QTL Mapping using Diversity Outbred Mice

Daniel M. Gatti

08 October 2013

1 Introduction

Quantitative Trait Locus (QTL) mapping in DO mice is performed in several steps. First, we use the founder haplotype contributions to perform linkage mapping. In the mapping model, we adjust for kinship between DO mice using the R package QTLRel. Then, we perform permutations to determine and empirical signficance threshold. Next, we select chromosomes with QTL peaks above the signficance threshold, examine the founder allele effects and determine support intervals. Finally, we impute the founder SNPs onto the DO genomes to perform association mapping in the QTL intervals.

2 Mapping Models

2.1 Linkage Mapping

Linkage mapping involves the use of founder haplotype probabilities. We perform point mapping at each marker on the array. We fit an additive model that regresses the phenotype on the eight founder haplotype contributions and incorporates an adjustment for the kinship between samples.

$$y = X\alpha + H\beta + Zu + \varepsilon \tag{1}$$

where:

- n is the number of samples
- y is an n x 1 vector of phenotype values for each sample
- X is an n x p matrix of p fixed covariates (sex, diet, etc.)
- α is a p x 1 vector of fixed effects
- H is an n x 8 matrix of founder haplotype contributions (each row sums to 1)
- β is an 8 x 1 vector of founder haplotype effects
- Z is an n x n matrix of error covariances between samples
- u is an n x 1 vector of ???
- ε is an n x 1 vector of residual errors

2.2 Association Mapping

Between each pair of markers, we assign the genotype state with the highest probability to each DO sample. We then query the Sanger Mouse Genomes SNP file to obtain all of the founder SNPs in the interval.

For each Sanger SNP, we impute the Sanger SNPs onto DO genomes as follows:

$$a_j = \sum_{i=1}^{8} s_i h_{ij} \tag{2}$$

where:

- a is the allele call (coded as 0, 1 or 2) for sample j
- s is the Sanger founder allele call (coded as 0 or 1)
- h is the founder haplotype contribution of founder i for sample j

$$y = X\alpha + A\beta + Zu + \varepsilon \tag{3}$$

where:

- n is the number of samples
- y is an n x 1 vector of phenotype values for each sample
- X is an n x p matrix of p fixed covariates (sex, diet, etc.)
- α is a p x 1 vector of fixed effects
- A is an n x 3 matrix of imputed allele calls
- β is an 3 x 1 vector of allele effects
- Z is an n x n matrix of error covariances between samples
- u is an n x 1 vector of ???
- ε is an n x 1 vector of residual errors

3 QTL Mapping

We will use example data from Svenson et.al, *Genetics*, 2012. Breifly, 149 mice (75 F, 74 M) were placed on either a chow (n = 100) or a high fat diet (n = 49). A variety of clinical phenotypes were measured at two time points, roughly 14 weeks apart. In this example, we will map the hemoglobin distribution width (HDW) at the second time point. We will load this data from the Bioconductor data package `MUGAExampleData`.

```
> library(DOQTL)
> library(MUGAExampleData)
> data(pheno)
> data(model.probs)
```

QTL mapping requires phenotype and genotype data. Here, we have a `data.frame` of phenotypes called `pheno` and a 3D array of founder haplotype contributions (num.samples x 8 founders x num.markers) called `model.probs`. The sample IDs must be in rownames(pheno) and dimnames(model.probs)[[1]] and they must match each other. We will map the hemoglobin distribution width at time point 2 (HDW2).

First, we need to create a kinship matrix using the founder contributions.

```
> K = kinship.probs(model.probs)
```

Second, we need to create a matrix of additive covariates to run in the model. In this case, we will use sex, diet and CHOL1. Note that the sample IDs must be in `rownames(covar)`.

```
> covar = data.frame(sex = as.numeric(pheno$Sex == "M"), diet = as.numeric(pheno$Diet == "hf"))
> rownames(covar) = rownames(pheno)
```

Third, we need to get the marker locations on the array.

```
> load(url("ftp://ftp.jax.org/MUGA/muga_snps.Rdata"))
```

Fourth, we map the phenotype using `scanone`.

```
> qtl = scanone(pheno = pheno, pheno.col = "HDW2", probs = model.probs, K = K,
+               addcovar = covar, snps = muga_snps)

[1] "Mapping with 141 samples."
[1] "Mapping with 7654 markers."
[1] "HDW2"
Warning: solution lies close to zero for some positive variance components, their standard errors may no
Warning: solution lies close to zero for some positive variance components, their standard errors may no
```

Fifth, we run permutations to determine significane thresholds. We recommend running at least 1,000 permutations. In this demo, we run 100 permutations to save time.

```
> perms = scanone.perm(pheno = pheno, pheno.col = "HDW2", probs = model.probs,
+         addcovar = covar, snps = muga_snps, nperm = 100)
> thr = quantile(perms, probs = 0.95)
```

We then plot the LOD curve for the QTL.

```
> plot(qtl, sig.thr = thr, main = "HDW2")
```

The largest peak appears on Chr 9. The linkage mapping model (Eqn. 1) produces an estimate of the effect of each founder allele at each marker. We can plot these effects (model coefficients) on Chr 9 to see which founders contribute to a high HDW.

```
> coefplot(qtl, chr = 9)
```

Note that the DO mice with alleles from three strains, 129S1/SvImJ, NZO/HlLtJ and WSB/EiJ, have lower changes in cholesterol than the other five strains. Remember these strains because they will appear again below. We then determine the width of the QTL support interval using `bayesint`. Note that this function only provides reasonable support intervals if there is a single QTL on the chromosome.

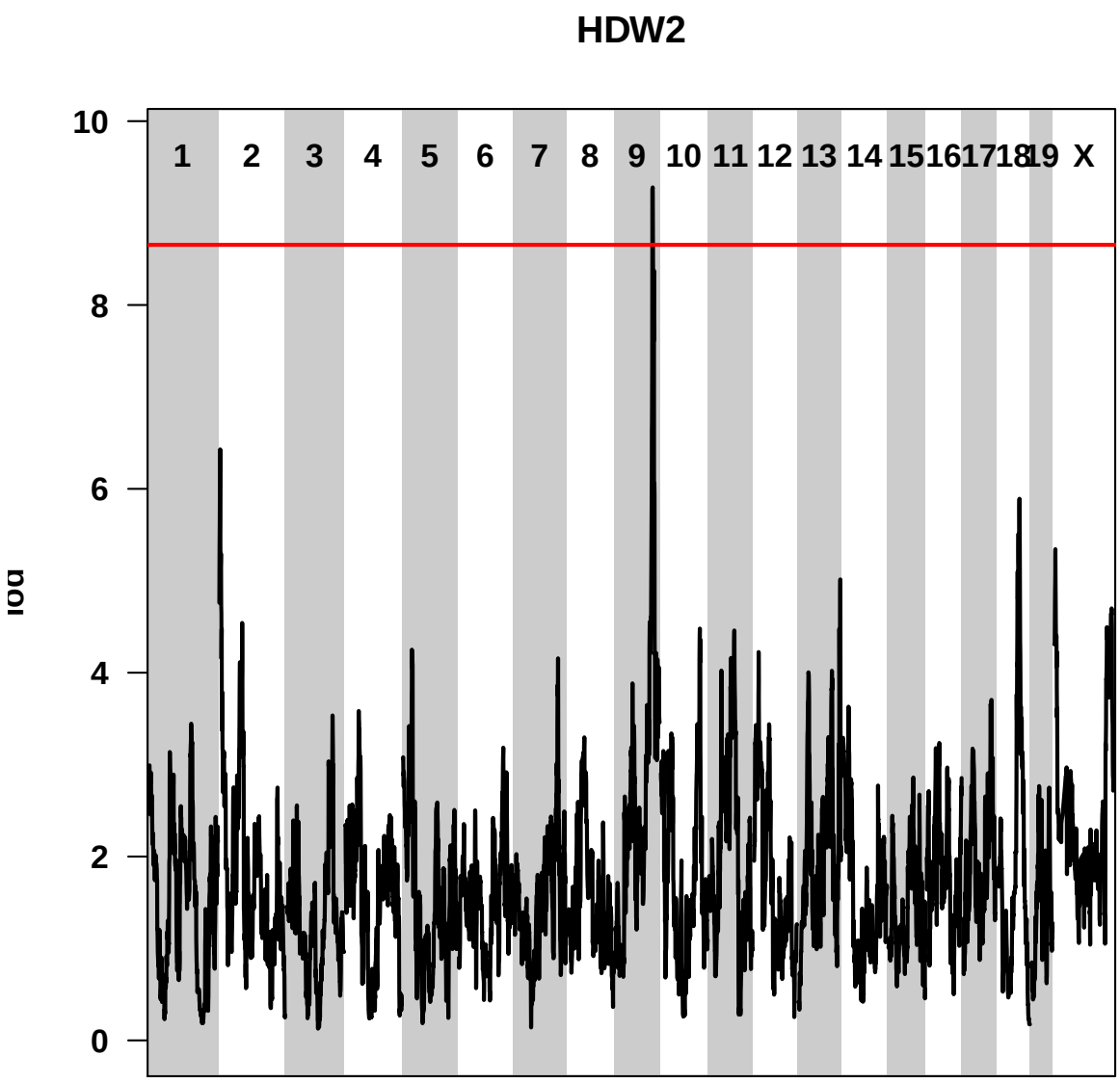

Figure 1: QTL plot of HDW2. The LOD of the mode in Eqn. 1 is plotted along the mouse genome. The red line is the $p < 0.05$ significance threshold.

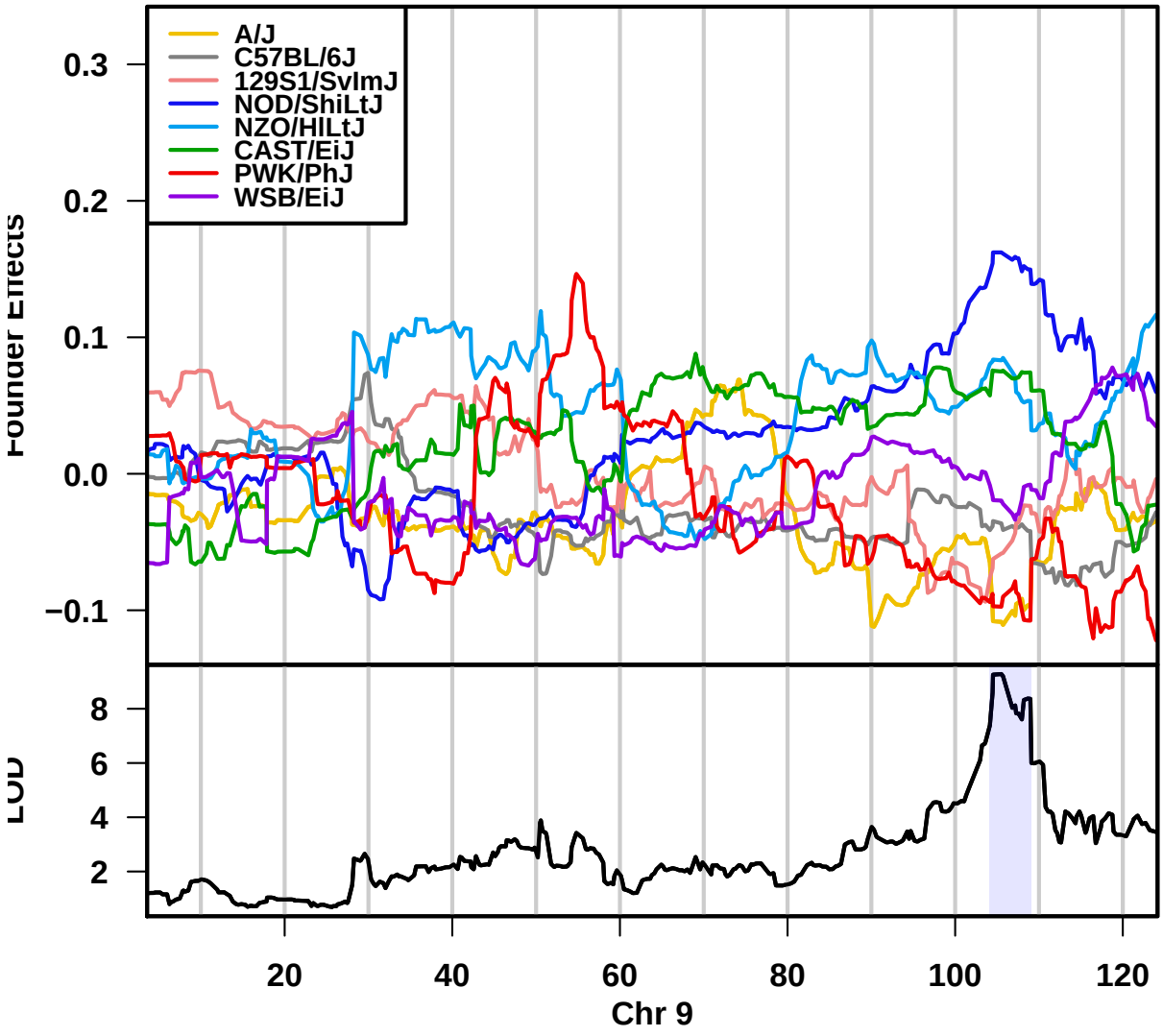

Figure 2: Coefficient plot of HDW2 on Chr 9. The top panel shows the 8 estimated founder allele effects along Chr 9. The NOD/ShiLtJ allele contributes to high values and the A/J and PWK/PhJ alleles contribute to low values. The bottom panel shows the LOD score.

```
> interval = bayesint(qtl, chr = 9)
> interval
```

```
                  marker chr      pos     cM perc.var      lrs      lod
UNC090280590 UNC090280590   9 104.1423 49.551 22.27697 34.02250 7.387892
UNC091160886 UNC091160886   9 105.5128 50.043 27.13349 42.73303 9.279360
UNC090227520 UNC090227520   9 109.0960 53.114 18.50575 27.62609 5.998929
                         p neg.log10.p
UNC090280590 1.705843e-05    4.768061
UNC091160886 3.755845e-07    6.425292
UNC090227520 2.569688e-04    3.590120
```

The QTL support interval is 4.7 Mb wide. Finally, we narrow the candidate gene list by imputing the founder SNPs onto the DO genomes. This idea is essentially assocation mapping in an outbred population.

```
> ma = assoc.map(pheno = pheno, pheno.col = "HDW2", probs = model.probs, K = K,
+                      addcovar = covar, snps = muga_snps, chr = interval[1,2],
+                      start = interval[1,3], end = interval[3,3])
```

```
[1] "Mapping with 135 samples."
[1] "Retrieving SNPs..."
[1] "Retrieved 139299 SNPs."
[1] "Retaining 127565 high quality SNPs."
[1] "Retaining 65528 polymorphic SNPs."
[1] "Calculating mapping statistic..."
Warning: solution lies close to zero for some positive variance components, their standard errors may no
```

```
> tmp = assoc.plot(ma, thr = 4)
> unique(tmp$sdps)
```

```
NULL
```

We can get the genes in the QTL interval using the `get.mgi.features()` function.

```
> mgi = get.mgi.features(chr = interval[1,2], start = interval[1,3],
+       end = interval[3,3], type = "gene", source = "MGI")
> nrow(mgi)
```

```
[1] 220
```

```
> head(mgi)
```

```
     seqid source type     start      stop score strand phase              ID
1        9    MGI gene 104002544 104153483     .      +     . MGI:MGI:1921275
6        9    MGI gene 104151282 104262930     .      -     . MGI:MGI:2676368
923      9    MGI gene 104262105 104263617     .      +     . MGI:MGI:5610791
939      9    MGI gene 104288240 104337728     .      -     . MGI:MGI:1928480
991      9    MGI gene 104301928 104304909     .      -     . MGI:MGI:5610416
1158     9    MGI gene 104355987 104385032     .      +     . MGI:MGI:5579254
        Name Parent
```

```
[1] "Mapping with 135 samples."
[1] "Retrieving SNPs..."
[1] "Retrieved 139299 SNPs."
[1] "Retaining 127565 high quality SNPs."
[1] "Retaining 65528 polymorphic SNPs."
[1] "Calculating mapping statistic..."
Warning: solution lies close to zero for some positive variance components, their standard errors may no

NULL
```

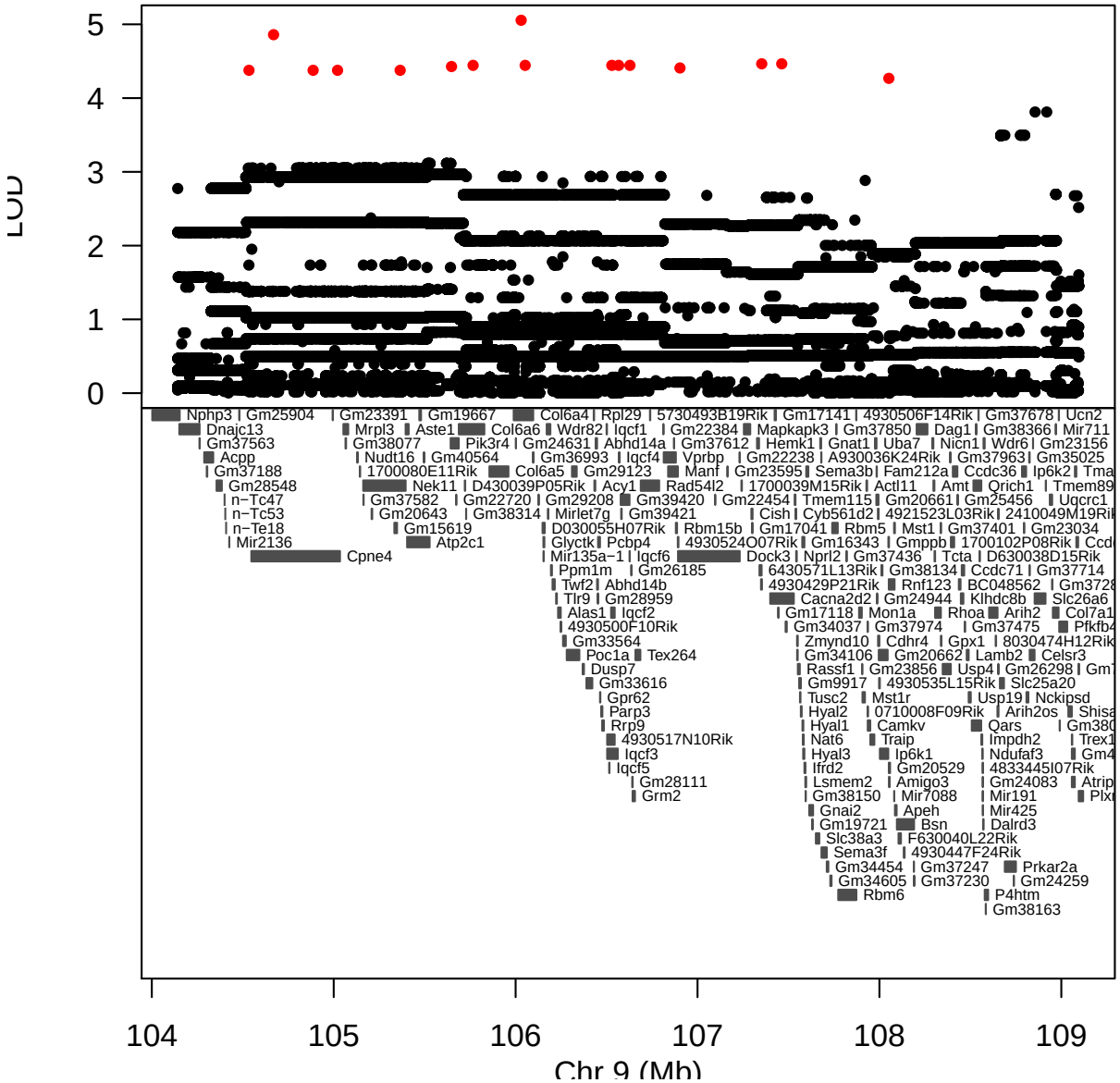

Figure 3: Association mapping plot of HDW2 in the Chr 9 support interval. The top panel shows the LOD score from assocation mapping (Eqn. 3) in the QTL support interval. The bottom panel shows the genes and non-coding RNAs from the Mouse Genome Informatics database.

```
1      Nphp3     NA
6    Dnajc13     NA
923  Gm37563     NA
939     Acpp     NA
991  Gm37188     NA
1158 Gm28548     NA
                                                                      Dbxref
1        VEGA:OTTMUSG00000031730,NCBI_Gene:74025,ENSEMBL:ENSMUSG00000032558
6       VEGA:OTTMUSG00000049291,NCBI_Gene:235567,ENSEMBL:ENSMUSG00000032560
923                       VEGA:OTTMUSG00000049370,ENSEMBL:ENSMUSG00000104040
939      VEGA:OTTMUSG00000024988,NCBI_Gene:56318,ENSEMBL:ENSMUSG00000032561
991                       VEGA:OTTMUSG00000049372,ENSEMBL:ENSMUSG00000102183
1158 VEGA:OTTMUSG00000049293,NCBI_Gene:102636046,ENSEMBL:ENSMUSG00000099599
                                                mgiName              bioType
1                       nephronophthisis 3 (adolescent) protein coding gene\r
6    DnaJ heat shock protein family (Hsp40) member C13 protein coding gene\r
923                               predicted gene%2c 37563   unclassified gene\r
939                          acid phosphatase%2c prostate protein coding gene\r
991                               predicted gene%2c 37188   unclassified gene\r
1158                                 predicted gene 28548        lincRNA gene\r
```

There are 169 genes in the QTL support interval. Several SNPs have LOD scores above 4. This is a somewhat arbitrary cutoff and an appropriate threshold will be supplied in future version of DOQTL. In this case, there may be more than one variant that influences the phenotype.

4 SessionInfo

```
> sessionInfo()

R version 3.5.1 Patched (2018-07-12 r74967)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 16.04.5 LTS

Matrix products: default
BLAS: /home/biocbuild/bbs-3.8-bioc/R/lib/libRblas.so
LAPACK: /home/biocbuild/bbs-3.8-bioc/R/lib/libRlapack.so

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats4    parallel  stats     graphics  grDevices utils     datasets
[8] methods   base

other attached packages:
```

```
 [1] MUGAExampleData_1.2.0             DOQTL_1.18.0
 [3] VariantAnnotation_1.28.1          Rsamtools_1.34.0
 [5] SummarizedExperiment_1.12.0       DelayedArray_0.8.0
 [7] BiocParallel_1.16.0               matrixStats_0.54.0
 [9] Biobase_2.42.0                    BSgenome.Mmusculus.UCSC.mm10_1.4.0
[11] BSgenome_1.50.0                   rtracklayer_1.42.0
[13] Biostrings_2.50.0                 XVector_0.22.0
[15] GenomicRanges_1.34.0              GenomeInfoDb_1.18.0
[17] IRanges_2.16.0                    S4Vectors_0.20.0
[19] BiocGenerics_0.28.0

loaded via a namespace (and not attached):
 [1] httr_1.3.1               bit64_0.9-7              foreach_1.4.4
 [4] gtools_3.8.1             assertthat_0.2.0         blob_1.1.1
 [7] GenomeInfoDbData_1.2.0   robustbase_0.93-3        progress_1.2.0
[10] RSQLite_2.1.1            lattice_0.20-35          RUnit_0.4.32
[13] digest_0.6.18            Matrix_1.2-15            XML_3.98-1.16
[16] pkgconfig_2.0.2          biomaRt_2.38.0           zlibbioc_1.28.0
[19] xtable_1.8-3             corpcor_1.6.9            mvtnorm_1.0-8
[22] gdata_2.18.0             annotate_1.60.0          regress_1.3-15
[25] GenomicFeatures_1.34.1   nnet_7.3-12              magrittr_1.5
[28] crayon_1.3.4             mclust_5.4.1             memoise_1.1.0
[31] doParallel_1.0.14        MASS_7.3-51.1            hwriter_1.3.2
[34] class_7.3-14             tools_3.5.1              prettyunits_1.0.2
[37] hms_0.4.2                trimcluster_0.1-2.1      stringr_1.3.1
[40] Rhdf5lib_1.4.0           kernlab_0.9-27           cluster_2.0.7-1
[43] AnnotationDbi_1.44.0     fpc_2.1-11.1             compiler_3.5.1
[46] rlang_0.3.0.1            rhdf5_2.26.0             grid_3.5.1
[49] RCurl_1.95-4.11          iterators_1.0.10         bitops_1.0-6
[52] annotationTools_1.56.0   codetools_0.2-15         flexmix_2.3-14
[55] DBI_1.0.0                QTLRel_1.0               R6_2.3.0
[58] GenomicAlignments_1.18.0 prabclus_2.2-6           bit_1.1-14
[61] modeltools_0.2-22        stringi_1.2.4            Rcpp_0.12.19
[64] DEoptimR_1.0-8           diptest_0.75-7
```