Improvements to the dlmap package

B. Emma Huang and Andrew W. George

January 11, 2010

1 Upgrades

Since the initial release of the `dlmap` package (version 1.0) we have made improvements in a number of areas. We have generalized the types of models and populations that can be analyzed, as well as streamlining the data input and result visualization processes. Hence we have changed the original vignette to reflect the new and improved package. This vignette supplements the help documentation in providing lengthier examples of how to use functions in the package.

Specific improvements to the package include:

- streamlined construction of input `dlcross` object
- ability to analyze backcrosses, doubled haploids, RILs, F2 intercrosses
- ability to analyze association mapping populations
- simple `plot` and `summary` functions to visualize output

2 Introduction

The `dlmap` package represents the implementation of the DLMapping algorithm as described in [3]. DLMapping is a novel method of QTL mapping in a mixed model framework with separate detection and localization stages. The following vignette documents its usage through examples based on the datasets included in the package.

The mixed model framework of the algorithm requires supplementary packages for model fitting. Two such packages are supported through different versions of the package functions. The `asreml` functions are faster and more capable of handling complex models, but require a license for ASReml.

The other functions make use of the freely available R library `nlme`. Some familiarity with one of the packages is recommended in order to use the `dlmap` package. We demonstrate below the usage of both `asreml` and `lme` functions to perform DLMapping.

In the following sections, we present the steps required to perform a sample QTL analysis. We first sketch the DLMapping algorithm for those who are unfamiliar with its structure. Second, we describe the format of input files for `dlmap`. We then step through two examples using the datasets included in the package. In the first example, there is a single phenotypic observation for each genotype, and we compare the performance of three functions: composite interval mapping (CIM), and DLMapping using each of the mixed model packages (ASReml and nlme). Performing the analysis, interpreting the log files and output, and plotting results are all demonstrated in this section. In the second example, there is more than one phenotypic observation per genotype, and in this case only DLMapping using ASReml is applicable.

The examples presented here do not cover every possible usage of the library functions, but clarify their basic implementation. Further detail can be found in the online help files for the package.

3 Methods

We begin by providing readers with an overview of our QTL mapping strategy. A more detailed exposition is given in [3]. Our algorithm consists of two parts: a detection stage and a localization stage. Both stages are iterative and formulated within a mixed linear model framework.

Detection Stage

- *Step D1: Specify mixed linear models.* A full model and a reduced (or nested) model for each chromosome under investigation are constructed. These models contain fixed and random marker effects to simultaneously account for the extraneous effects of detected and undetected QTL, respectively.

- *Step D2: Identify chromosomes containing undetected QTL.* A likelihood based test statistic is calculated for each chromosome under investigation. This test statistic measures the strength of evidence for the presence of undetected QTL on a chromosome. The genome wide significance of the test statistic is determined via permutation.

- *Step D3: Identify markers to treat as fixed effects.* For each chromosome found to contain significant evidence for undetected QTL in the

previous step, the following procedure is performed. First, we construct a linear mixed model for each marker on the chromosome. The marker is treated as a fixed effect. Secondly, we calculate a Wald statistic for the fixed marker effect. Thirdly, we identify the marker with the largest Wald statistic on a chromosome. This marker is most strongly associated with the QTL and is incorporated into subsequent models as a fixed marker effect.

These three steps are repeated until chromosomes no longer contain detectable QTL. Upon completion of the detection stage, r_j QTL have been detected on chromosome j. We then perform r_j interval mapping scans on chromosome j to localize these QTL.

Localization Stage
Perform interval mapping scan on a chromosome containing unmapped QTL. Firstly, we compute the expected genotype of a QTL conditional on its hypothesized position and the genotypes of the flanking markers. Secondly, we construct a linear mixed model for each hypothesized position. This model, analogous to the models used in the detection stage, contains fixed and random effects to account for the confounding effects on localization of mapped and unmapped QTL, respectively. We also include a fixed effect for the QTL size in the model, formed from the expected QTL genotypes. Thirdly, we calculate the Wald statistic of the QTL effect. The hypothesized QTL position yielding the mixed model with the highest Wald statistic is the estimated location of the QTL.

The QTL size for this position is included as a fixed effect in subsequent scans. These steps are repeated for each detected QTL on a chromosome. Once the detected QTL have been iteratively positioned, we construct a final multiple regression model to accurately estimate the sizes of the QTL.

4 dlcross object

The `dlmap` fitting procedure requires as input an object of class `dlcross` which contains the data frame to be used in model fitting. This object is created from genotype, phenotype and map data by using the constructor function `dlcross`. There are multiple input options, including compatibility with R/qtl `cross` objects, and all of the formats supported by the function `read.cross`. Two new formats for data with phenotypic replicates or association mapping populations are described in the following subsections.

Functions exist to easily print, summarize and plot the `dlcross` object:

```
> library(dlmap)
> data(BSdat)
> dl.in1 <- dlcross(format = "rqtl", genobj = BSdat, idname = "ID")
> summary(dl.in1)

This is an object of class dlcross.
Summary of genetic and phenotypic data:

         This is a bc population.

         No. individuals:          250

         No. phenotypic traits:    2
         Percent phenotyped:       100 100

         No. chromosomes:          9

         Total markers:            99
         No. markers:              11 11 11 11 11 11 11 11 11
         Percent genotyped:        100

There are 250 unique genotypes and 250 unique phenotyped individuals.
```

4.1 format="dlmap"

In order to accommodate datasets with extensive phenotypic data, we have created a novel "dlmap" format for input. This requires three files, representing genotypic, phenotypic, and marker map data. The files can be simple text; or, if the data has already been read into R, objects can be input in place of the files. The need for this format arises because `dlmap`, in contrast to many other packages, can handle complex environmental and genetic relationships simultaneously. Hence data may be observed which has multiple phenotypic observations for each genotype.

Suppose there are $n.gen$ genotyped individuals, $n.ind$ phenotyped individuals, $n.obs$ phenotypic observations (per trait), and M markers in the data. In general, $n.gen \leq n.ind \leq n.obs$ since there may be multiple observations per individual, and more individuals may be phenotyped than genotyped. For example, control individuals whose genotypes are not of interest may be included in the field design in a plant study. Individuals which are genotyped but not phenotyped will not be considered in the analysis. A description of

```
> plot(dl.in1)
```

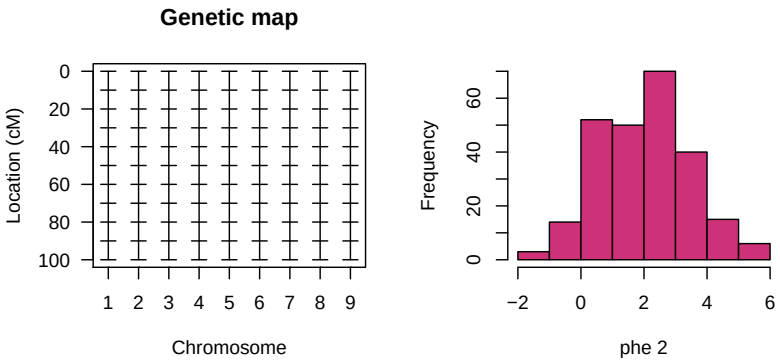

Figure 1: Plot summary of dlcross object, including genetic map and first few phenotypes

each file follows, along with the first few rows and columns of example files. The format for each file is also outlined in the online help files.

Genotype File
The columns in the genotype data file represent a unique identifier for each genotype and the genotype at each marker. The first row must be a header which contains the name of the unique identifier, followed by the marker names. The next *n.gen* rows contain the values for each genotyped individual. Entries can be space or tab delimited. Missing values should be coded as NA. Genotypes should take values from AA, AB, and BB or 0, 1, 2.

ID	D1M1	D1M2	D1M3	D1M4
S1	0	0	0	0
S2	0	0	0	0
S3	1	1	0	0
S4	0	0	0	0
S5	0	0	0	0

Phenotype File
The columns in the phenotype data file represent a unique identifier for each individual and any non-genotypic variables. The first row must be a header which contains the name of the unique identifier, followed by the variable names. The identifier name **must** match the name given in the genotype file. The next *n.obs* rows contain the values for each phenotypic observation. Entries can be space or tab delimited; missing values should be coded as NA.

ID	phenotype
S1	2.084419
S2	2.076666
S3	2.740571
S4	2.373890
S5	2.382941

Map File
The map data file contains either two or three columns. There must be a header row, but the column names are up to the user. The first column must contain the marker names in map order. This should be the same as the marker columns in the genotype file. The second column indicates on which chromosome each marker can be found. The third (optional) column indicates the position of the marker on the chromosome (in cM). If this column is omitted, the marker positions will be estimated from the data. Entries can be space or tab delimited. There should not be any missing data.

MrkID	Chr	Pos
D1M1	1	0
D1M2	1	10
D1M3	1	20
D1M4	1	30
D1M5	1	40

4.2 format="other"

In this version of `dlmap`, we introduce the ability to analyze data from an association mapping population. The data for this format should be input as for the "dlmap" format. However, this format does not require a full genetic map for the analysis. Thus the map file need only contain two columns - one for the marker names, and one for their chromosome assignments. Marker positions can be omitted. Use of this format will also set options for input to the mapping procedure, e.g., positioning QTL at markers rather than performing interval mapping scans.

4.3 Additional Comments

1. Data can be entered as a simple text file. Variables containing character values will be read in as factors, while numeric values will be read in as numeric. Hence, the safest way to ensure that factor variables are not treated improperly is to code all factor values as alphanumerics. An example of this might be a variable representing the plot index of a field trial. While a natural coding is to use the numbers 1, 2, etc., this would be read in as a numeric variable. Instead the variable should be coded as P1, P2, etc. If this is not done, the user **must** make sure to use proper asreml syntax when fitting the model to treat variables as factors. (e.g. dev() command)

2. The name of the unique identifier variable used in both the genotypic and phenotypic data files must be the same. This variable is used to merge the data together.

5 Example 1: Single phenotype per genotype

The `dlmap` package contains multiple datasets with marker and phenotype data. We will examine these in a simple QTL mapping analysis.

`BSdat` is marker data from a simulated backcross. The data has class `cross` and a summary is displayed by typing the object's name. Thus we

can see that it contains nine chromosomes, each with 11 markers genotyped on 250 progeny. There are two phenotypes in the object - an identifier for each genotype and a single trait.

```
> data(BSdat)
> BSdat

  This is an object of class "cross".
  It is too complex to print, so we provide just this summary.
    Backcross

    No. individuals:    250

    No. phenotypes:     2
    Percent phenotyped: 100 100

    No. chromosomes:    9
        Autosomes:      1 2 3 4 5 6 7 8 9

    Total markers:      99
    No. markers:        11 11 11 11 11 11 11 11 11
    Percent genotyped:  100
    Genotypes (%):      AA:48.6  AB:51.4
```

The data was generated using the map included in the object, which has markers evenly spaced at intervals of 10 cM on each chromosome. If we estimate the map from the data, however, the markers will no longer be evenly spaced, as displayed in Figure 2. Either the included or estimated map can be used in the `dlmap` analysis by altering the value of the argument `estmap`.

As described in the documentation for the dataset, the data were generated with seven true QTL, two in coupling on chromosome 1, two in repulsion on chromosome 2, and one on each of chromosomes 3, 4, and 5. These QTL are positioned at 30 and 70 cM for the first two chromosomes, and at 0, 20 and 40 cM for the other three respectively. All QTL have additive effects of magnitude 0.76.

5.1 Standard analysis with Composite Interval Mapping

Composite Interval Mapping (CIM) is a popular QTL mapping method which has been implemented in such programs as QTLCartographer [4] and the

```
> BSmod <- replace.map(BSdat, est.map(BSdat))
> plot.map(BSmod)
```

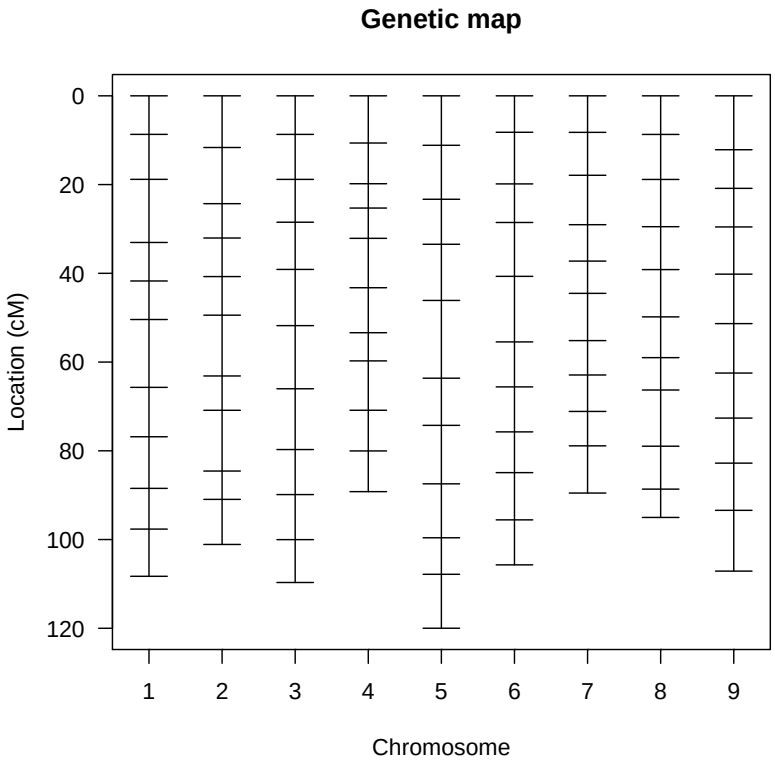

Figure 2: Linkage map estimated from BSdat data

qtl package [2]. The dataset BSdat was generated according to a simulation from [1] where the intent was to test the performance of CIM using different numbers of cofactors. We run the analysis using R/qtl with five marker cofactors.

```
> gp <- calc.genoprob(BSdat, step = 2)
> BScim <- cim(gp, n.marcov = 5)
```

```
> plot(BScim)
> abline(h = 3.56)
```

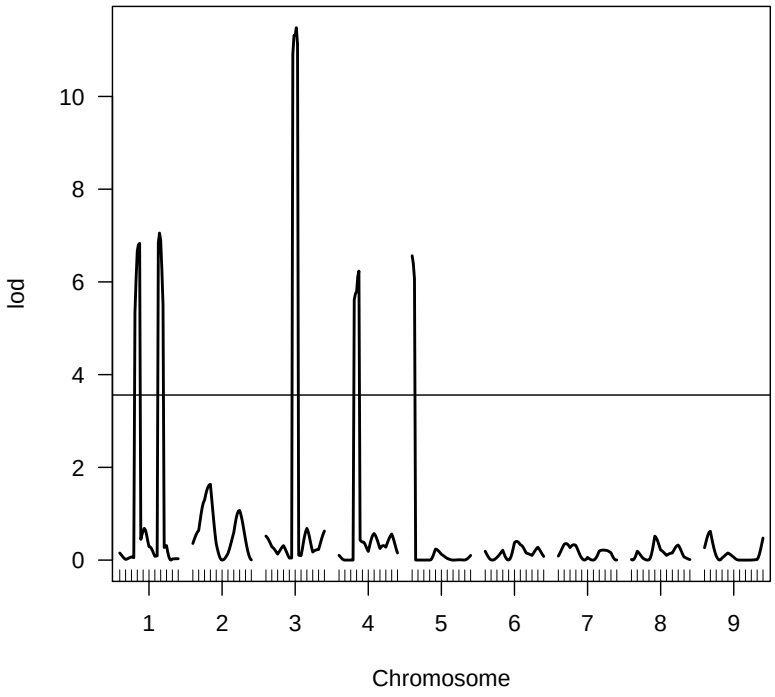

Figure 3: LOD profile for CIM analysis of BSdat with 5 cofactors

This produces a LOD profile at steps of 2 cM along the genome, which is plotted in Figure 3. The horizontal line indicates the threshold for significance of QTL. This genomewide threshold was derived in [1] from 50,000 simulations under the null hypothesis of no QTL. We can see from this plot

that with five cofactors, CIM misses the two QTL on Chromosome 2 which are in repulsion, but detects all the others. Even using seven cofactors, which corresponds to the correct number of QTL, the maximum LOD score on this chromosome (3.5) falls below the significance threshold of 3.77. Thus we would like to use `dlmap` to (hopefully) identify all seven QTL.

5.2 DLMapping with lme

The `dlmap.lme` function is more restricted in its capabilities than `dlmap.asreml`. For example, it can only handle up to 200 markers in a dataset, cannot incorporate additional random effects or covariance structure, and cannot handle multiple phenotypic observations with genotypic data (i.e. replicates of genotypes as are typical of plant studies). Assuming files have been created with default names, then we run the analysis with

```
> system.time(BSlme <- dlmap.lme(object=dl.in1, phename="phenotype",
+ filestem="BS"))

   user  system elapsed
 160.97    0.02  161.34

> names(BSlme)

[1] "input"       "no.qtl"      "final.model" "profile"     "Summary"

> summary(BSlme)
 Summary of input data:

This is an object of class dlcross.
Summary of genetic and phenotypic data:

         This is a bc population.

         No. individuals:          250

         No. phenotypic traits:    2
         Percent phenotyped:       100 100

         No. chromosomes:          9

         Total markers:            99
```

```
        No. markers:              11 11 11 11 11 11 11 11 11
        Percent genotyped:        100

There are 250 unique genotypes and 250 unique phenotyped individuals.

 Summary of final results:
     Chr   Pos Left Marker Right Marker  Effect     SD Z-value p-value
C1M5   1 41.73        D1M4         D1M6 -0.8445 0.1571   -5.38       0
C1M8   1 76.84        D1M7         D1M9 -0.7737 0.1551   -4.99       0
C2M4   2 32.04        D2M3         D2M5 -0.5824 0.1386    -4.2       0
C2M9   2 84.56        D2M8         D2M9  0.5205 0.1395    3.73  0.0002
C3M6   3 51.79        D3M5         D3M7 -0.9032  0.128   -7.06       0
C4M4   4 25.29        D4M3         D4M5 -0.7359 0.1289   -5.71       0
C5M1   5     0        D5M1         D5M2 -0.6975 0.1276   -5.47       0
```

The output is a `dlmap` object with 5 components. There are print, summary, and plot commands to visualize the results graphically and numerically. These are discussed further below.

- `input`: the original `dlcross` input object
- `no.qtl`: the total number of QTL detected
- `final.model`: the output after fitting all the QTL in a multiple regression
- `profile`: a list with components for each chromosome where QTL are detected. Each component is a matrix with two rows. The first row contains the positions for the localization scan of that chromosome (determined by the arguments `step` and `fixpos`), while the second row contains the Wald statistic for the given position. QTL are located based on the size of the Wald statistic, so plotting the profile will show a profile similar to the LOD profile from CIM. Can be plotted using the function `profileplot`
- `Summary`: information about the detected QTL. It is a data frame with seven columns:
  - Column 1 indicates the chromosome of each detected QTL
  - Column 2 indicates the position in cM of the QTL
  - Column 3 indicates the name of the marker flanking the QTL on the left

- Column 4 indicates the name of the marker flanking the QTL on the right
- Column 5 indicates the additive (dominant) effect estimates for the QTL
- Column 6 indicates the Z-ratio for the size estimate in the multiple regression model
- Column 7 indicates the p-value for the Z-ratio

Thus we see that all seven QTL are detected with DLMapping, even with the use of the conservative Bonferroni correction. The position estimates are rough since we scanned for QTL only at markers rather than at intermarker positions. In order to perform the same grid search as CIM, we would add the argument `step=2`. Alternately, we can search a specified number of evenly spaced positions between markers by setting the argument `fixpos`.

5.3 DLMapping with asreml

The same analysis can also be run using `asreml` to fit the mixed models instead of `lme`. The results are identical; one difference in the output is that the `final.model` component of the output list is the output from fitting an `asreml` model rather than from a multiple linear regression.

```
> system.time(BSas <- dlmap.asreml(object=dl.in1, phename="phenotype",
+ filestem="BS"))

   user  system elapsed
  45.97    1.53   47.56

> names(BSas)

[1] "input"       "no.qtl"      "final.model" "profile"     "Summary"

> summary(BSas)

 Summary of input data:

This is an object of class dlcross.
Summary of genetic and phenotypic data:

         This is a bc population.
```

```
        No. individuals:          250

        No. phenotypic traits:    2
        Percent phenotyped:       100 100

        No. chromosomes:          9

        Total markers:            99
        No. markers:              11 11 11 11 11 11 11 11 11
        Percent genotyped:        100

There are 250 unique genotypes and 250 unique phenotyped individuals.

 Summary of final results:
     Chr   Pos Left Marker Right Marker Effect    SD Z-value p-value
C1M5   1 41.73        D1M4         D1M6 -0.845 0.157   -5.38       0
C1M8   1 76.84        D1M7         D1M9 -0.774 0.155   -4.99       0
C2M4   2 32.04        D2M3         D2M5 -0.582 0.139   -4.19       0
C2M9   2 84.56        D2M8         D2M9  0.521  0.14    3.72  0.0002
C3M6   3 51.79        D3M5         D3M7 -0.903 0.128   -7.05       0
C4M4   4 25.29        D4M3         D4M5 -0.736 0.129   -5.71       0
C5M1   5     0        D5M1         D5M2 -0.697 0.128   -5.45       0
```

There are many more options available in the `asreml` implementation of DLMapping. In addition to the `step` and `fixpos` options for specifying the grid search to localize QTL, we can set the number of permutations to perform and fit much more complicated models for phenotypic variation. The default number of permutations is 0, in which case p-values are adjusted with the Bonferroni correction. Permutation testing is not implemented for `dlmap.lme` due to the time requirements. Fitting the same model using `dlmap.lme` and `dlmap.asreml` takes 268 and 41 seconds, respectively.

5.4 DLMapping Log Files

In the process of performing the DLMapping analysis, two log files will be created. The names of these files can be specified with the argument `filestem`, which has a default value of "dl". The two files will then be created in the working directory with names "filestem.trace" and "filestem.det.log". If the option to run permutations is selected, there will also be files created containing all of the permutation test statistics for each iteration of the detection

stage. These files will have the extension ".permX" where X denotes the given iteration.

The trace file is created in order to port all of the output from `asreml` model fitting to a separate file. For each model that is fit, `asreml` outputs the convergence process and various licensing information which can obscure other, more important messages. In the trace file, this output is labelled by whether the models are fit for testing, for marker scans in the detection stage, or for interval mapping scans in the localization stage. However, for the most part this output will not provide much additional useful information.

Note: The trace file will not be created with `dlmap.lme` because `lme` does not output the same information to the screen.

The detection log (.det.log) file provides some additional information about the detection stage. For each iteration of the detection stage, it contains the likelihood ratio test statistics for each chromosome, along with adjusted p-values. The p-values are adjusted for the number of chromosomes tested, either by the Bonferroni correction or by permutation. The genomewide threshold at the specified significance level is given using the same criterion for multiple testing, and the marker selected for each significant chromosome is specified. Thus this gives a much more complete picture of the QTL detection process than the final output. The test statistics from the first iteration may be of interest in order to identify chromosomes which were significant at different alpha levels.

The output from this file is included below as an example.

```
**************************************************
Iteration 1: No. Permutations=0
        Chr1     Chr2    Chr3    Chr4   Chr5   Chr6    Chr7    Chr8  Chr9
 Obs:   80.4961  8.9876  38.2735 14.638 14.581 0.0181  0.0292  0     0.018
 P-val: 0        0.0122  0       6e-04  6e-04  1       1       1     1
5% Genomewide Threshold: 6.4475
Significant chromosomes to be used for scanning/testing:
        Chr1  Chr2  Chr3  Chr4  Chr5
 Mrk:   5     4     6     4     1
************************************************************
Iteration 2: No. Permutations=0
Chromosomes from previous iteration:
        Chr1     Chr2    Chr3    Chr4  Chr5
 Obs:   13.6822  8.0164  0.6127  0     0
 P-val: 5e-04    0.0116  1       1     1
```

```
5% Genomewide Threshold: 5.4119
Significant chromosomes for next round of testing/scanning:
         Chr1  Chr2
 Mrk:    8     9
************************************************************
Iteration 3: No. Permutations=0
Chromosomes from previous iteration:
         Chr1    Chr2
 Obs:    0       0.3385
 P-val:  0.9967  0.5607
5% Genomewide Threshold: 3.8415
```

5.5 Plotting Results

There are two plot functions to visualize the `dlmap` object in addition to tabulating results via the summary function. The default plot function displays the detected QTL on the genetic linkage map. If no QTL are detected, the genetic map itself is plotted for all or a subset of chromosomes. If QTL are detected, this subset is chosen to be those with QTL. In this case, the function will mark the estimated positions of QTL, highlight the flanking markers, and shade the regions between the flanking markers. This helps to visualize where QTL have been detected. The plot for this example is given below, where QTL have been positioned using a step size of 2 cM.

The second type of plot displays the QTL profiles for chromosomes where they were detected. The function `profileplot` takes as input an object of class `dlmap` and constructs plots of the Wald statistic on each chromosome where QTL were detected. These statistics are used to localize QTL and hence the profile plots are analogous to a LOD profile from CIM.

6 Example 2: Multiple phenotypes per genotype

The second example we present here is representative of a more complicated design where we may observe multiple observations per genotype, as in a large field trial. We use the same object `BSdat` for the marker data but the object `BSphe` now contains the phenotypic data. The data is generated as if from a randomized complete block design where we have four observations per genotype. From Figure 6, which shows boxplots of the trait within each block, we can see clear differences between blocks. We can account for this

```
> BSplot <- dlmap.asreml(object=dl.in1, phename="phenotype",
+ step=2)
> plot(BSplot)
```

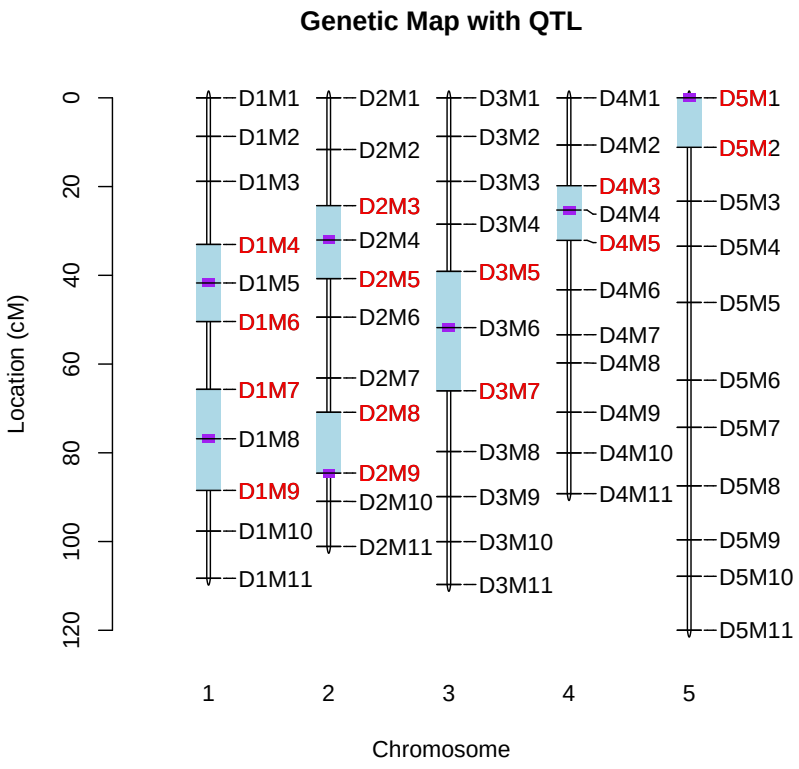

Figure 4: Linkage map for chromosomes with detected QTL

```
> profileplot(BSplot)
```

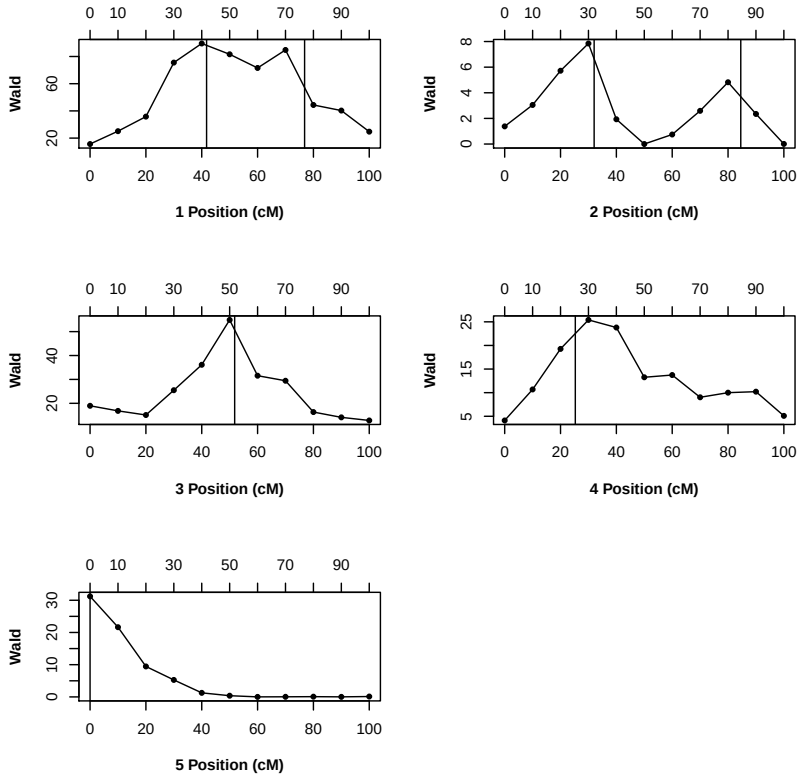

Figure 5: Profile plot for chromosomes with detected QTL

effect using the mixed modeling framework of `dlmap` and thus gain power to detect QTL via the additional observations.

```
> data(BSphe)
> names(BSphe)

[1] "ID"        "Block"     "phenotype"

> table(BSphe$Block)

  1   2   3   4
250 250 250 250

> boxplot(BSphe$phenotype ~ BSphe$Block)
```

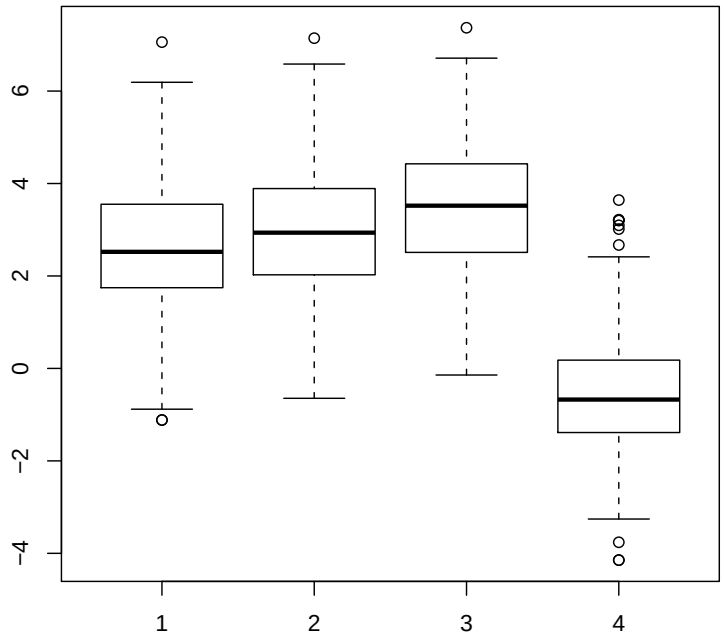

Figure 6: Distribution of quantitative trait within blocks

With `dlmap.asreml` we can analyze the data in the context of the additional phenotypic data in `BSphe`. This is not possible using `dlmap.lme`

or CIM. For this data we construct new input files which include all of the phenotypic data, and then fit a model which has a random effect for block. This requires more time than the simple model due to the larger dataset, but at 166 seconds is still faster than `dlmap.lme`. As previously, all seven QTL are detected, but the QTL effects are more significant due to the increased number of observations. We can also recover the BLUPs for the block random effects through the `final.model` component of the output. The log files have similar formats to those described in the first example.

```
> dl.in2 <- dlcross(format = "rqtl", genobj = BSdat, pheobj = BSphe,
+     idname = "ID")
> summary(dl.in2)

This is an object of class dlcross.
Summary of genetic and phenotypic data:

         This is a bc population.

         No. individuals:          250

         No. phenotypic traits:    3
         Percent phenotyped:       100 100 100

         No. chromosomes:          9

         Total markers:            99
         No. markers:              11 11 11 11 11 11 11 11 11
         Percent genotyped:        100

There are 250 unique genotypes and 1000 unique phenotyped individuals.

> system.time(BSasph <- dlmap.asreml(object=dl.in2, phename = "phenotype",
+  env = T, random = ~Block))

   user  system elapsed
 164.19    1.41  166.02

> summary(BSasph)

 Summary of input data:
```

```
This is an object of class dlcross.
Summary of genetic and phenotypic data:

         This is a bc population.

         No. individuals:          250

         No. phenotypic traits:    3
         Percent phenotyped:       100 100 100

         No. chromosomes:          9

         Total markers:            99
         No. markers:              11 11 11 11 11 11 11 11 11
         Percent genotyped:        100

There are 250 unique genotypes and 1000 unique phenotyped individuals.

 Summary of final results:
     Chr   Pos Left Marker Right Marker Effect    SD Z-value p-value
C1M4   1 33.05        D1M3         D1M5 -0.706 0.072   -9.81       0
C1M8   1 76.84        D1M7         D1M9  -0.76 0.071   -10.7       0
C2M4   2 32.04        D2M3         D2M5 -0.845 0.072  -11.74       0
C2M8   2 70.87        D2M7         D2M9  0.897 0.072   12.46       0
C3M6   3 51.79        D3M5         D3M7 -0.761 0.062  -12.27       0
C4M4   4 25.29        D4M3         D4M5 -0.699 0.062  -11.27       0
C5M1   5     0        D5M1         D5M2 -0.707 0.062   -11.4       0

> BSasph$final.model$coefficients$random

   Block_1    Block_2    Block_3    Block_4
 0.4835664  0.8503207  1.3388842 -2.6727712

> BSasph$final.model$gammas

     Block R!variance
  3.508775   1.000000
```

7 References

1. Broman, K. W. and T. P. Speed. 2002. A model selection approach for the identification of quantitative trait loci in experimental crosses. JRSS-B 64:641-656.

2. Broman, K.W., Wu, H., Sen, S., and G. A. Churchill. 2003. R/qtl: QTL mapping in experimental crosses. Bioinformatics 19:889-890.

3. Huang, B. E. and A. W. George. Look before you leap: A new approach to QTL mapping. *manuscript in preparation.*

4. Wang, S., Basten, C. J. and Z.-B. Zeng. 2007. Windows QTL Cartographer 2.5. Department of Statistics, North Carolina State University, Raleigh, NC.
(http://statgen.ncsu.edu/qtlcart/WQTLCart.htm)