

# A data analysis tutorial for DSPR data

Elizabeth King  
Department of Ecology & Evolutionary Biology  
University of California, Irvine  
egking@uci.edu

April 26, 2012

## 1 Prerequisites

If you do not have R, see (<http://www.r-project.org>) to download and install it and obtain the reference manual.

## 2 Installing the DSPRqtlData Packages

- Install the data package (DSPRqtlDataA/DSPRqtlDataB). If only one population was used, only one data package (A or B) needs to be installed.
- The data packages are ~3.2 GB each and will take several minutes to download and install depending on your connection speed. If you do not install these packages, the functions in DSPRqtl will fetch each position file from the internet. This method takes considerably longer to run.

Within R (not recommended in RStudio), type:

```
install.packages("DSPRqtlDataA",  
                 repos = "http://wfitch.bio.uci.edu/R/",  
                 type = "source")
```

*and/or*

```
install.packages("DSPRqtlDataB",  
                 repos = "http://wfitch.bio.uci.edu/R/",  
                 type = "source")
```

for the pA and pB data packages respectively. Note, you must use the repos statement. These packages are too large to be stored on CRAN.

## 3 Loading and Using the Data Packages

Once the data packages are installed, they can be loaded using:

```
library(DSPRqtlDataA)
```

and/or

```
library(DSPRqtlDataB)
```

Then, the set of additive HMM probabilities and raw HMM probabilities for any given position can be obtained with:

```
data(A_chromosome.arm_position.in.base.pairs)
```

e.g.,

```
data(A_X_12000000)
# This gives a data.frame named A_X_12000000
```

Our list of regularly spaced positions every 10 kb is available here: <http://FlyRILs.org/Data> or within the DSPRqtl package (see below for install instructions).

```
library(DSPRqtl)
data(positionlist_wgenetic)
# This gives a data.frame named poslist
```

## 4 Installing the DSPRqtl Analysis Package

To install the analysis package, within R, type:

```
install.packages("DSPRqtl",
                 repos = "http://wfitch.bio.uci.edu/R/",
                 type = "source")
```

## 5 Data Analysis Tutorial

Before you begin, please also obtain the DSPRqtl manual (see <http://FlyRILs.org/Tools/Tutorial>) and refer to it throughout this tutorial. Also load the DSPRqtl package:

```
library(DSPRqtl)
```

### 5.1 Phenotype Data

Format your phenotype file and read it into R. All phenotype files must have at a minimum a column named "patRIL" (and in the case of crossing designs, also one named "matRIL") containing the numeric RIL IDs (e.g., 11001, 11002, ...) and a column with phenotype measurements (with a name chosen by the user). If covariates are to be used, these should also be in this file (or added to it within R before beginning the analysis). To see an example, load the example ADH data in the DSPR package. Throughout this tutorial, the example dataset ADHdata (included in the R package) will be used to illustrate each step.

```
data(ADHdata)
# a data.frame named ADHdata

head(ADHdata)

##   patRIL matRIL   adh
## 1  11001  11001 -44.09
## 2  11002  11002 -55.76
## 3  11003  11003 -47.58
## 5  11005  11005 -39.89
## 6  11006  11006 -56.91
## 7  11007  11007 -73.39
```

To get your data into R, you must read the file in. R will accept many types of files. For help reading files into R, see (<http://cran.r-project.org/doc/manuals/R-data.html>). The code to read in a text file is shown below as an example.

```
phenotype.dat <- read.table("your_file_path",
                             header = TRUE)
```

## 5.2 Genome Scan

Perform a genome scan. Currently only measurements taken on the sets of inbred lines are available. Crossing design analysis is in development. Handling interactive covariates and analyzing multiple phenotypes at once is also in development. The DSPRscan function performs a genome scan that regresses the 8 founder genotype probabilities on the measured phenotype at evenly spaced 10 kb positions across the genome:

```
DSPRscan(model, design, phenotype.dat, batch)
```

- `model` is the null model. For a phenotype with no covariates, it is of the form  $\text{phenotype} \sim 1$ . With a single additive covariate, it would be:  $\text{phenotype} \sim \text{covariate}$ .
- `design` is either "inbredA" or "inbredB" for the pA and pB RILs.
- `phenotype.dat` is the `data.frame` read in at section 3 containing the RIL IDs, phenotype, and any covariates.
- `batch` is the number of positions tested at one time. Defaults to 1000. If memory use is a problem, reduce this number.

For a single phenotype with the data package installed, a genome scan should take ~15–20 min.

Using the ADH example data:

```
data(ADHdata)
scan.results <- DSPRscan(adh ~ 1,
                          design = "inbredA",
                          phenotype.dat = ADHdata)
```

The output of DSPRscan is a list containing:

- `LODscores` This is a `data.frame` with positions and LOD scores.

- `model` This is the model statement.
- `design` This is the design specified.
- `phenotype` This is the phenotype data.

There is example output from a finished genome scan of the ADH data in this package as well. Type:

```
data(ADHscan)
```

To extract the LOD scores `data.frame`:

```
ADH.lod.scores <- ADHscan$LODscores
```

```
ADH.lod.scores[100:105, ]
```

```
##      chr      Ppos      Gpos      LOD
## 100    X 1150000 0.1359 1.662
## 101    X 1160000 0.1419 1.661
## 102    X 1170000 0.1479 1.661
## 103    X 1180000 0.1540 1.661
## 104    X 1190000 0.1601 1.663
## 105    X 1200000 0.1663 1.664
```

### 5.3 Permutation Test

Perform a permutation test using the function `DSPRperm` to obtain a significance threshold. For initial data exploration, the value 6.8 can be used. This threshold seems to be fairly stable for multiple phenotypes we've tested but we recommend each user performs a permutation test for their specific data set.

```
DSPRperm(model, design, phenotype.dat, batch, niter, alpha)
```

For `model`, `design`, `phenotype.dat`, `batch` see above description for `DSPRscan`. The two additional arguments to `DSPRperm` are:

- `niter` The number of permutations to perform. Default is 1000.
- `alpha` The desired Type I error rate.

The output of `DSPRperm` is a list containing:

- `maxLODs` This is a vector of maximum LOD scores for each permutation.
- `alpha` The specified alpha level
- `threshold` The significance threshold.

Once a permutation test is performed, the `maxLODs` can also be used to determine the significance threshold at another alpha level. For example,

```
perm.test <- DSPRperm(adh ~ 1,
                      design = "inbredA",
                      phenotype.dat = ADHdata)
```

For  $\alpha = 0.01$ :

```
quantile(perm.test$maxLODs, 1-0.01)
```

## 5.4 Identify Significant QTL

Get a summary of the significant peaks. Finding and summarizing significant peaks can be done in a single step using the function `DSPRpeaks`. The individual functions to get the values are also available (see the `DSPRqtl` manual). Once peaks are identified, it is important for the user to confirm these represent distinct peaks.

```
DSPRpeaks(qtldat, threshold, LODdrop)
```

- `qtldat` Output from `DSPRscan`.
- `threshold` The threshold found with the permutation test. Defaults to 6.8.
- `LODdrop` The desired LODdrop for support intervals. Defaults to 2.

The output of `DSPRpeaks` is a list of peaks. Each peak is a list containing:

- `threshold` The specified threshold.
- `peak` The peak position and LOD score.
- `LODdrop` The specified LODdrop.
- `CI` The confidence interval.
- `founderNs` The number of RILs with each founder genotype at the peak.
- `geno.means` The founder means and standard errors at the peak.
- `perct.var` The percent variation explained by the QTL.
- `entropy` The proportion missing information.

Using the `ADHscan` output:

```
peaks <- DSPRpeaks(ADHscan, threshold = 6.8, LODdrop = 2)
```

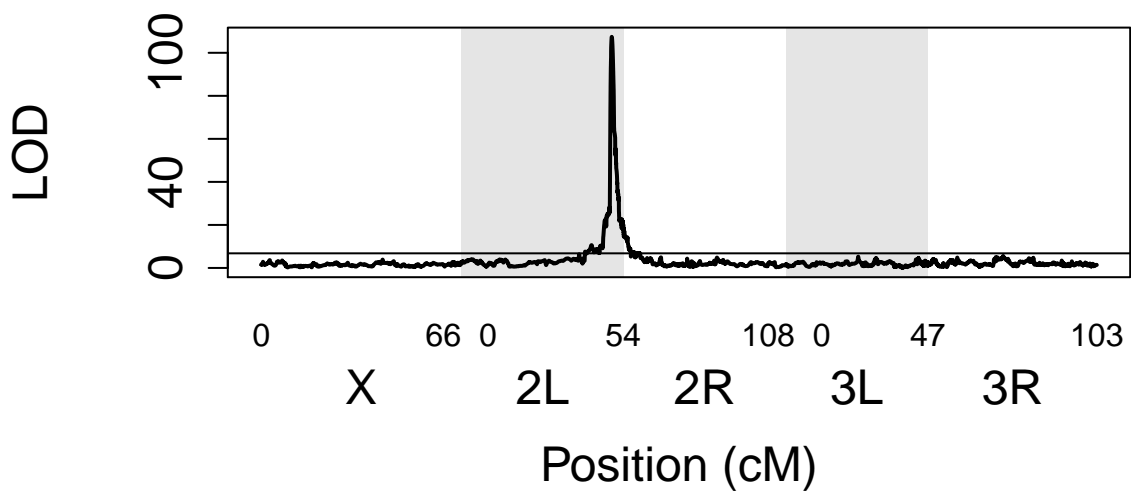
And the main QTL is found at position 26 in the list:

```
peaks[[26]]  
  
## $threshold  
## [1] 6.8  
##  
## $peak  
##   chr    Ppos  Gpos  LOD  
##  26   2L 14500000 49.91 107.4  
##  
## $LODdrop  
## [1] 2  
##  
## $CI
```

```
##          chr      Ppos  Gpos   LOD
## Lower Bound  2L 14410000 49.79 105.4
## Upper Bound  2L 14620000 50.07 105.4
##
## $founderNs
##      A1      A2      A3      A4      A5      A6      A7
##      94      21      1     355     186      20      9
##      A8      Hets Uncertain
##      42      10      84
##
## $geno.means
##      Estimate Std. Error
## A1    -29.03     0.8126
## A2    -32.61     1.7962
## A3    -58.39     6.2789
## A4    -50.40     0.4272
## A5    -49.55     0.5805
## A6    -51.30     1.7197
## A7    -55.61     2.4104
## A8    -53.16     1.2484
##
## $perct.var
## [1] 45.22
##
## $entropy
## [1] 0.03688
##
```

The DSPRscan results can be plotted using the DSPRplot function. Multiple scan results can be plotted on the same plot. Pass the scan results as a `list()` to the DSPRscan function.

```
DSPRplot(list(ADHscan), threshold = 6.8)
```



## 5.5 Local Interval Mapping

The user may wish to perform local interval mapping to compare the peak locations and confidence intervals. The `LocalInt` function will perform interval mapping for a range of positions given by the user. `FindCI` can then be used to re-estimate confidence intervals.

```
LocalInt(peakChr, peakPos, range, phenotype.dat, pheno.name, design)
```

- `peakChr` Chromosome arm at the peak.
- `peakPos` Position in base pairs at the peak.
- `range` The range of positions to examine. Default is 100 (on either side, positions are 10 kb apart)
- `phenotype.dat` The phenotype data.frame. See section 3.
- `pheno.name` The name of the column containing the phenotype.
- `design` "inbredA" or "inbredB"

The output of `LocalInt` is the same as the LODscores from `DSPRscan` but only for the specified set of positions.

Using the ADH sample data:

```
# The main QTL
main.peak <- peaks[[26]]
peakChr <- main.peak$peak$chr
peakPos <- main.peak$peak$Ppos
```

```
peak.int <- LocalInt(peakChr,
                    peakPos,
                    phenotype.dat = ADHdata,
                    pheno.name = "adh",
                    design = "inbredA")
```

## 6 Questions?

If you have any questions or have trouble with this tutorial, please contact [flyrils@gmail.com](mailto:flyrils@gmail.com).