MDSMap:High density linkage maps using multi-dimensional scaling

Katharine F. Preedy, Christine A. Hackett and Bram Boskamp
Biomathematics and Statistics Scotland (BioSS), Invergowrie, DD2 5DA
katharine.preedy@bioss.ac.uk

August 3, 2018

Summary

We present an R-package MDSMap, which contains functions relevant to robust methods for rapid construction of high density linkage maps suitable for a variety of experimental genetic populations from homozygous or heterozygous parents in diploids or autotetraploids. The approach uses multi-dimensional scaling to order markers within a linkage group using pairwise recombination fractions (or mappings thereof) weighted by LOD scores (logarithm_{10} of the odds ratio). Functions are included to allow evaluation adjustment of the estimated order and comparison to some external "truth". The estimation is a two step process using unconstrained SMACOF followed by either spherically constrained SMACOF or fitting a principal curve using the R-packages **smacof** and **princurve**.

Key Words: multidimensional scaling, linkage mapping, principal curves, autotetraploid

1 Introduction

Modern genotyping techniques are producing increasingly high numbers of genetic markers that can be scored in experimental populations of plants and animals. Ordering these markers to form a reliable linkage map is computationally challenging. There is a wide literature on this topic, but much has focused on populations derived from diploid, homozygous parents. Here we present a method which uses weighted multidimensional scaling (MDS) to order markers from more general experimental crosses, with homozygous or heterozygous parents that are diploid or autotetraploid. We demonstrate the method using simulated data and also experimental data from a tetraploid potato population. The method was originally developed for incorporation into TetraploidSNPMap (Hackett et al., 2017) and is discussed in detail in Preedy and Hackett (2016). The general approach is to use pairwise recombination fractions (or Haldane or Kosambi map distances) and their associated

LOD (or LOD^2) scores and map them into a 2 or 3 dimensional Euclidean space using the SMACOF method of weighted metric multidimensional scaling (MDS). They are then mapped onto a curve either by fitting a principal curve (Hastie and Stuetzele, 1989) or by constraining the final configuration of the MDS to lie on an arc.

2 Multidimensional Scaling

Multidimensional scaling (MDS) refers to a class of ordination techniques designed to display 'distances' among points in geometrical space. It is generally used to reduce data from many dimensions, m, to fewer, possibly more comprehensible dimensions, n. If there are m observations then MDS techniques use an $m \times m$ matrix of observed distances (or dissimilarities) between points and the desired number of dimensions, $n < m$, is specified. A configuration of points in n-dimensional space is sought that best preserves the observed distances between points by minimising a loss function L. For a given configuration X, the loss function $L(X)$ is a function of the difference between the observed distances in the m-dimensional configuration (which may be formed using any metric) and the Euclidean distances between points in the n-dimensional configuration

$$L(X) = \sum_{i=1}^{m} \|w_i d_i . \hat{d}_i(X)\| \tag{1}$$

where $\|.\|$ is any metric function (i.e. it satisfies intuitive properties of a distance such as non-negativity, symmetry and the triangle inequality $\|x.y\| + \|y.z\| \geq \|x.z\|$), d_i is the m-dimensional vector of observed distances between point i and the other points, w_i is a vector of weights associated with point i and $\hat{d}_i(X)$ is the m-dimensional vector of distances between point i and the other points in configuration X. In its simplest form classical multidimensional scaling is also known as principal co-ordinates analysis and, though the distance matrix may be calculated in a variety of ways, the metric is always Euclidean, $\|d_i.\hat{d}_i\| = \sqrt{\sum_j \left(d_{ij} - \hat{d}_{ij}\right)^2}$ and the weights are always equal to one. If the distance matrix is Euclidean, then this is equivalent to principal components analysis and the function to be minimised reduces to $\sqrt{\sum_{ij} \left(d_{ij}^2 - \hat{d}_{ij}^2\right)}$. Metric multidimensional scaling (or weighted metric multidimensional scaling) generalises classical multidimensional scaling to allow for different metrics (and weights) and a commonly used loss function in this context is stress, defined as

$$\sigma(X) = \sum_{i<j<m} w_{ij} \left(dij - \hat{d}_{ij}(X)\right)^2 \tag{2}$$

There are many ways of minimising $\sigma(X)$ and we use a common method, the stress minimisation by majorization approach implemented in the **smacof** package. This minimises $\sigma(X)$ iteratively by minimising at each step a simple function that bounds σ from above, called the majorizing function. The method is described in detail in de Leeuw and Mair (2009).

The analysis described above is an unconstrained MDS. It is also possible to constrain the final configuration of points to lie on a circle by imposing a penalty for deviations from that circle, in a constrained MDS. This is done by defining a new point in the centre of the data and constraining all points to be equidistant from it. The variation in distance from the centre point is added to the stress function.

3 Principal Curves

Formally, principal curves (PC) were defined by Hastie and Stuetzle (1989) as self-consistent smooth one-dimensional curves that pass through the middle of a p-dimensional data set providing a nonlinear summary of the data. (In this context, the projection of a data point onto a curve is the closest point on that curve, and for a curve to be self-consistent, any set of data points that project onto the same point, z, on the curve must have point z as their mean.) Fitting a PC is an iterative two-stage process. A summary straight line, such as a principal component, is fitted. Then this summary line is transformed to a smooth curve, using splines, to achieve self-consistency. Since splines depend on the smoothing constraint, PCs are not unique. We used the algorithm implemented in **princurve** (Hastie and Weingessel, 2018) which uses the first principal component (from a Principal Components Analysis) as the initial summary of the data, cubic splines for fitting smooth curves and local averaging to determine self-consistency. The smoothing constraint can be selected by an explicit option or determined automatically by leave-one-out cross validation.

4 The Basic Algorithm

We take an input of pairwise recombination fractions and LOD scores for the population of interest and cast the data into distance or LOD score matrices. If a map distance is used recombination fractions, r, are converted to map distances, d, using either Haldane's mapping function

$$d_h = -\frac{1}{2}\ln(1-2r) \tag{3}$$

or the Kosambi mapping function

$$d_k = \frac{1}{4}\ln\left(\frac{1+2r}{1-2r}\right) \tag{4}$$

For mapping using principal curves the algorithm is as follows:

1. Use the smacofSym function from the the **smacof** package to perform two or three dimensional weighted unconstrained MDS on the distance matrix.

2. Plot final configuration to find potential outliers from Smacofsym.

3. Fit the principal curves using package **princurve**.

4. Plot the first principal curve on the final configuration of the unconstrained fit and assess whether it looks reasonable.

5. The projections of the markers onto the first principal curve give the estimated map positions.

For mapping using constrained MDS follow steps 1-2 as for two dimensional principal curves

4. Use the smacofSphere function in two dimensions to constrain the points to approximate to the arc of a circle with a penalty, p, for deviations from the arc.

5. Plot the final configuration from smacofSym and smacofSphere to check for any points which have major changes in rank with respect to either dimension in the final configuration.

6. Check the stress ratio smacofsphere stress/smacofsym stress. This is a metric for the increase in stress (which approximates to a measure of the reduction in fit) caused by forcing the points to lie on an arc and should be below 1.1. If the ratio is above this, return to step 4 and reduce the penalty p.

7. Project the final configuration onto a line to get order and estimated map length.

   (a) Centre sphere on (0,0)

   (b) Calculate the polar coordinates of each point in the configuration.

   (c) Rotate so that the mapping starts at the beginning of the arc.

   (d) Radius of the sphere is the median distance of points from (0,0) rescaled so that the sum of the configuration is the same as the sum of the observed distances.

   (e) Order the markers by increasing angle.

   (f) Inter-marker distances are equal to the radius multiplied by the difference in angle between the points.

In both cases the fit of individual points can be assessed via the nearest neighbour measure (nnfit) derived from the matrix of distances. This is a measure given for each marker and is the sum of the absolute difference between the observed and estimated distance between that marker and the nearest informative neighbours on either side that is the nearest neighbours with a non-zero LOD score. (Neighbouring markers where different parents are heterozygous are uninformative about recombination). For some markers near the ends of the chromosome there will be a neighbour on only one side. High values of the criterion can be used to identify possible outliers. The mean NNfit provides a measure of the fit to the complete set of data. It can be used to compare models with different weight functions (LOD or LOD^2) and in different numbers of dimensions when using the Principal Curves method as long as the same data and same distance metric are used.

5 Examples

5.1 Backcross from Homozygous Parents

The package contains a wrapper function for the **qtl** package (Broman et al., 2003) to simulate a backcross population and write it to a file in the form that would have been output by JoinMap 4 (Van Ooijen, 2006), namely with the number of markers and the number of pairs on the first line, then the marker names and pairwise recombination fractions and associated LOD scores below. This example uses a simulated population of 200 individuals with 200 markers on a chromosome of length 100cM with a 1% random error rate.

```
fname<-'bcsim'
sim.bc.rflod.file(fname)
```

For the purposes of this example both the spherically constrained method and the method of principal curves are employed to estimate the map using the default settings of a Haldane map function and a LOD^2 weighting.

5.1.1 Spherically constrained estimation

```
map.s<-estimate.map(fname,p=100,n=NULL,ispc=FALSE,displaytext=FALSE)
```

Diagnostic plots are automatically generated. (See Figure 1 for the output from this example, but note that every simulation would be different). In general, where a lot of markers are involved it is easier to plot numbers than marker names. However, this can be altered by setting `displaytext=TRUE`. The marker name associated with each number can be accessed using `map$locikey`. The final estimated map is stored in `map.s$locimap$`. The higher the value of p the higher the penalty for deviations from the sphere. A good rule of thumb is that if the ratio of the stress from the spherically constrained smacof to the unconstrained smacof is > 1.1 p should be reduced to avoid overly distorting the configuration of the markers. In the case below the ratio is 1.0032, well within that margin, so p does not need to be altered. The bottom left plot shows the final configuration from the unconstrained smacof and can be used to check for any major outliers - in this case markers 105 and 102 stand out slightly but not enough to consider removing them. If we were concerned about them we could remove them by setting `n=c(102,105)` and rerunning `estimate.map`. Once satisfied that there are no major outliers in the unconstrained configuration the next step is to consider the plot which contains both the unconstrained (black numbers) and spherically constrained configurations (red numbers) and the plot of the nearest neighbour fits. The purpose of the former plot is to check for markers significantly changing rank in either dimension 1 or 2 as this may indicated that part of the map has been inverted. In this case, although there are slight changes (for instance markers 28 and 102), none of them are large enough to warrant concern and the final plot indicates that the nearest neighbour fit discrepancies are not large for the points of concern. If deciding on whether changing p improves the fit, a reduction in the mean nearest neighbour fit (or

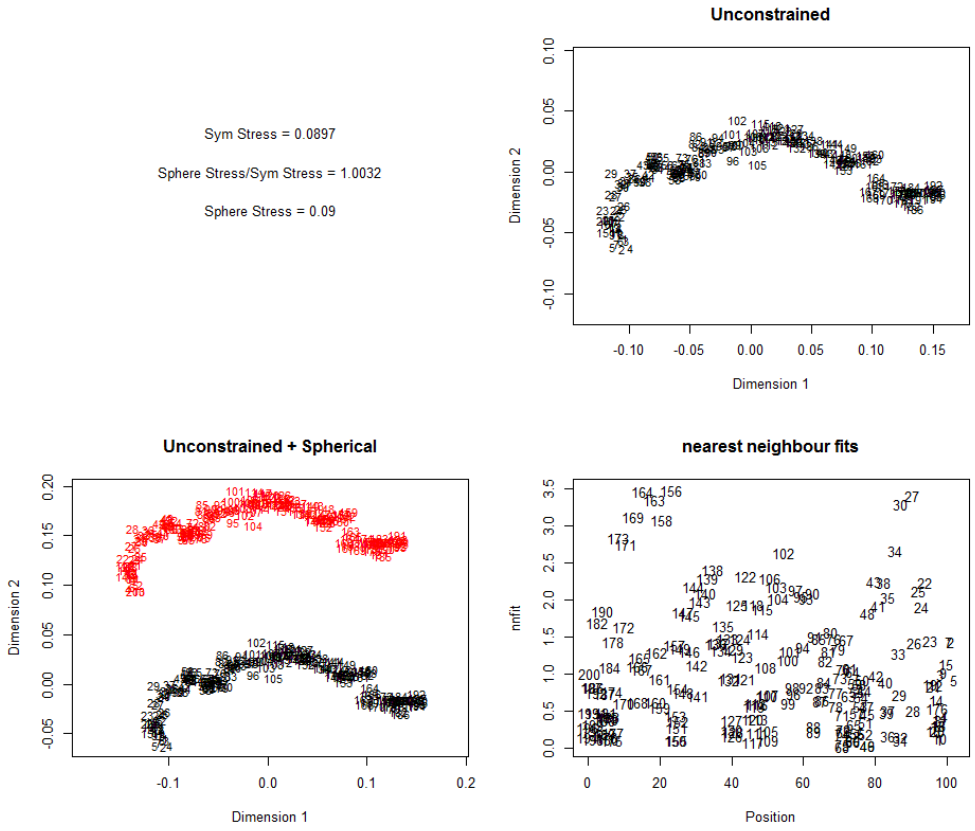

Figure 1: Diagnostic plots from a spherically constrained map estimation of a simulated backcross population

`map.s$totalnnfit`) would indicate a closer representation of the observed pairwise inter-marker distances. In this case the fit is satisfactory so no further modifications are needed and the final map can be accessed using `map.s$locimap`. In this situation, the markers were, in fact, presented in order, so the number on the configuration plot gives the true order of the markers and the quality of the fit can be assessed by plotting that against the estimated marker position.

Figure 2 shows that in this case, the order has been inverted, but is broadly accurate. If it is desired to do so the map can be inverted using `invertmap(map.s$locimap)`.

5.1.2 Principal Curves Method

In general the principal curves method is faster, more robust and nearly as accurate as the spherically constrained method so is recommended as the default approach. The command for estimating the map using principal curves is similar to that for the spherically constrained approach. However, the principal curves method is selected byusing the default

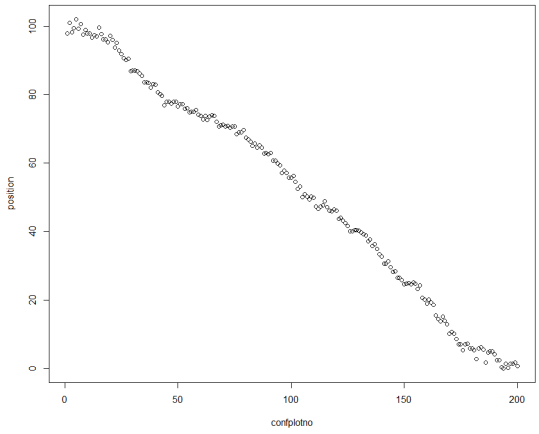

Figure 2: Diagnostic plots from a spherically constrained map estimation of a simulated backcross population

for the `ispc` argument, `ispc=TRUE`. In this case `p` refers to the smoothing parameter which may vary between 0 and 1. If, as is the default, it is left NULL, then the smoothing parameter is selected using leave-one-out cross validation. For simple backcross populations a 2-dimensional fit is generally sufficient.

```
map<-estimate.map(fname,p=NULL,n=NULL,ispc=TRUE,ndim=2,displaytext=FALSE)
```

Figure 3 shows the diagnostic plots for the map estimate. As before, numbers have been plotted rather than marker names, and the names associated with each number are stored in `map.pc$locikey`. The final estimated map is stored in `map.pc$locimap`. The first plot shows the final configuration from the unconstrained MDS together with the first principal curve and the second plot shows the nearest neighbour fits for each marker plotted by its estimated position. There are no major outliers. We have used exactly the same input data and markers (not having dropped any from either estimation), the same LOD weights and map distance function so it is reasonable to compare the total nearest neighbour fits between the estimated maps. The mean nearest neighbour fit for this map is 1.007 (accessed using `map.pc$meannnfit`) which is lower than the total nearest neighbour fit for the spherically constrained map estimate (1.021) suggesting that the method of principal curves remains closer to the original data.

5.2 Full-sib Population from Autotetraploid Heterozygous Parents

The data used in this example comes from the potato Stirling x 12601ab1 mapping population described in Hackett et al. (2013). The pairwise recombination fractions and LOD scores for linkage group I are stored in the file 'lgI.txt'. The two-dimensional fit is slightly

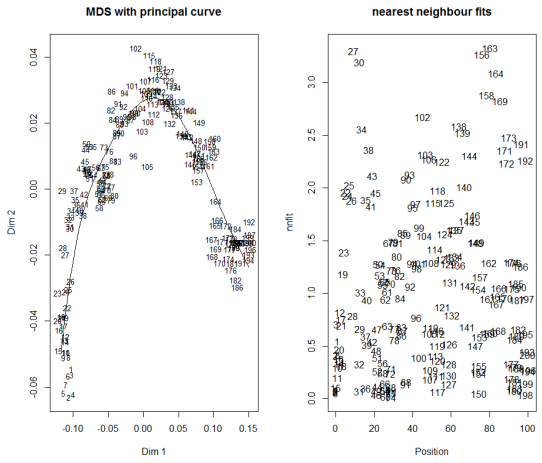

Figure 3: Diagnostic plots from principal curves map estimation of a simulated backcross population

faster and the diagnostic plot easier to interpret. However, tetraploid data can sometimes be represented better in 3 dimensions so it makes sense to fit both and compare which represent the data better.

```
fname<-system.file("extdata", "lgI.txt", package="MDSMap")
map2d<-estimate.map(fname,p=NULL,n=NULL,ispc=TRUE,ndim=2,displaytext=FALSE)
map3d<-estimate.map(fname,p=NULL,n=NULL,ispc=TRUE,ndim=3,displaytext=FALSE)
```

Note that, in addition to the diagnostic plots, the 3-dimensional fit launches a 3-d graph which can be rotated using the mouse to explore for outliers. In this example the total nearest neighbour fit is 137.75 for the 2-dimensional fit, and 139.65 for the 3-dimensional fit, indicating that the 2-dimensional fit gives a better representation of the observed data with diagnostic plots displayed in Figure 4. The plots suggest a bit of a gap between marker number 49 and the other markers, and also a relatively high discrepancy between the observed and fitted difference between that marker and its nearest informative neighbour so there may be some uncertainty as to the exact distance between markers 49 and 8. `with(map2d,locikey[locikey$confplotno==49,])` reveals this to be locus c2_9722. However, in generally the fit appears reasonable and the map can be accessed using `map2d$locimap`. A plot of the fitted map (generated using the commands below) is displayed in Figure 5.

```
with(map2d,plot(locimap$position,locimap$position, pch="",xlab="position",ylab="position"))
with(map2d, text(locimap$position,locimap$position, locimap$confplotno,cex=0.8))
```

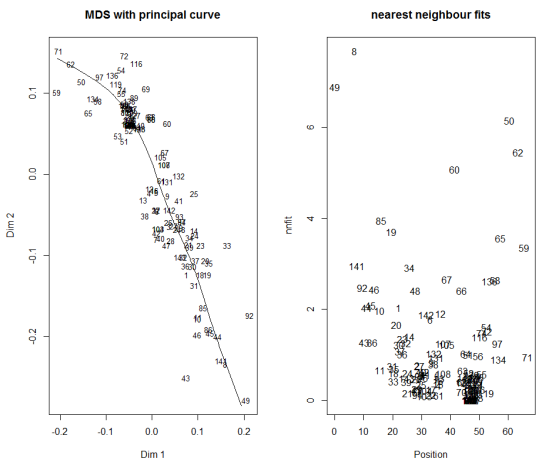

Figure 4: Diagnostic plots from principal curves map estimation from the data from the potato Stirling x 12601ab1 mapping population

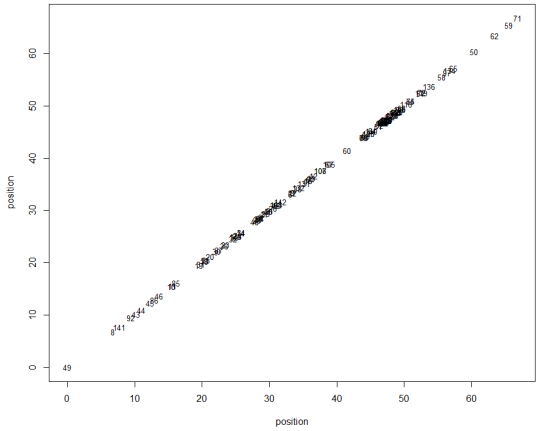

Figure 5: Diagnostic plots from principal curves map estimation from the data from the potato Stirling x 12601ab1 mapping population

6 Advanced functions and integration into other software

In the case of a simulation, the true positions of the markers is known and in cases where simulations are to be used to decide which weight function (LOD or LOD^2) to use or which mapping function (none, Haldane or Kosambi). The function `mean.distance.from.truth` can be used to compare the estimated with the 'real' map. The default is to use LOD^2 weights and a haldane map. Finally, the nearest neighbour fits can be calculated from a file

if the map is refined using alternative software using the function `calc.nnfit.from.file`.

The authors gratefully acknowledge the helpful feedback from Peter Bourke at Wageningen University which has considerably improved the operation of this package.

References

Broman KW, Wu H, Sen S, Churchill GA (2003) R/qtl: QTL mapping in experimental crosses. *Bioinformatics* **19** 889-890

de Leeuw J, Mair P (2009) Multidimensional scaling using majorization: SMACOF in R. *Journal of Statistical Software* **31** 1-30

Hackett CA, McLean K, Bryan GJ (2013) Linkage analysis and QTL mapping using SNP dosage data in a tetraploid potato mapping population. *PLoS One*, **8**:e63939

Hackett CA, Boskamp B, Vogogias A, Preedy KF, Milne I (2017) TetraploidSNPMap: software for linkage analysis and QTL mapping in autotetraploid populations using SNP dosage data. *Journal of Heredity* **108** 438-442.

Hastie T, Stuetzle W (1989) Principal Curves. *Journal of the American Statistical Associaton* **84** 502-516

Hastie T, Weingessel A, Hornik K, Bengtsson H, Cannoodt R (2018) R/princurve: Fits a principal curve in arbitrary dimension. R package version 2.1.2. http://CRAN.R-project.org/package=princurve

Preedy KF and Hackett CA (2016) A rapid marker ordering approach for high-density genetic linkage maps in experimental autotetraploid populations using multidimensional scaling. *Theoretical and Applied Genetics* **129** 2117-2132

Van Ooijen JW (2006) JoinMap 4; Software for the calculation of genetic linkage maps in experimental populations. Wageningen; Netherlands: Kyazma B.V