a$		a, ab	1:1
		11	$ab \times oo$	$ab \times o$		a, b	1:1
		12	$bo \times aa$	$b \times a$		ab, a	1:1
		13	$ao \times oo$	$a \times o$		a, o	1:1
	D_2	14	$cc \times ab$	$c \times ab$		ac, bc	1:1
		15	$aa \times ab$	$a \times ab$		a, ab	1:1
		16	$oo \times ab$	$o \times ab$		a, b	1:1
		17	$aa \times bo$	$a \times b$		ab, a	1:1
		18	$oo \times ao$	$o \times a$		a, o	1:1

- The input file must be saved in text format, with ".txt" extensions

Data files

https://github.com/augusto-garcia/onemap/tree/master/data

Constructing a genetic map for a outcross population

Example - Outcross Population

1 Entering the data

1.1 Reading outcross

```
example.out<- read_onemap(inputfile ='example.out.txt')
```

```
##  Working...
##
##  --Read the following data:
##  Type of cross:           outcross
##  Number of individuals:   100
##  Number of markers:       30
##  Chromosome information:  no
##  Position information:    no
##  Number of traits:        3
##  Missing trait values:
```

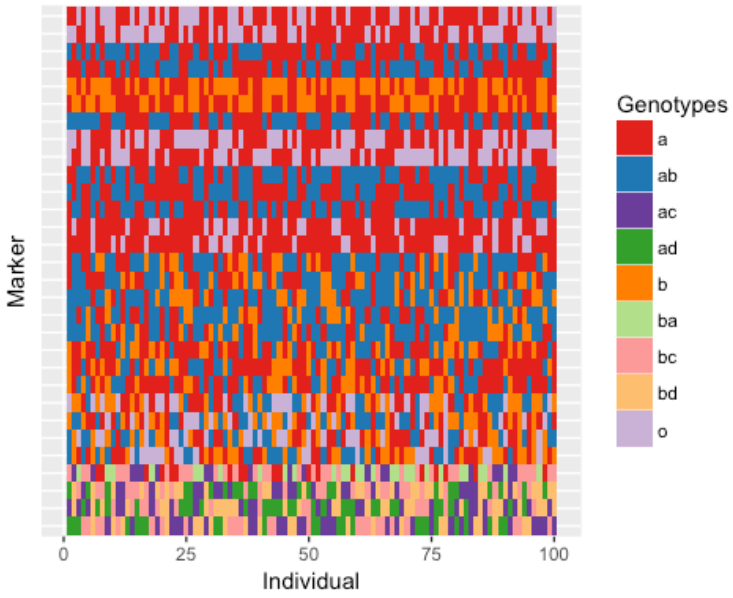
```
##    Pheno1: 0
##    Pheno2: 3
##    Pheno3: 0
```

1.2 Loading outcross file from the package data

```
data("example_out")
```

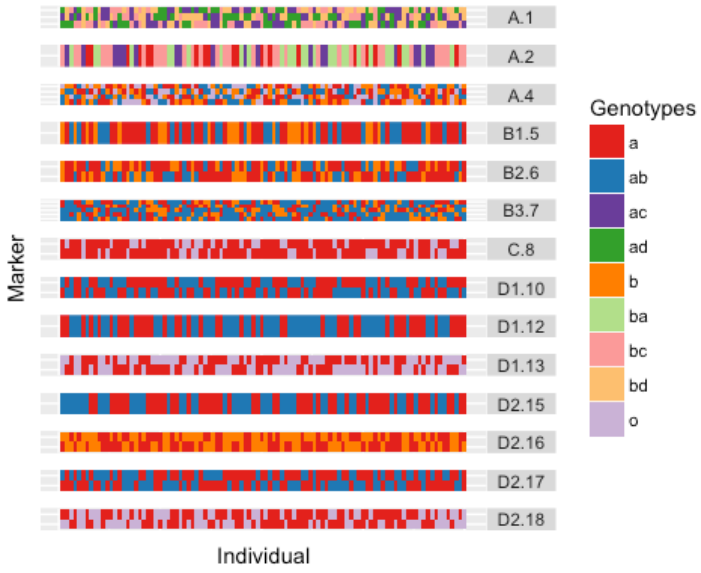
1.3 Plotting the fractions types

```
plot(example.out)
```



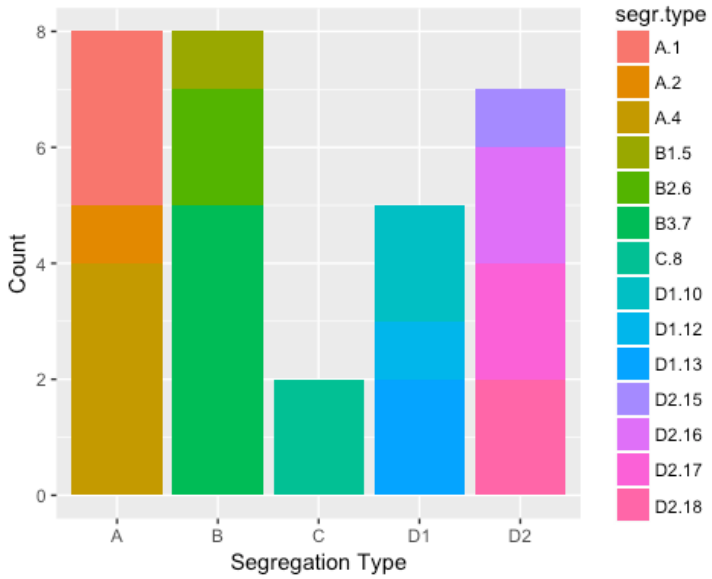
1.4 Plotting the recombination fractions by marker type

```
plot(example.out, all=FALSE)
```



1.5 Plotting the number of markers by segregation type

```
plot_by_segreg_type(example.out)
```



2 Testing segregation pattern for a specific marker

```
test_segregation_of_a_marker(example.out, 4)
```

2.1 Testing segregation pattern for all the markers

```
segreg_test <- test_segregation(example.out)
print(segreg_test)
```

```
##    Marker      H0 Chi-square      p-value % genot.
## 1      M1   1:2:1       1.76 4.147829e-01      100
## 2      M2     1:1       0.04 8.414806e-01      100
## 3      M3     1:1       0.36 5.485062e-01      100
## 4      M4 1:1:1:1       2.64 4.505201e-01      100
## 5      M5     1:1       1.96 1.615133e-01      100
## 6      M6   1:2:1       1.52 4.676664e-01      100
## 7      M7     1:1       0.16 6.891565e-01      100
## 8      M8   1:2:1       0.86 6.505091e-01      100
## 9      M9     1:1       0.04 8.414806e-01      100
## 10    M10     1:1       0.36 5.485062e-01      100
## 11    M11     1:1       0.16 6.891565e-01      100
## 12    M12 1:1:1:1       6.48 9.045460e-02      100
## 13    M13     3:1       0.00 1.000000e+00      100
## 14    M14 1:1:1:1       0.40 9.402425e-01      100
## 15    M15 1:1:1:1       2.24 5.241127e-01      100
## 16    M16     1:1       1.44 2.301393e-01      100
## 17    M17   1:2:1      35.58 1.878889e-08      100
## 18    M18 1:1:1:1       1.44 6.961859e-01      100
## 19    M19   1:2:1      37.98 5.659105e-09      100
```

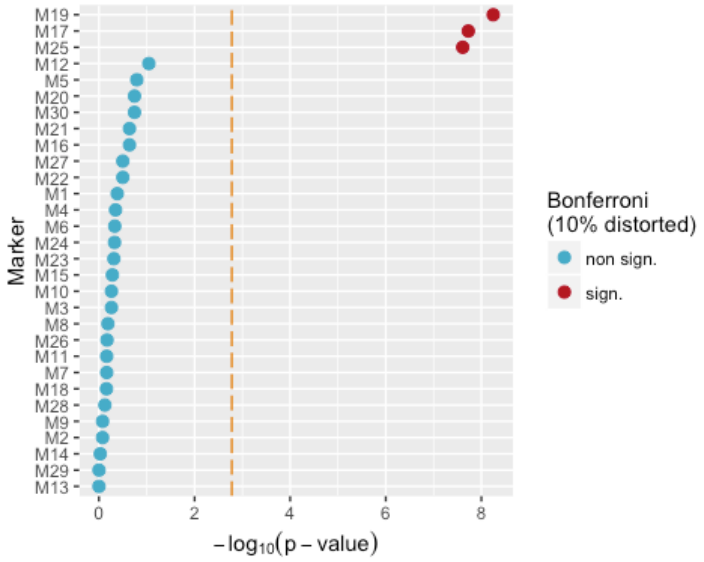
```
## 20    M20 1:1:1:1       4.88 1.807980e-01      100
## 21    M21     1:1       1.44 2.301393e-01      100
## 22    M22     1:1       1.00 3.173105e-01      100
## 23    M23     3:1       0.48 4.884223e-01      100
## 24    M24   1:2:1       1.50 4.723666e-01      100
## 25    M25   1:2:1      35.04 2.461278e-08      100
## 26    M26 1:1:1:1       1.52 6.776621e-01      100
## 27    M27     1:1       1.00 3.173105e-01      100
## 28    M28 1:1:1:1       1.20 7.530043e-01      100
## 29    M29     1:1       0.00 1.000000e+00      100
## 30    M30   1:2:1       3.42 1.808658e-01      100
```

2.2 Segregation distortion test and selection of markers that are not distorted

```
no.dist<-select_segreg(segreg_test, distorted = FALSE, numbers = TRUE)
```

2.3 Plotting the segregation test

```
plot(segreg_test)
```



3 Estimating two-point recombination fraction

- The first step is estimating the recombination fraction between all pairs of markers, using two-point tests
- For it we need a LOD, the program can suggest a LOD by:

3.1 Suggest LOD

```
(LOD_sug <- suggest_lod(example.out))

## [1] 3.230611
```

3.2 Estimation of the recombination fractions

```
twopts<- rf_2pts(example.out, LOD=3)

## Computing 435 recombination fractions ...
```

3.3 Verifying the recombination fraction (RF) object

```
twopts

##   This is an object of class 'rf_2pts'
##
##   Criteria: LOD = 3 , Maximum recombination fraction = 0.5
##
##   This object is too complex to print
##   Type 'print(object, c(mrk1=marker, mrk2=marker))' to see
##     the analysis for two markers
##     mrk1 and mrk2 can be the names or numbers of both markers
```

3.3.1 To see the RF analysis for two markers

```
print(twopts, c("M1","M3"))

##   Results of the 2-point analysis for markers: M1 and M3
##   Criteria: LOD =  3 , Maximum recombination fraction =  0.5
##
##            rf      LOD
## CC 0.2954514 1.646878
## CR 0.2954514 1.646878
## RC 0.7045486 1.646878
## RR 0.7045486 1.646878
```

- Each line corresponds to a possible linkage phase. 1 denotes coupling phase in both parents (CC), 2 and 3 denote coupling phase in parent 1 and 2, respectively, and repulsion in the other (CR and RC), and 4 denotes repulsion phase in both parents (RR). Theta is the maximum likelihood estimate of the recombination fraction, with its LOD Scores.

- Once the recombination fractions and linkage phases for all pairs of markers have been estimated and tested, markers can be assigned to linkage groups. To do this, first use the function make_seq to create a sequence with the markers you want to assign

4 Create a sequence contaning informations about the markers

```
mark.all<-make_seq(twopts,"all")
```

- The function make_seq is used to create sequences from objects of several kinds. Here, the object is of class rf.2pts and the second argument specifies which markers you want to use. In this example, the argument "all" indicates that all markers will be analyzed. If you want to use only a subset of markers, say M1 and

M2, the option will be c(1,2). These numbers refer to the lines where markers are located on the data file. Since the identification of the markers can be cumbersome, you should use the function marker type to see their numbers, names and types

4.1 Verifying the markers type (using only the markers that haven't presented distortion)

```
mark.all<-mark.new
marker_type(mark.all)

##   Marker 1 ( M1 ) is of type B3.7
##   Marker 2 ( M2 ) is of type D2.18
##   Marker 3 ( M3 ) is of type D1.13
##   Marker 4 ( M4 ) is of type A.4
##   Marker 5 ( M5 ) is of type D2.18
##   Marker 6 ( M6 ) is of type B3.7
##   Marker 7 ( M7 ) is of type D2.15
##   Marker 8 ( M8 ) is of type B3.7
##   Marker 9 ( M9 ) is of type D1.10
##   Marker 10 ( M10 ) is of type D2.17
##   Marker 11 ( M11 ) is of type D2.16
##   Marker 12 ( M12 ) is of type A.2
##   Marker 13 ( M13 ) is of type C.8
##   Marker 14 ( M14 ) is of type A.4
##   Marker 15 ( M15 ) is of type A.4
##   Marker 16 ( M16 ) is of type D2.17
##   Marker 18 ( M18 ) is of type A.1
##   Marker 20 ( M20 ) is of type A.1
##   Marker 21 ( M21 ) is of type D2.16
##   Marker 22 ( M22 ) is of type D1.10
##   Marker 23 ( M23 ) is of type C.8
##   Marker 24 ( M24 ) is of type B3.7
##   Marker 26 ( M26 ) is of type A.1
##   Marker 27 ( M27 ) is of type D1.12
##   Marker 28 ( M28 ) is of type A.4
##   Marker 29 ( M29 ) is of type D1.13
##   Marker 30 ( M30 ) is of type B3.7
```

4.2 Example: Create sequences only for the markers M1, M2 and M3

```
mark.1_3<-make_seq(twopts,c(1,2,3))
marker_type(mark.1_3)

##   Marker 1 ( M1 ) is of type B3.7
##   Marker 2 ( M2 ) is of type D2.18
##   Marker 3 ( M3 ) is of type D1.13
```

5 Assign markers to linkage groups

5.1 Grouping

```
LGs<- group(mark.all, LOD = 3, max.rf = 0.4)

##    Selecting markers:
##    group    1
##     ............
##    group    2
##     .........
##    group    3
##     .....
```

- For this function, optional arguments are LOD and max.rf, which define thresholds to be used when assigning markers to linkage groups. If none provided (default), criteria previously defined for the object twopts are used.

5.2 Basic information about the groups (such as the number of groups, number of linked markers, etc):

```
print(LGs)

##   This is an object of class 'group'
##   It was generated from the object "mark.all"
##
##   Criteria used to assign markers to groups:
##     LOD = 3 , Maximum recombination fraction = 0.5
##
##   No. markers:            27
##   No. groups:             3
##   No. linked markers:     27
##   No. unlinked markers:   0
##
##   Printing groups:
##   Group 1 : 13 markers
##     M1 M2 M3 M5 M6 M10 M11 M12 M14 M15 M26 M28 M30
##
##   Group 2 : 9 markers
##     M4 M9 M16 M20 M21 M23 M24 M27 M29
##
##   Group 3 : 5 markers
##     M7 M8 M13 M18 M22
```

5.2.1 Using a higher threshold to assign the groups

```
LGs2<- group(mark.all, LOD = 6)

##    Selecting markers:
##    group    1
```

```
##     ............
##    group    2
##     .......
##    group    3
##     ....

print(LGs2)

##   This is an object of class 'group'
##   It was generated from the object "mark.all"
##
##   Criteria used to assign markers to groups:
##     LOD = 6 , Maximum recombination fraction = 0.5
##
##   No. markers:            27
##   No. groups:             3
##   No. linked markers:     24
##   No. unlinked markers:   3
##
##   Printing groups:
##   Group 1 : 13 markers
##     M1 M2 M3 M5 M6 M10 M11 M12 M14 M15 M26 M28 M30
##
##   Group 2 : 7 markers
##     M4 M9 M16 M20 M21 M23 M27
##
##   Group 3 : 4 markers
##     M8 M13 M18 M22
##
##   Unlinked markers: 3  markers
##     M7 M24 M29
```

- Once marker assignment to linkage groups is finished, the mapping step can take place. First of all, you must set the mapping function that should be used to display the genetic map through the analysis. You can choose between Kosambi or Haldane mapping functions

6 Set map function

```
set_map_fun(type="haldane")  #or set_map_fun(type="kosambi")
```

- Now, you must define which linkage group will be mapped. In other words, a linkage group must be "extracted" from the object of class group (i.e. the LGs object), in order to be mapped. For simplicity, we will start here with the smallest one, which is linkage group 3.

7 Defining group (i.e group 3 - LG3)

```
LG3<- make_seq(LGs, 3)
```

7.1 Information about the LG3

```
LG3

##
## Number of markers: 5
## Markers in the sequence:
## M7 M8 M13 M18 M22
##
## Parameters not estimated.
```

8 Ordering markers

- To order these markers, you can use a two-point based algorithm such as Seriation (Bue- tow and Chakravarti, 1987), Rapid Chain Delineation (Doerge, 1996), Recombination Counting and Ordering (Van Os et al., 2005) and Unidirectional Growth (Tan and Fu, 2006)

8.1 Choosing the order algorithm

```
LG3.ser<-seriation(LG3)

##
## order obtained using SERIATION algorithm:
##
##  7 22 13 8 18
##
## calculating multipoint map using tol =  1e-04 .

LG3.rcd<- rcd(LG3)

##
## order obtained using RCD algorithm:
##
##  7 22 8 13 18
##
## calculating multipoint map using tol =  1e-04 .

LG3.rec<-record(LG3)

##
## order obtained using RECORD algorithm:
##
##  7 22 13 8 18
##
## calculating multipoint map using tol 1e-04 .

LG3.ug<- ug(LG3)

##
## order obtained using UG algorithm:
##
```

```
##  7 22 13 8 18
##
## calculating multipoint map using tol  1e-04 .
```

8.2 Comparing all possible orders (exhaustive search), the function compare can be used

```
LG3.comp<- compare(LG3) #careful because it can take a long time

##
## Comparing 60 orders:
##
  |======================================================================|
100%
```

- This order step can take some time, depending on marker types in the linkage group. In the example, LG3 contains one marker of type D1 and one of type D2, besides one marker segregating in 3:1 fashion (type C). Thus, although the number of possible orders is relatively small (60), for each order there are various possible combinations of linkage phases. Also, the convergence of the EM algorithm takes considerably more time, since markers of type C are not very informative.
- The first argument to compare function is an object of class sequence (the extracted group LG3), and the object generated by this function is of class compare.

8.2.1 To see the results of the previous step:

```
LG3.comp

##
## Number of orders: 50
## Best 30 unique orders                 LOD    Nested LOD
## -----------------------------------------------------
## order  1:    22   7  13   8  18
##         CC  CC  RR  RR         0.00       0.00
##         CC  RC  RR  RR        -5.20      -5.20
## -----------------------------------------------------
## order  2:     7  22  13   8  18
##         CC  CC  RR  RR         0.00       0.00
##         CC  CR  RR  RR        -5.22      -5.22
## -----------------------------------------------------
## order  3:     7  18   8  13  22
##         CC  RR  RR  CC        -0.74       0.00
## -----------------------------------------------------
## order  4:     7  13   8  18  22
##         CC  RR  RR  CC        -0.93       0.00
## -----------------------------------------------------
## order  5:    22   7  18   8  13
##         CC  CC  RR  RR        -1.52       0.00
```

```
##         CC  RC  RR  RR        -5.18      -3.67
## ----------------------------------------------------
## order  6:     7  22  18   8  13
##         CC  CC  RR  RR        -1.52       0.00
##         CC  CR  RR  RR        -5.22      -3.71
## ----------------------------------------------------
## order  7:    22   7   8  13  18
##         CC  RR  RR  CC        -4.97       0.00
##         CC  CR  RR  CC        -8.99      -4.02
## ----------------------------------------------------
## order  8:     7  22   8  13  18
##         CC  RR  RR  CC        -4.97       0.00
##         CC  RC  RR  CC        -9.01      -4.04
## ----------------------------------------------------
## order  9:     7  18  13   8  22
##         CC  CC  RR  RC        -4.99       0.00
## ----------------------------------------------------
## order 10:    22   7  13  18   8
##         CC  CC  CC  RR        -5.03       0.00
##         CC  RC  CC  RR        -9.96      -4.93
## ----------------------------------------------------
## order 11:     7  22  13  18   8
##         CC  CC  CC  RR        -5.03       0.00
##         CC  CR  CC  RR        -9.99      -4.96
## ----------------------------------------------------
## order 12:     7   8  13  18  22
##         CR  RR  CC  CC        -5.22       0.00
## ----------------------------------------------------
## order 13:    22   7  18  13   8
##         CC  CC  CC  RR        -5.30       0.00
##         CC  RC  CC  RR        -8.97      -3.67
## ----------------------------------------------------
## order 14:     7  22  18  13   8
##         CC  CC  CC  RR        -5.30       0.00
##         CC  CR  CC  RR        -9.01      -3.71
## ----------------------------------------------------
## order 15:     7  13  18   8  22
##         CC  CC  RR  RC        -5.50       0.00
## ----------------------------------------------------
## order 16:     7   8  18  13  22
##         CR  RR  CC  CC        -5.75       0.00
## ----------------------------------------------------
## order 17:    22   7   8  18  13
##         CC  RR  RR  CC        -6.03       0.00
##         CC  CR  RR  CC        -9.97      -3.94
## ----------------------------------------------------
## order 18:     7  22   8  18  13
##         CC  RR  RR  CC        -6.03       0.00
##         CC  RC  RR  CC        -9.99      -3.96
## ----------------------------------------------------
```

```
## order 19:    18   8   7  13  22
##         RR  CR  RC  CC        -9.60       0.00
##         RR  CR  CC  CC       -13.87      -4.26
## ------------------------------------------------------
## order 20:     7  13  22   8  18
##         CC  CC  RR  RR        -9.85       0.00
##         CC  CC  RC  RR       -14.37      -4.51
## ------------------------------------------------------
## order 21:     8  18   7  13  22
##         RR  CC  CC  CC       -10.34       0.00
##         RR  CC  RC  CC       -13.62      -3.28
## ------------------------------------------------------
## order 22:     7  13  22  18   8
##         CC  CC  CC  RR       -10.40       0.00
##         CC  CC  CR  RR       -14.37      -3.97
## ------------------------------------------------------
## order 23:    18   8   7  22  13
##         RR  CR  RC  CC       -10.70       0.00
##         RR  CR  CC  CC       -14.65      -3.94
## ------------------------------------------------------
## order 24:    13   7  22   8  18
##         CC  CC  RR  RR       -10.70       0.00
## ------------------------------------------------------
## order 25:     8  18   7  22  13
##         RR  CC  CC  CC       -10.96       0.00
##         RR  CC  RC  CC       -14.62      -3.67
## ------------------------------------------------------
## order 26:    13   7  22  18   8
##         CC  CC  CC  RR       -10.96       0.00
## ------------------------------------------------------
## order 27:     7  18   8  22  13
##         CC  RR  RC  CR       -11.00       0.00
##         CC  RR  RC  CC       -14.13      -3.13
## ------------------------------------------------------
## order 28:     7   8  18  22  13
##         CR  RR  CC  CC       -11.51       0.00
##         CR  RR  CC  CR       -14.23      -2.72
## ------------------------------------------------------
## order 29:    13   7   8  18  22
##         CC  RR  RR  CC       -13.09       0.00
## ------------------------------------------------------
## order 30:    13   7  18   8  22
##         CC  CC  RR  RC       -13.60       0.00
## ------------------------------------------------------
```

- By default, OneMap stores 50 orders, which may or may not be unique. The value of LOD refers to the overall LOD Score, considering all orders tested. Nested LOD

refers to LOD Scores within a given order, i.e., scores for different combinations of linkage phases for the same marker order.

- For example, order 1 has the largest value of log-likelihood and, therefore, its LOD Score is zero for a given combination of linkage phases (CC, CC, RR, RR). For this same order and other linkage phases, LOD Score is -2.43. Analyzing the results for order 2, notice that its highest LOD Score is very close to zero, indicating that this order is also quite plausible. Notice also that Nested LOD will always contain at least one zero value, corresponding to the best combination of phases for markers in a given order. Due to the information provided by two-point analysis, not all combinations are tested and that is the reason why the number of Nested LOD is different for each order.

9 Choosing the order to compute the final map for LG3:

9.1 Selecting order with the highest likelihood

```
LG3.final<- make_seq(LG3.comp,1,1)
```

- The first argument is the object of class compare. The second argument indicates which order is chosen: 1 is for the order with highest likelihood, 2 is for the second best, and so on. The third argument indicates which combination of phases is chosen for a given order: 1 also means the combination with highest likelihood among all combinations of phases (based on Nested LOD).

10 Final map

```
LG3.final

##
## Printing map:
##
## Markers           Position           Parent 1       Parent 2
##
## 22 M22               0.00           a |  | b       a |  | a
##  7 M7                0.00           a |  | a       a |  | b
## 13 M13              48.22           a |  | o       a |  | o
##  8 M8               52.70           b |  | a       b |  | a
## 18 M18              58.91           a |  | b       c |  | d
##
## 5 markers           log-likelihood: -318.8464
```

- At the leftmost position, marker names are displayed. Position shows the cumulative distance using the Haldane mapping function. Finally, Parent 1 and Parent 2 show the diplotypes of both parents, that is, the manner in which alleles are arranged in the chromosomes, given the estimated linkage phase. Notation is the same as that used by Wu et al. (2002). Details about how ordering algorithms can be chosen and used are presented by Mollinari et al. (2009).

Creating a genetic map for a more complex group

1 Extracting the group from the object LGs (i.e LG2 bigger number of markers)

```
LG2.rcd<- rcd(LG2)
## order obtained using RCD algorithm:
##
##  20 4 23 9 21 27 16 29 24
##
## calculating multipoint map using tol =  1e-04 .
```

1.1 Get a preliminary order estimate using rcd

```
LG2.rcd<- rcd(LG2)
## order obtained using RCD algorithm:
##
##  23 4 20 27 21 9 16 29 24
##
## calculating multipoint map using tol =  1e-04 .
```

1.1.1 Check the segregation types of all markers in this group

```
marker_type(LG2)

##   Marker 4 ( M4 ) is of type A.4
##   Marker 9 ( M9 ) is of type D1.10
##   Marker 16 ( M16 ) is of type D2.17
##   Marker 20 ( M20 ) is of type A.1
##   Marker 21 ( M21 ) is of type D2.16
##   Marker 23 ( M23 ) is of type C.8
##   Marker 24 ( M24 ) is of type B3.7
##   Marker 27 ( M27 ) is of type D1.12
##   Marker 29 ( M29 ) is of type D1.13
```

- Based on their segregation types and distribution on the preliminary map, markers M4 , M20 and M24 and M23 are the most informative ones (type A is the better, followed by type B,…)

2 Create a framework of ordered markers using compare for the most informative markers

```
LG2.init<- make_seq(twopts, c(4,20,24, 23))
LG2.comp<- compare(LG2.init)
## Comparing 12 orders:

  |============================================================|
100%

LG2.comp
## Number of orders: 18
```

```
## Best 12 unique orders             LOD    Nested LOD
## ------------------------------------------------
## order  1:    20   4  23  24
##         CR  CR  CR         0.00       0.00
##         CR  CR  RC        -1.55      -1.55
## ------------------------------------------------
## order  2:    23   4  20  24
##         CR  CR  CR        -0.88       0.00
## ------------------------------------------------
## order  3:     4  20  23  24
##         CR  CC  CR        -3.32       0.00
##         CR  CC  RC        -4.36      -1.04
## ------------------------------------------------
## order  4:    24   4  20  23
##         CC  CR  CC        -3.39       0.00
## ------------------------------------------------
## order  5:    20   4  24  23
##         CR  CC  CR        -6.73       0.00
##         CR  CC  RC        -9.81      -3.08
## ------------------------------------------------
## order  6:     4  20  24  23
##         CR  CR  CR        -8.45       0.00
##         CR  CR  RC        -9.36      -0.91
## ------------------------------------------------
## order  7:    24   4  23  20
##         CC  CR  CC       -10.77       0.00
## ------------------------------------------------
## order  8:     4  23  20  24
##         CR  CC  CR       -11.07       0.00
## ------------------------------------------------
## order  9:    23   4  24  20
##         CR  CC  CR       -25.60       0.00
## ------------------------------------------------
## order 10:     4  23  24  20
##         CR  CR  CR       -25.68       0.00
##         CR  RC  CR       -28.36      -2.69
## ------------------------------------------------
## order 11:     4  24  23  20
##         CC  CR  CC       -27.66       0.00
##         CC  RC  CC       -30.87      -3.21
## ------------------------------------------------
## order 12:     4  24  20  23
##         CC  CR  CC       -28.40       0.00
## ------------------------------------------------
```

3 Selecting the best order

```
(LG2.frame<- make_seq(LG2.comp,1,1))
## Printing map:
##
```

```
## Markers           Position           Parent 1       Parent 2
##
## 20 M20               0.00           a |  | b       c |  | d
##  4 M4               14.38           a |  | o       o |  | b
## 23 M23              35.28           a |  | o       a |  | o
## 24 M24              83.99           a |  | b       b |  | a
##
## 4 markers          log-likelihood: -357.8632
```

4 Trying to map the remaining markers one at a time

```
LG2.extend<-try_seq(LG2.frame,9)

##  9 --> M9  : .....

LG2.extend
## LOD scores correspond to the best linkage phase combination
## for each position
##
## The symbol "*" outside the box indicates that more than one
## linkage phase is possible for the corresponding position
##
##
##        Marker tested: 9
##
##        Markers     LOD
##      ====================
##      |                   |
##      |            -5.91  |  1
##      |  20               |
##      |           -15.80  |  2  *
##      |   4               |
##      |            -2.46  |  3  *
##      |  23               |
##      |             0.00  |  4  *
##      |  24               |
##      |            -6.18  |  5
##      |                   |
##      ====================
```

- Based on the LOD Scores, marker M9 is probably better located between markers M23 and M24. However, the "*" symbol indicates that more than one linkage phase is possible. Detailed results can be seen with

Print the map with M9 in the position 4

```
print(LG2.extend,4)

##
## LOD is the overall LOD score (among all orders)
##
```

```
## NEST.LOD is the LOD score within the order
##
## Marker tested: 9
## ---------------------
## |      |       |      |
## | 20   |       |      |
## |      |  CR   |  CR  |
## |  4   |       |      |
## |      |  CR   |  CR  |
## | 23   |       |      |
## |      |  CC   |  CC  |
## |  9   |       |      |
## |      |  CR   |  CC  |
## | 24   |       |      |
## |      |       |      |
## |------------------|
## | LOD |   0.0|  -3.6|
## |------------------|
## |NEST.|      |      |
## | LOD |   0.0|  -3.6|
## ---------------------
```

4.1 Saving the best order

```
(LG2.frame <- make_seq(LG2.extend,4,1))

##
## Printing map:
##
## Markers           Position           Parent 1       Parent 2
##
## 20 M20               0.00           a |  | b       c |  | d
##  4 M4               14.38           a |  | o       o |  | b
## 23 M23              35.15           a |  | o       a |  | o
##  9 M9               51.33           a |  | b       a |  | a
## 24 M24              87.12           a |  | b       b |  | a
##
## 5 markers           log-likelihood: -401.5487
```

- When using make.seq with an object of class try, the second argument is the position on the map (according to the scale on the right of the output) and the last argument indicates linkage phases (defaults to 1, higher nested LOD).

5 Adding other markers, one by one

```
LG2.extend <- try_seq(LG2.frame,29)

## 29 --> M29 : ......

LG2.frame <- make_seq(LG2.extend,6)
LG2.extend <- try_seq(LG2.frame,27)
```

```
## 27 --> M27 : .......

LG2.frame <- make_seq(LG2.extend,1)
LG2.extend <- try_seq(LG2.frame, 16)

## 16 --> M16 : ........

LG2.frame <- make_seq(LG2.extend,2)
LG2.extend <- try_seq(LG2.frame,21)

## 21 --> M21 : .........

LG2.final <- make_seq(LG2.extend,5)
```

6 Automated process

6.1 Automated sequentially addition of markers

```
LG2.ord<- order_seq(LG2, n.init = 5, THRES = 3, touchdown = TRUE)

##
## Cross type: outcross
## Using segregation types of the markers to choose initial subset
##
## Comparing 60 orders:
##
  |======================================================================|
100%
##
##
## Running try algorithm
## 27 --> M27 : ......
## 29 --> M29 : .......
## 16 --> M16 : .......
## 21 --> M21 : .......
##
## LOD threshold = 3
##
## Positioned markers: 27 20 4 21 23 9 24
##
## Markers not placed on the map: 16 29
##
## Trying to map remaining markers with LOD threshold  2
## 16 --> M16 : ........
## 29 --> M29 : ........
##
## LOD threshold = 2
##
## Positioned markers: 27 20 4 21 23 9 24 29
##
## Markers not placed on the map: 16
##
```

```
##
## Calculating LOD-Scores
## 16 --> M16 : .........
##
##
## Placing remaining marker(s) at most likely position
## 16 --> M16 : .........
##
## Estimating final genetic map using tol = 10E-5.
```

- Basically, this function automates what the try_seq function does, using some pre-defined rules. In the function, n.init = 5 means that five markers (the most informative ones) will be used in the compare step; THRES = 3 indicates that the try_seq step will only add markers to the sequence which can be mapped with LOD Score greater than 3; draw.try=FALSE will not display a diagnostic graphic for each try_seq step; if TRUE wait=1 indicates the minimum time interval in seconds to display the diagnostic graphic.

- NOTE: Although very useful, this function can be misleading, specially if there are not many fully informative markers, so use it carefully. Results can vary for each running, of course.

6.2 Check the final order

```
LG2.ord

##
## Best sequence found.
## Printing map:
##
## Markers           Position           Parent 1       Parent 2
##
## 27 M27               0.00           b |  | o       a |  | a
## 20 M20              25.54           a |  | b       c |  | d
##  4 M4               39.92           a |  | o       o |  | b
## 21 M21              55.49           o |  | o       a |  | b
## 23 M23              60.98           a |  | o       a |  | o
##  9 M9               79.45           a |  | b       a |  | a
## 24 M24             114.11           a |  | b       b |  | a
## 29 M29             141.39           o |  | a       o |  | o
##
## 8 markers            log-likelihood: -540.1795
##
##



##
## The following markers could not be uniquely positioned.
## Printing most likely positions for each unpositioned marker:
##
```

```
## ------------
##  |     |  16 |
##  |-----|-----|
##  |     | **  |
##  |  27 |     |
##  |     | *** |
##  |  20 |     |
##  |     |     |
##  |   4 |     |
##  |     |     |
##  |  21 |     |
##  |     |     |
##  |  23 |     |
##  |     |     |
##  |   9 |     |
##  |     |     |
##  |  24 |     |
##  |     |     |
##  |  29 |     |
##  |     |     |
## ------------
##
## '***' indicates the most likely position(s) (LOD = 0.0)
##
## '**' indicates very likely positions (LOD > -1.0)
##
## '*' indicates likely positions (LOD > -2.0)
```

- Note that markers 16 and 29 could not be safely mapped to a single position (LOD Score > THRES in absolute value). The output displays the "safe" order and the most likely positions for markers not mapped, where "*" indicates the most likely position and also to other plausible positions.

7 To get the safe order

```
LG2.safe<- make_seq(LG2.ord, "safe")
```

8 To get all the markers

```
LG2.all<- make_seq(LG2.ord, "force")
```

- Notice that, for this linkage group, the "forced" map obtained with order_seq is the same as that obtained with compare plus try.seq, but this is not always the case

- The order_seq function can also performs two rounds of the try.seq algorithms, first using THRES defined by user and then THRES - 1 as threshold. This generally results in safe orders with more markers mapped, but may take longer to run.

9 Using the touchdown options:

```
LG2.ord2 <- order_seq(LG2, n.init=5, THRES=3, touchdown=TRUE)

##
## Cross type: outcross
## Using segregation types of the markers to choose initial subset
##
## Comparing 60 orders:

  |======================================================================|
100%
##
## Running try algorithm
## 27 --> M27 : ......
## 29 --> M29 : .......
## 16 --> M16 : .......
## 21 --> M21 : .......
##
## LOD threshold = 3
##
## Positioned markers: 27 20 4 21 23 9 24
##
## Markers not placed on the map: 16 29
##
##
##
## Trying to map remaining markers with LOD threshold  2
## 16 --> M16 : ........
## 29 --> M29 : ........
##
## LOD threshold = 2
##
## Positioned markers: 27 20 4 21 23 9 24 29
##
## Markers not placed on the map: 16
##
##
## Calculating LOD-Scores
## 16 --> M16 : .........
##
##
## Placing remaining marker(s) at most likely position
## 16 --> M16 : .........
##
## Estimating final genetic map using tol = 10E-5.

LG2.ord2

##
## Best sequence found.
```

```
## Printing map:
##
## Markers           Position           Parent 1       Parent 2
##
## 27 M27               0.00           b |  | o       a |  | a
## 20 M20              25.54           a |  | b       c |  | d
##  4 M4               39.92           a |  | o       o |  | b
## 21 M21              55.49           o |  | o       a |  | b
## 23 M23              60.98           a |  | o       a |  | o
##  9 M9               79.45           a |  | b       a |  | a
## 24 M24             114.11           a |  | b       b |  | a
## 29 M29             141.39           o |  | a       o |  | o
##
## 8 markers            log-likelihood: -540.1795
##
##
##
## The following markers could not be uniquely positioned.
## Printing most likely positions for each unpositioned marker:
##
## ------------
## |    |  16 |
## |----|-----|
## |    | **  |
## | 27 |     |
## |    | *** |
## | 20 |     |
## |    |     |
## |  4 |     |
## |    |     |
## | 21 |     |
## |    |     |
## | 23 |     |
## |    |     |
## |  9 |     |
## |    |     |
## | 24 |     |
## |    |     |
## | 29 |     |
## |    |     |
## ------------
##
## '***' indicates the most likely position(s) (LOD = 0.0)
##
## '**' indicates very likely positions (LOD > -1.0)
##
## '*' indicates likely positions (LOD > -2.0)
```

9.1 Check for alternative orders

```
ripple_seq(LG2.all, ws = 4, LOD = 3)
```

```
## 27-16-20-4-|-21-...
##   Alternative orders:
##    27 16 20 4 21 ... :    0.00 ( linkage phases: 1 1 2 2 ... )
##    16 27 20 4 21 ... :   -0.02 ( linkage phases: 1 1 2 2 ... )
##
## ...-27-|-16-20-4-21-|-23-... OK
##
## ...-16-|-20-4-21-23-|-9-... OK
##
## ...-20-|-4-21-23-9-|-24-... OK
##
## ...-4-|-21-23-9-24-|-29-... OK
##
## 21-|-23-9-24-29
##   Alternative orders:
##   ... 21 23 9 24 29 :    0.00 ( linkage phases: ... 1 1 2 3 )
##   ... 21 23 9 29 24 :   -2.07 ( linkage phases: ... 1 1 4 3 )
```

- We should do this ***<u>to any of the orders we found</u>***, either using try_seq or order_seq. Here, we choose LG2.all only for didactic purpose. The second argument, ws = 4, means that subsets (windows) of four markers will be permutated sequentially (4! orders for each window), to search for other plausible orders. The LOD argument means that only orders with LOD Score smaller than 3 will be printed.

- The output shows sequences of four numbers, since ws = 4. They will be followed by an OK, if there is no alternative orders with LOD Scores smaller than LOD = 3 in absolute value, or by a list of alternative orders. On the example, just the last sequence showed an alternative order with LOD smaller than LOD=3 (2.06, in absolute value). However, the best order was the previous one (LOD=0.00).

- If there was an alternative order most likely than the original, one should check the difference between these orders (and linkage phases) and change it using, for example, the function drop_marker and seq_try or typing the new order. You can use $seq.num and $seq.phase after the name of these sequences (for example, LG2.all$seq.num and LG2.all$seq.phases) to obtain the original order and linkage phases, make the necessary changes (by copying and paste) and then use the function map to reestimate the genetic map for the new order.

- Here, the function ripple_seq showed that the final order obtained is indeed the best for this linkage group.

10 Printing map

```
LG2.all
```

```
##
## Printing map:
##
## Markers           Position           Parent 1       Parent 2
##
## 27 M27                0.00           b |   | o       a |   | a
## 16 M16                0.00           a |   | a       b |   | o
## 20 M20               18.55           a |   | b       c |   | d
##  4 M4                32.94           a |   | o       o |   | b
## 21 M21               48.51           o |   | o       a |   | b
## 23 M23               54.00           a |   | o       a |   | o
##  9 M9                72.47           a |   | b       a |   | a
## 24 M24              107.12           a |   | b       b |   | a
## 29 M29              134.41           o |   | a       o |   | o
##
## 9 markers            log-likelihood: -576.3973
```

Genetic Map of linkage group 1

1 Defining the group

```
LG1<- make_seq(LGs,1)
```

2 Ordering the markers

```
LG1.ord<- order_seq(LG1,n.init = 6, touchdown = TRUE)

##
## Cross type: outcross
## Using segregation types of the markers to choose initial subset
## WARNING: this operation may take a VERY long time
##
## Comparing 360 orders:
##
  |======================================================================|
100%
##
##
## Running try algorithm
##  6 --> M6  : .......
## 30 --> M30 : ........
##  2 --> M2  : ........
##  5 --> M5  : .........
## 10 --> M10 : .........
## 11 --> M11 : .........
##  3 --> M3  : ..........
##
## LOD threshold = 3
##
## Positioned markers: 12 3 14 2 1 28 26 6 15 11
##
```

```
## Markers not placed on the map: 5 10 30
##
##
##
## Trying to map remaining markers with LOD threshold  2
##  5 --> M5  : ...........
## 10 --> M10 : ............
## 30 --> M30 : ............
##
## LOD threshold = 2
##
## Positioned markers: 12 3 14 2 1 10 28 26 6 15 5 11
##
## Markers not placed on the map: 30
##
##
## Calculating LOD-Scores
## 30 --> M30 : ............
##
##
## Placing remaining marker(s) at most likely position
## 30 --> M30 : ............
##
## Estimating final genetic map using tol = 10E-5.
```

2.1 Printing map

```
LG1.ord

##
## Best sequence found.
## Printing map:
##
## Markers           Position           Parent 1       Parent 2
##
## 12 M12               0.00           a |  | b       a |  | c
##  3 M3               23.31           a |  | o       o |  | o
## 14 M14              38.05           o |  | a       o |  | b
##  2 M2               45.72           o |  | o       a |  | o
##  1 M1               70.77           a |  | b       a |  | b
## 10 M10              77.08           a |  | a       b |  | o
## 28 M28              96.14           o |  | a       o |  | b
## 26 M26             138.33           b |  | a       c |  | d
##  6 M6              148.16           a |  | b       a |  | b
## 15 M15             190.35           a |  | o       b |  | o
##  5 M5              195.62           o |  | o       a |  | o
## 11 M11             219.52           o |  | o       a |  | b
##
## 12 markers            log-likelihood: -886.8574
##
##
```

```
##
## The following markers could not be uniquely positioned.
## Printing most likely positions for each unpositioned marker:
##
## ------------
## |    |  30 |
## |----|-----|
## |    | *** |
## | 12 |     |
## |    | **  |
## |  3 |     |
## |    |     |
## | 14 |     |
## |    |     |
## |  2 |     |
## |    |     |
## |  1 |     |
## |    |     |
## | 10 |     |
## |    |     |
## | 28 |     |
## |    |     |
## | 26 |     |
## |    |     |
## |  6 |     |
## |    |     |
## | 15 |     |
## |    |     |
## |  5 |     |
## |    |     |
## | 11 |     |
## |    |     |
## ------------
##
## '***' indicates the most likely position(s) (LOD = 0.0)
##
## '**' indicates very likely positions (LOD > -1.0)
##
## '*' indicates likely positions (LOD > -2.0)
```

3 Safe map and “forced” map

```
(LG1.safe<- make_seq(LG1.ord, "safe"))

##
## Printing map:
##
## Markers           Position           Parent 1       Parent 2
##
## 12 M12               0.00           a |  | b       a |  | c
##  3 M3               23.31           a |  | o       o |  | o
```

```
## 14 M14                38.05           o |   | a       o |   | b
##  2 M2                 45.72           o |   | o       a |   | o
##  1 M1                 70.77           a |   | b       a |   | b
## 10 M10                77.08           a |   | a       b |   | o
## 28 M28                96.14           o |   | a       o |   | b
## 26 M26               138.33           b |   | a       c |   | d
##  6 M6                148.16           a |   | b       a |   | b
## 15 M15               190.35           a |   | o       b |   | o
##  5 M5                195.62           o |   | o       a |   | o
## 11 M11               219.52           o |   | o       a |   | b
##
## 12 markers            log-likelihood: -886.8574
```

```
(LG1.force<- make_seq(LG1.ord, "force"))
## Printing map:
##
## Markers           Position           Parent 1       Parent 2
##
## 30 M30               0.00           a |   | b       a |   | b
## 12 M12               1.01           b |   | a       c |   | a
##  3 M3               24.32           o |   | a       o |   | o
## 14 M14              39.07           a |   | o       b |   | o
##  2 M2               46.74           o |   | o       o |   | a
##  1 M1               71.78           b |   | a       b |   | a
## 10 M10              78.10           a |   | a       o |   | b
## 28 M28              97.15           a |   | o       b |   | o
## 26 M26             139.35           a |   | b       d |   | c
##  6 M6              149.17           b |   | a       b |   | a
## 15 M15             191.36           o |   | a       o |   | b
##  5 M5              196.63           o |   | o       o |   | a
## 11 M11             220.53           o |   | o       b |   | a
##
## 13 markers            log-likelihood: -896.667
```

4 Check for alternative orders

```
ripple_seq(LG1.force)
```

```
## 30-12-3-14-|-2-...
##   Alternative orders:
##    30 12 3 14 2 ... :    0.00 ( linkage phases: 4 1 4 2 ... )
##    12 30 3 14 2 ... :   -0.20 ( linkage phases: 4 4 4 2 ... )
##
## ...-30-|-12-3-14-2-|-1-... OK
##
## ...-12-|-3-14-2-1-|-10-... OK
##
## ...-3-|-14-2-1-10-|-28-...
##   Alternative orders:
##   ... 3 14 2 1 10 28 ... :    0.00 ( linkage phases: ... 4 2 3 1 4 ...
)
```

```
##   ... 3 14 2 10 1 28 ... :   -2.57 ( linkage phases: ... 4 2 3 1 4 ...
)
##
## ...-14-|-2-1-10-28-|-26-...
##   Alternative orders:
##   ... 14 2 1 10 28 26 ... :    0.00 ( linkage phases: ... 2 3 1 4 2
... )
##   ... 14 2 10 1 28 26 ... :   -2.57 ( linkage phases: ... 2 3 1 4 2
... )
##
## ...-2-|-1-10-28-26-|-6-...
##   Alternative orders:
##   ... 2 1 10 28 26 6 ... :    0.00 ( linkage phases: ... 3 1 4 2 3 ...
)
##   ... 2 10 1 28 26 6 ... :   -2.57 ( linkage phases: ... 3 1 4 2 3 ...
)
##
## ...-1-|-10-28-26-6-|-15-... OK
##
## ...-10-|-28-26-6-15-|-5-... OK
##
## ...-28-|-26-6-15-5-|-11-... OK
##
## 26-|-6-15-5-11 OK
```

Map for a arbitrary order

- If, for any reason, one wants to estimate parameters for a given linkage map (e.g. for other orders on published papers), it is possible to define a sequence and use the map function.

1 Constructing a map with a arbitrary order

1.1 Example, for markers M30, M12, M3, M14 and M2, in this order, use

```
any.seq <- make_seq(twopts,c(30,12,3,14,2))
(any.seq.map <- map(any.seq))
## Printing map:
##
## Markers           Position           Parent 1       Parent 2
##
## 30 M30               0.00           a |  | b       a |  | b
## 12 M12               1.01           b |  | a       c |  | a
##  3 M3               24.32           o |  | a       o |  | o
## 14 M14              39.07           a |  | o       b |  | o
##  2 M2               47.78           o |  | o       o |  | a
##
## 5 markers           log-likelihood: -320.9012
```

- The map function searches for the best combination of phases between markers and print the results. Furthermore, a sequence can also have user-defined linkage phases.

2 Add (incorrect) phases for the same order of markers

```
any.seq <- make_seq(twopts,c(30,12,3,14,2),phase=c(4,1,4,3))
(any.seq.map <- map(any.seq))
## Printing map:
##
## Markers           Position           Parent 1       Parent 2
##
## 30 M30               0.00           a |  | b       a |  | b
## 12 M12               1.01           b |  | a       c |  | a
##  3 M3               24.32           o |  | a       o |  | o
## 14 M14              39.07           a |  | o       b |  | o
##  2 M2              695.18           o |  | o       a |  | o
##
## 5 markers           log-likelihood: -362.3389
```

3 Add and drop markers

- If you need to add or drop markers from a predefined sequence, functions add_marker and drop_marker can be used. For example, to add markers 4 to 8 to any.seq

3.1 Add marker

```
(any.seq <- add_marker(any.seq, 4:8))
## Number of markers: 10
## Markers in the sequence:
## M30 M12 M3 M14 M2 M4 M5 M6 M7 M8
##
## Parameters not estimated.
```

4 Removing markers

4.1 Example: removing 3, 4, 5, 12 and 30 from any.seq:

```
(any.seq <- drop_marker(any.seq, c(3,4,5,12,30)))

##
## Number of markers: 5
## Markers in the sequence:
## M14 M2 M6 M7 M8
##
## Parameters not estimated.
```

5 Plotting the recombination fraction matrix

- For a given sequence, it is possible to plot the recombination fraction matrix and LOD Scores based on a color scale using the function rf.graph.table. This matrix can be useful to make some diagnostics about the map. For example, using the function group with LOD=2.5

5.1 Setting a lower LOD score for grouping

```
(LGs <- group(mark.all, LOD=2.5))

##    Selecting markers:
##    group    1
##     ..............
##    group    2
##     ..............

##   This is an object of class 'group'
##   It was generated from the object "mark.all"
##
##   Criteria used to assign markers to groups:
##     LOD = 2.5 , Maximum recombination fraction = 0.5
##
##   No. markers:            30
##   No. groups:             2
##   No. linked markers:     30
##   No. unlinked markers:   0
##
##   Printing groups:
##   Group 1 : 15 markers
##     M1 M2 M3 M5 M6 M10 M11 M12 M14 M15 M17 M25 M26 M28 M30
##
##   Group 2 : 15 markers
##     M4 M7 M8 M9 M13 M16 M18 M19 M20 M21 M22 M23 M24 M27 M29
```

- Due to the small value used for the LOD Score (2.5, not adequate and resulting in false positives), markers from di↵erent groups were placed together.

5.2 Ordering markers

```
LG.err<-make_seq(LGs, 2)
LG.err.ord<-order_seq(LG.err)

##
## Cross type: outcross
## Using segregation types of the markers to choose initial subset
##
## Comparing 60 orders:

  |======================================================================|
100%
```

```
##
## Running try algorithm
## 24 --> M24 : ......
## 13 --> M13 : ......
## 23 --> M23 : .......
##  9 --> M9  : .......
## 22 --> M22 : .......
## 27 --> M27 : .......
## 29 --> M29 : ........
##  7 --> M7  : ........
## 16 --> M16 : ........
## 21 --> M21 : ........
##
## LOD threshold = 3
##
## Positioned markers: 27 20 4 19 21 18 8 13
##
## Markers not placed on the map: 7 9 16 22 23 24 29
##
##
## Calculating LOD-Scores
##  7 --> M7  : .........
##  9 --> M9  : .........
## 16 --> M16 : .........
## 22 --> M22 : .........
## 23 --> M23 : .........
## 24 --> M24 : .........
## 29 --> M29 : .........
##
## Placing remaining marker(s) at most likely position
## 23 --> M23 : .........
## 24 --> M24 : ..........
##  7 --> M7  : ...........
## 22 --> M22 : ............
## 29 --> M29 : .............
## 16 --> M16 : ..............
##  9 --> M9  : ...............
##
## Estimating final genetic map using tol = 10E-5.
```

5.3 Map using option “force”

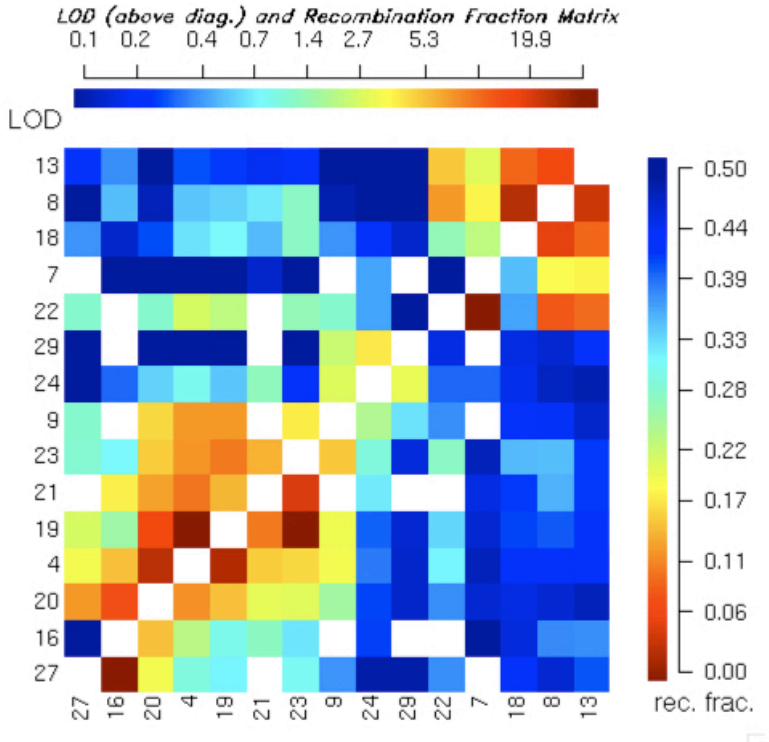
```
(LG.err.map<-make_seq(LG.err.ord, "force"))
## Printing map:
##
## Markers           Position           Parent 1       Parent 2
##
## 27 M27               0.00           b |  | o       a |  | a
## 16 M16               0.00           a |  | a       b |  | o
## 20 M20              18.55           a |  | b       c |  | d
```

```
##  4 M4                  32.94           a |   | o       o |   | b
## 19 M19                 35.27           b |   | a       a |   | o
## 21 M21                 47.58           o |   | o       a |   | b
## 23 M23                 52.66           a |   | o       a |   | o
##  9 M9                  71.64           a |   | b       a |   | a
## 24 M24                106.16           a |   | b       b |   | a
## 29 M29                132.74           o |   | a       o |   | o
## 22 M22                256.21           a |   | b       a |   | a
##  7 M7                 256.21           a |   | a       a |   | b
## 18 M18                318.10           a |   | b       c |   | d
##  8 M8                 324.11           b |   | a       b |   | a
## 13 M13                328.41           a |   | o       a |   | o
##
## 15 markers            log-likelihood: -907.33
```

- A careful examination of the results shows that there are problems on the map. This can be done by plotting the recombination fraction matrix

5.4 Plotting the recombination fraction matrix

```
rf_graph_table(LG.err.map)
```



References

- Adler, J. R in a Nutshell A Desktop Quick Reference, 2009. Broman, K. W., Wu, H., Churchill, G., Sen, S., Yandell, B. qtl: Tools for analyzing QTL experiments R package version 1.09-43, 2008. (http://www.rqtl.org/)
- Buetow, K. H., Chakravarti, A. Multipoint gene mapping using seriation. I. General methods. American Journal of Human Genetics 41, 180-188, 1987.
- Doerge, R.W. Constructing genetic maps by rapid chain delineation. Journal of Agricultural Genomics 2, 1996.
- Garcia, A.A.F., Kido, E.A., Meza, A.N., Souza, H.M.B., Pinto, L.R., Pastina, M.M., Leite, C.S., Silva, J.A.G., Ulian, E.C., Figueira, A. and Souza, A.P. Development of an integrated genetic map of a sugarcane (Saccharum spp.) commercial cross, based on a maximum- likelihood approach for estimation of linkage and linkage phases. Theoretical and Applied Genetics 112, 298-314, 2006.
- Haldane, J. B. S. The combination of linkage values and the calculation of distance between the loci of linked factors. Journal of Genetics 8, 299-309,

1919.

- Jiang, C. and Zeng, Z.-B. Mapping quantitative trait loci with dominant and missing markers in various crosses from two inbred lines. Genetica 101, 47-58, 1997.
- Kosambi, D. D. The estimation of map distance from recombination values. Annuaire of Eu- genetics 12, 172-175, 1944.
- Lander, E. S. and Green, P. Construction of multilocus genetic linkage maps in humans. Proc. Natl. Acad. Sci. USA 84, 2363-2367, 1987.
- Lander, E.S., Green, P., Abrahanson, J., Barlow, A., Daly, M.J., Lincoln, S.E. and Newburg, L. MAPMAKER, An interactive computing package for constructing primary genetic linkage maps of experimental and natural populations. Genomics 1, 174-181, 1987.
- Lincoln, S. E., Daly, M. J. and Lander, E. S. Constructing genetic linkage maps with MAP- MAKER/EXP Version 3.0: a tutorial and reference manual. A Whitehead Institute for Biomedical Research Technical Report 1993.
- Margarido, G. R. A., Souza, A.P. and Garcia, A. A. F. OneMap: software for genetic mapping in outcrossing species. Hereditas 144, 78-79, 2007.
- Mollinari, M., Margarido, G. R. A., Vencovsky, R. and Garcia, A. A. F. Evaluation of algorithms used to order markers on genetics maps. Heredity 103, 494-502, 2009.
- Oliveira, K.M., Pinto, L.R., Marconi, T.G., Margarido, G.R.A., Pastina, M.M., Teixeira, L.H.M., Figueira, A.M., Ulian, E.C., Garcia, A.A.F., Souza, A.P. Functional genetic link- age map on EST-markers for a sugarcane (Saccharum spp.) commercial cross. Molecular Breeding 20, 189-208, 2007.
- Oliveira, E. J., Vieira, M. L. C., Garcia, A. A. F., Munhoz, C. F.,Margarido, G. R.A., Consoli, L., Matta, F. P., Moraes, M. C., Zucchi, M. I., and Fungaro,M. H. P. An Integrated Molec- ular Map of Yellow Passion Fruit Based on Simultaneous Maximum-likelihood Estimation of Linkage and Linkage Phases J. Amer. Soc. Hort. Sci. 133, 35-41, 2008.
- Tan, Y., Fu, Y. A novel method for estimating linkage maps. Genetics 173, 2383-2390, 2006.
- Van Os H, Stam P, Visser R.G.F., Van Eck H.J. RECORD: a novel method for ordering loci on a genetic linkage map. Theor Appl Genet 112, 30-40, 2005.
- Voorrips, R.E. MapChart: software for the graphical presentation of linkage maps and QTLs. Journal of Heredity 93, 77-78, 2002.
- Wang S., Basten, C. J. and Zeng Z.-B. Windows QTL Cartographer 2.5.

Department of Statis- tics, North Carolina State University, Raleigh, NC, 2010. (http://statgen.ncsu.edu/ qtlcart/WQTLCart.htm)

- Wu, R., Ma, C.X., Painter, I. and Zeng, Z.-B. Simultaneous maximum likelihood estimation of linkage and linkage phases in outcrossing species. Theoretical Population Biology 61, 349-363, 2002.