

# Statistical analysis code for analysis of CASTxB6 F2 mouse cross

## 2. QTL analysis

Peter Langfelder

May 3, 2011

### Contents

1 Preliminaries	1
2 Permutation tests	2
3 QTL analysis of HDL with sex as a covariate	2
3.a HDL variance explained by peak markers	8
4 Sex x genotype interactions	9
5 eQTL of genes identified as consistently related to and causal for HDL	14

### 1 Preliminaries

In this document we detail the QTL analysis of the CASTxB6 cross. The first step is to set up the R session and load the requisite genotype data. Missing data have been imputed using a standard imputation procedure. Quality analysis of genotypes indicated that many of the B6xC6378 animals had genotypes not consistent with expected proportions of homozygous and heterozygous alleles in an F2 cross. For this reason we exclude all B6xC6378 animals from this study.

```
# Load the WGCNA library and the qtl library
library(WGCNA)
library(qtl)
# Source additional custom functions (for producing QTL plots)
source("networkFunctions-extras-05.R");
# Load the imputed snp data
file = bzfile("../Data-CXB/CXB_BXC_MF_clinical_traits_GT_imputed_031408.csv.bz2");
geno0 = read.csv(file);
# Display the first few rows and columns
geno0[1:5, 1:20]
# Isolate marker names
markerNames = names(geno0)[substring(names(geno0), 1, 2) %in% c("rs", "mC") ];
genoTable0 = geno0[, c(1, match(markerNames, names(geno0))) ];
# For the cQTL analysis: keep all good mice with valid genotypes.
dirInfo = read.csv("../QTL/BxC-CrossDirection-correctedByGenotypeAndSource.csv");
goodMice = dirInfo$genotypeQuality=="good" & dirInfo$direction=="CxB";
keepMice = intersect(geno0$Mice_id, dirInfo$Mice_id[goodMice]);
```

```
length(keepMice)
# Note: 293 mice
geno = genoTable0[ match(keepMice, geno0$Mice_id), ];
```

We next split the clinical traits from the genotype table

```
traits0 = geno0[, -match(markerNames, names(geno0))];
traits1 = traits0[, -c(2:7, 9:17) ];
nTraits = ncol(traits1);
```

We now use a custom function to turn the genotypes and phenotypes into a cross object that the qtl package can work with. This function removes trait outliers whose deviation from the mean is larger than 3 standard deviations.

```
# Note: makeCross removes outlier traits with std deviation more than +/- 3
crossData = makeCross(geno, traits1, outlierZ = 3);
summary(crossData$cross)
traitsForQTL = crossData$cross$pheno;
chromo = crossData$chromo
basePair = crossData$basePair
```

## 2 Permutation tests

Next we perform permutation tests. These tests will take several hours to several days, depending on the computer these calculations are run on. We run a permutation test for the analysis above (sex as a covariate), and we also run a permutation analysis for the sex x genotype interaction analyzed below. We save the permutation test results so the permutations need only be run once.

```
set.seed(10);
system.time( {permTest = scanone(cross, pheno.col = c(3:nTraits), addcovar = sex,
                              method = "mr", n.perm = 10000, verbose = TRUE, n.cluster = 8)})
# Save the results
save(permTest, file = "qtl-permutationTest-allGoodMice.RData");
# Permutation test for sex x genotype interaction
set.seed(10); # This seed must be the same as the one for the permutation test with additive covariate
print(system.time({
  permTest.int = scanone(crossData$cross, pheno.col = c(3:nTraits), addcovar = sex, intcovar = sex,
                       method = "mr", n.perm = 10000, verbose = TRUE, n.cluster = 8)}));
save(permTest.int, file = "qtl-permutationTest-allGoodMice-sexInt.RData");
```

## 3 QTL analysis of HDL with sex as a covariate

Since HDL is different between female and male mice, we add sex as a covariate in all QTL analyses. We calculate the LOD scores for all measured traits.

```
sex = as.numeric(factor(crossData$cross$pheno$sex));
qtl = scanone(crossData$cross, pheno.col = c(3:nTraits), addcovar = sex, method = "mr");
nSNPs = nrow(qtl);
HDLcolumns = match(c("b_hdl_mgd1", "e_hdl_mgd1"), colnames(qtl));
# write the qtl table into a file
tmp = qtl;
class(tmp) = "data.frame";
out = cbind(SNP = rownames(lod.sexInt), tmp);
write.csv(out, file = gzfile("CxB-MF-sexCovariate-cQTL.csv.gz"), row.names = FALSE,
          quote = FALSE)
```

We now use the permutation analysis to calculate LOD thresholds corresponding to genome-wide significance p-values of 0.01, 0.05, 0.10, and 0.20. Below we will concentrate on the  $p = 0.05$  (significant) and  $p = 0.20$  (suggestive) levels.

```
thresholds = apply(permTest, 2, quantile, prob = c(0.99, 0.95, 0.9, 0.8));
hdlCols = grep("hdl", colnames(qt1))[1:2]
hdlCols.traits = grep("hdl", colnames(traitsForQTL))[1:2]
minLOD = mean(thresholds[2, hdlCols-2]);
suggestiveLOD = mean(thresholds[4, hdlCols-2]);
```

We now use custom functions to identify QTL peaks and peak regions. Peak region boundaries are SNPs at which the LOD scores drop by more than 1.5 from the peak.

```
peakInfo = apply(qt1[, hdlCols], 2, peakRegions, chromo, basePair, minLOD = minLOD,
               lodDrop = 1.5, minPeakDistance = 2e7);
suggestivePeakInfo = apply(qt1[, hdlCols], 2, peakRegions, chromo, basePair, minLOD = suggestiveLOD,
                          lodDrop = 1.5, minPeakDistance = 2e7);
peakSNPnames = rownames(qt1)[peakInfo$e_hdl_mgdl$peakIndicator];
peakSNPextNames = paste(peakSNPnames, peakInfo$e_hdl_mgdl$peakChromo, peakInfo$e_hdl_mgdl$peakBp,
                       sep = ".");
```

We next produce a plot of the LOD scores for HDL complete with indicators of significance, peaks, and peak region boundaries.

```
nSNPs = nrow(qt1);
prettyNames = c("HDL, Chow diet", "HDL, Western diet");
# Open a suitably sized graphics window
sizeGrWindow(12, 9);
# ..alternatively, can also plot into a pdf file
#pdf(file = "Plots/qt1-LODscoresForHDL.pdf", width = 12, h=9);
par(mfrow = c(2,1))
par(mar = c(3.3,3,2.2, 0.2));
par(mgp = c(1.7, 0.6, 0));
for (tr in 1:2) # This loop is for b_hdl and e_hdl
{
  point.x = plotQTL(qt1[, hdlCols[tr]],
                   chromo, basePair,
                   col = c("black"),
                   lwd = 1, ylab = "LOD",
                   main = spaste(LETTERS[tr], ". QTL analysis for ", prettyNames[tr]),
                   bpBasedSpacing = TRUE, gridColor = "grey", grid.lty = 2);
  # Add grid and significance indicators
  addGrid(linesPerTick = 2);
  abline(h = minLOD, col = "red", lty = 2);
  abline(h = suggestiveLOD, col = "green", lty = 2);
  # Add peak location markers
  nPeaks = sum(suggestivePeakInfo[[tr]]$peakIndicator);
  peakLocs = suggestivePeakInfo[[tr]]$peakIndex;
  points(point.x[peakLocs], qt1[peakLocs, hdlCols[tr]], pch = 21, col = "blue", lty = 2);
  # Add peak region markers
  fromLoc = suggestivePeakInfo[[tr]]$peakStartInd;
  toLoc = suggestivePeakInfo[[tr]]$peakEndInd;
  points(point.x[fromLoc], qt1[fromLoc, hdlCols[tr]], pch = 4, col = "darkgreen", lty = 2);
  points(point.x[toLoc], qt1[toLoc, hdlCols[tr]], pch = 4, col = "darkgreen", lty = 2);
  legendClean("topright", col = c("black", "red", "green"), lty=c(1,2,2), lwd = 1,
             legend = c("QTL LOD",
                       "p=0.05 threshold (significant)", "p=0.2 threshold (suggestive)"))
}
```

```
# If plotting into a file, close it:  
dev.off();
```

The resulting plot is Figure 2 in our article, and is also reproduced in Figure 1.

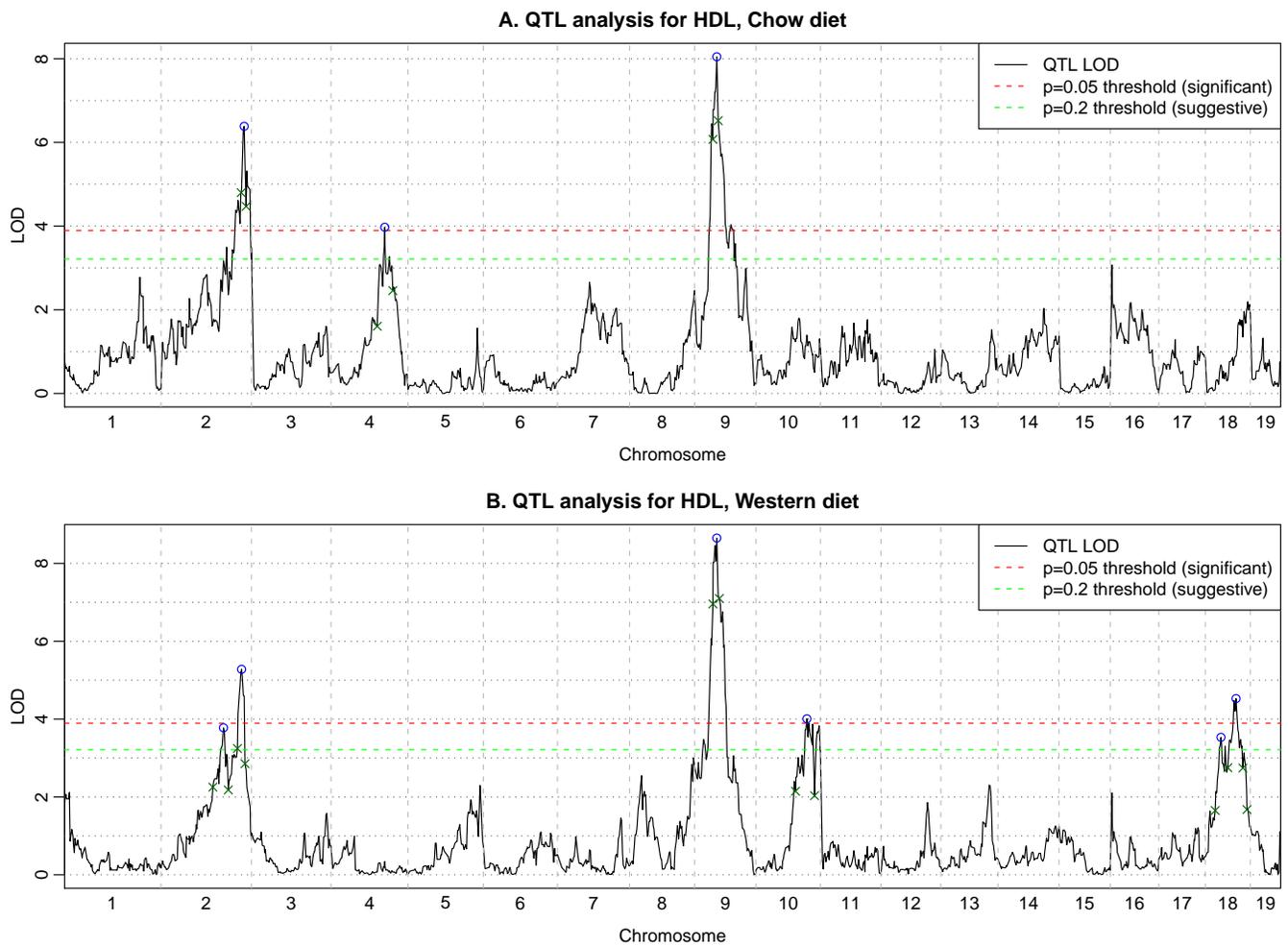


Figure 1: LOD scores of genotyped SNP markers for HDL on chow (A) and Western (B) diet.

Next we plot boxplots of HDL vs. genotypes at the peak mearkers.

```
alleleNames = c("B6", "H", "CAST");
# Open a suitably sized graphics window
sizeGrWindow(10, 8);
# ..alternatively, can also plot into a pdf file
#pdf(file = "Plots/ctl-HDLvsPeakGenoBoxplots.pdf", width = 10, h=8);
# Screen sectioning
layout(matrix(c(1,4,2,5,3,6,0,7), 2, 4));
par(mar = c(2.0, 4.8, 6.3, 2.4));
par(mgp = c(2.3, 0.6, 0));
# Loop over chow and western diet HDL
for (tr in 1:2)
{
  nPeaks = length(peakInfo[[tr]]$peakIndex);
  peakSNPnames = rownames(ctl)[peakInfo[[tr]] $peakIndicator];
  peakSNPextNames = paste(peakSNPnames, peakInfo[[tr]] $peakChromo, peakInfo[[tr]] $peakBp, sep = ".");
  hdl = traitsForQTL[, hdlCols.traits[tr]];
  for (p in 1:nPeaks)
  {
    peakGt = geno[, -1][,peakInfo[[tr]] $ peakIndex [p]];
    verboseBoxplot(hdl, peakGt, names = alleleNames, ylab = prettyNames[tr],
                  main = spaste(prettyNames[tr], " vs.\n", peakSNPnames[p], "\nChr. ",
                                peakInfo[[tr]] $peakChromo[p], ", pos. ",
                                round(peakInfo[[tr]] $peakBp[p]/1e6, 2),
                                " MB\n"), cex.main = 1.4, xlab = "");
    addGrid(linesPerTick = 2)
  }
}
# If plotting into a file, close it: dev.off();
```

The resulting boxplots are shown in Figure 2. At peaks on chromosomes 2,4,10, and 18, the CAST allele lowers HDL, while on chromosome 9 the CAST allele raises cholesterol.

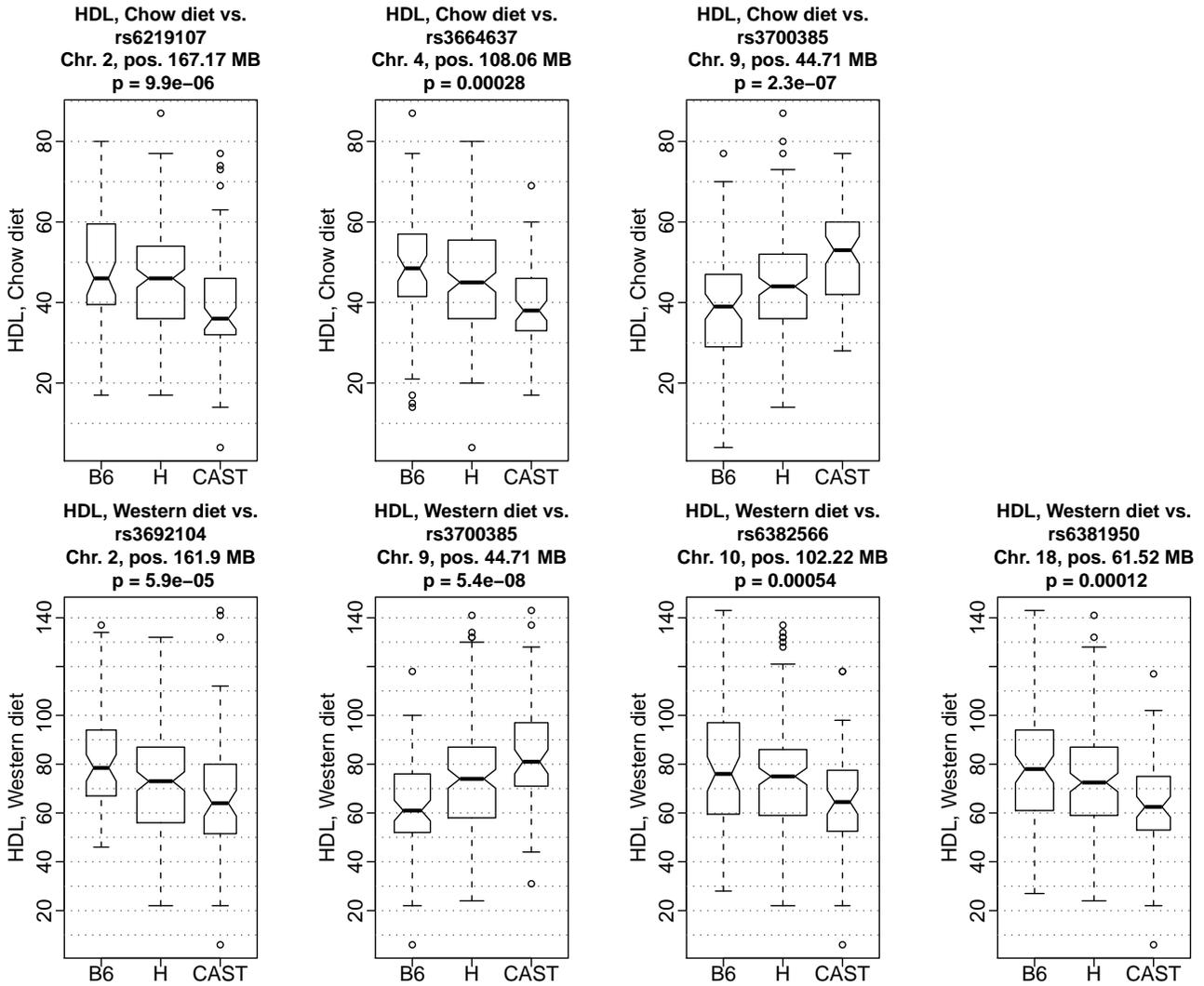


Figure 2: HDL vs. genotype at peak markers identified in the QTL analysis. Each plot shows HDL as a function of the three alleles at one of the peaks.

We also produce and save text tables with the median HDL levels and confidence intervals per genotype. We use the function `boxplot.stats` to calculate the requisite statistics. These tables are included in our main article.

```
# Helper function to isolate relevant statistics
boxplot.stats1 = function(...)
{
  s = boxplot.stats(...);
  interval = (s$conf[2] - s$conf[1])/2
  spaste(signif(s$stats[3], 2), " (", signif(interval, 1), ")")
}
# Set up tables
nPermutations = nrow(permTest);
tables = list();
# Loop over chow and Western HDL
for (tr in 1:2)
{
  nPeaks = length(suggestivePeakInfo[[tr]]$peakIndex);
  peakSNPnames = rownames(ctl)[suggestivePeakInfo[[tr]] $peakIndicator];
  hdl = traitsForQTL[, hdlCols.traits[tr]];
  tables[[tr]] = data.frame(matrix(0, nPeaks, 9));
  tables[[tr]][, 1] = peakSNPnames;
  tables[[tr]][, 2] = suggestivePeakInfo[[tr]] $peakChromo;
  tables[[tr]][, 3] = suggestivePeakInfo[[tr]] $peakBp;
  for (p in 1:nPeaks)
  {
    peakGt = geno[, -1][ ,suggestivePeakInfo[[tr]] $ peakIndex [p]];
    #means = tapply(hdl, peakGt, mean, na.rm = TRUE);
    #sd = tapply(hdl, peakGt, stdErr);
    #tables[[tr]][p, 4:6] = spaste(signif(means, 3), " (", signif(sd, 2), ")");

    stats = tapply(hdl, peakGt, boxplot.stats1, do.out = FALSE);
    tables[[tr]][p, 4:6] = stats;
    tables[[tr]][p, 7] = signif(kruskal.test(hdl, factor(peakGt))$p.value, 1);
    lod = ctl[suggestivePeakInfo[[tr]] $ peakIndex [p], hdlCols[tr]];
    tables[[tr]][p, 8] = signif(lod, 2);

    lod.p = (sum(permTest[, hdlCols[tr]-2] > lod) + 1)/nPermutations
    tables[[tr]][p, 9] = lod.p
  }
  colnames(tables[[tr]]) = c("SNP", "Chr", "Bp", "BB", "BC", "CC", "p-value", "LOD", "Perm. p-value");
  # Save the table into a csv file
  fileName = spaste("ctl-anovaFor-", colnames(traitsForQTL)[hdlCols.traits[tr]], ".csv")
  printFlush(paste("Saving table into file", fileName))
  write.csv(tables[[tr]], file = fileName, row.names = FALSE, quote = FALSE);
}
}
```

### 3.a HDL variance explained by peak markers

Here we calculate the proportion of HDL variance explained by the found markers using a leave-one-out scheme. For each sample we regress HDL on marker genotypes in a training set (all but the selected sample) and predict the HDL for the sample. In this way we obtain a less biased estimate of the proportion of variance explained. However, since the peak locations were determined from all data, the proportion of variance explained cannot be considered unbiased.

```
# Define two helper functions to simplify the calculations
# Linear model regression and prediction
```

```

prediction = function(x1, y1, xtest1)
{
  fit = lm(y1~., data = as.data.frame(x1));
  xtest1 = as.data.frame(xtest1);
  prediction = predict(fit, newdata = xtest1);
  prediction;
}
# Leave-one-out calculation of proportion of variance explained
propVarExplainedByPredictors = function(x,y)
{
  present = is.finite(y);
  y = y[present];
  x = x[present, ];
  nSamples = length(y);
  y.pred = rep(NA, nSamples);
  for (s in 1:nSamples)
    y.pred[s] = prediction(x[-s, ], y[-s], x[s, , drop = FALSE]);
  cor(y, y.pred, use = 'p')^2;
}
# Main code
for (tr in 1:2)
{
  peakGeno = as.matrix(geno[ , suggestivePeakInfo[[tr]]$peakIndex]);
  hdl = traitsForQTL[, hdlCols.traits[tr]];
  print(propVarExplainedByPredictors(peakGeno, hdl));
}

```

The results are approximately 0.16 and 0.24. Hence, peak markers explain approximately 16 % and 24 % of the variation of chow and Western diet HDL, respectively.

## 4 Sex x genotype interactions

Here we perform a QTL analysis with sex as an interactive trait. To obtain LOD for sexx genotype interactions, we subtract the LOD for sex as a covariate from the LOD for sex as a covariate and interacting trait.

```

qtl.int = scanone(crossData$cross, pheno.col = c(3:nTraits), addcovar = sex, intcovar = sex, method = "mr");
lod.sexInt = qtl.int - qtl;
lod.sexInt[lod.sexInt < 0] = 0;
# Save the results in a csv table
tmp = lod.sexInt;
class(tmp) = "data.frame";
out = cbind(SNP = rownames(lod.sexInt), tmp);
write.csv(out,
          file = gzfile("CxB-MF-sexInteraction-cQTL-LOD.csv.gz"), row.names = FALSE, quote = FALSE)

```

The same procedure performed on the permutation scores allows us to find the significance thresholds.

```

perm.sexInt = permTest.int - permTest;
thresholds.int = summary(perm.sexInt, alpha = c(0.05, 0.20))
minLOD.int = thresholds.int[1, hdlCols-2];
suggestiveLOD.int = thresholds.int[2, hdlCols-2];

```

We now identify peaks and peak regions:

```

peakInfo.int = list();
suggestivePeakInfo.int = list();

```

```

peakSNPNames.int = list();
peakSNPextNames.int = list();
for (tr in 1:2)
{
  peakInfo.int[[tr]] = peakRegions(as.numeric(lod.sexInt[, hdlCols[tr]]),
    chromo, basePair, minLOD = minLOD.int[tr],
    lodDrop = 1, minPeakDistance = 2e7);

  suggestivePeakInfo.int[[tr]] = peakRegions(as.numeric(lod.sexInt[, hdlCols[tr]]),
    chromo, basePair, minLOD = as.numeric(suggestiveLOD.int[tr]),
    lodDrop = 1, minPeakDistance = 2e7);
  peakSNPNames.int[[tr]] = rownames(ql)[peakInfo[[tr]]$peakIndicator];
  peakSNPextNames.int[[tr]] = paste(peakSNPNames[[tr]], peakInfo[[tr]]$peakChromo,
    peakInfo[[tr]]$peakBp, sep = ".");
}

```

Next we plot the interaction LOD scores together with identified peaks.

```

prettyNames.int = c("HDL on Chow diet", "HDL on Western diet");
sizeGrWindow(12, 9);
#pdf(file = "Plots/qlt-LODscores-SexIntForHDL.pdf", width = 12, h=9);
par(mfrow = c(2,1))
par(mar = c(3.3,3,2.2, 0.2));
par(mgp = c(1.7, 0.6, 0));
# Loop over chow and western HDL
for (tr in 1:2)
{
  point.x = plotQTL(lod.sexInt[, hdlCols[tr]],
    chromo, basePair,
    col = c("black"),
    lwd = 1, ylab = "LOD",
    main = spaste(LETTERS[tr], ". Sex x genotype QTL analysis for ", prettyNames.int[tr]),
    bpBasedSpacing = TRUE, gridColor = "grey", grid.lty = 2);
  # Add grid and thresholds
  addGrid(linesPerTick = 2);
  abline(h = minLOD.int[tr], col = "red", lty = 2);
  abline(h = suggestiveLOD.int[tr], col = "green", lty = 2);
  # Mark peaks
  nPeaks = sum(suggestivePeakInfo.int[[tr]]$peakIndicator);
  peakLocs = suggestivePeakInfo.int[[tr]]$peakIndex;
  points(point.x[peakLocs], lod.sexInt[peakLocs, hdlCols[tr]], pch = 21, col = "blue", lty = 2);
  # Mark boundaries of peak regions
  fromLoc = suggestivePeakInfo.int[[tr]]$peakStartInd;
  toLoc = suggestivePeakInfo.int[[tr]]$peakEndInd;
  points(point.x[fromLoc], lod.sexInt[fromLoc, hdlCols[tr]], pch = 4, col = "darkgreen", lty = 2);
  points(point.x[toLoc], lod.sexInt[toLoc, hdlCols[tr]], pch = 4, col = "darkgreen", lty = 2);
  legendClean("topright", col = c("black", "red", "green"), lty=c(1,2,2), lwd = 1,
    legend = c("sex interaction LOD",
      "p=0.05 threshold (significant)", "p=0.2 threshold (suggestive)"))
}
# If plotting into a file, close it.
dev.off();

```

The resulting plot is shown in Figure 3. We find several significant and suggestive peaks. The LOD thresholds are lower here than in the analysis with sex as additive covariate because of the subtraction necessary to arrive at

interaction loci.

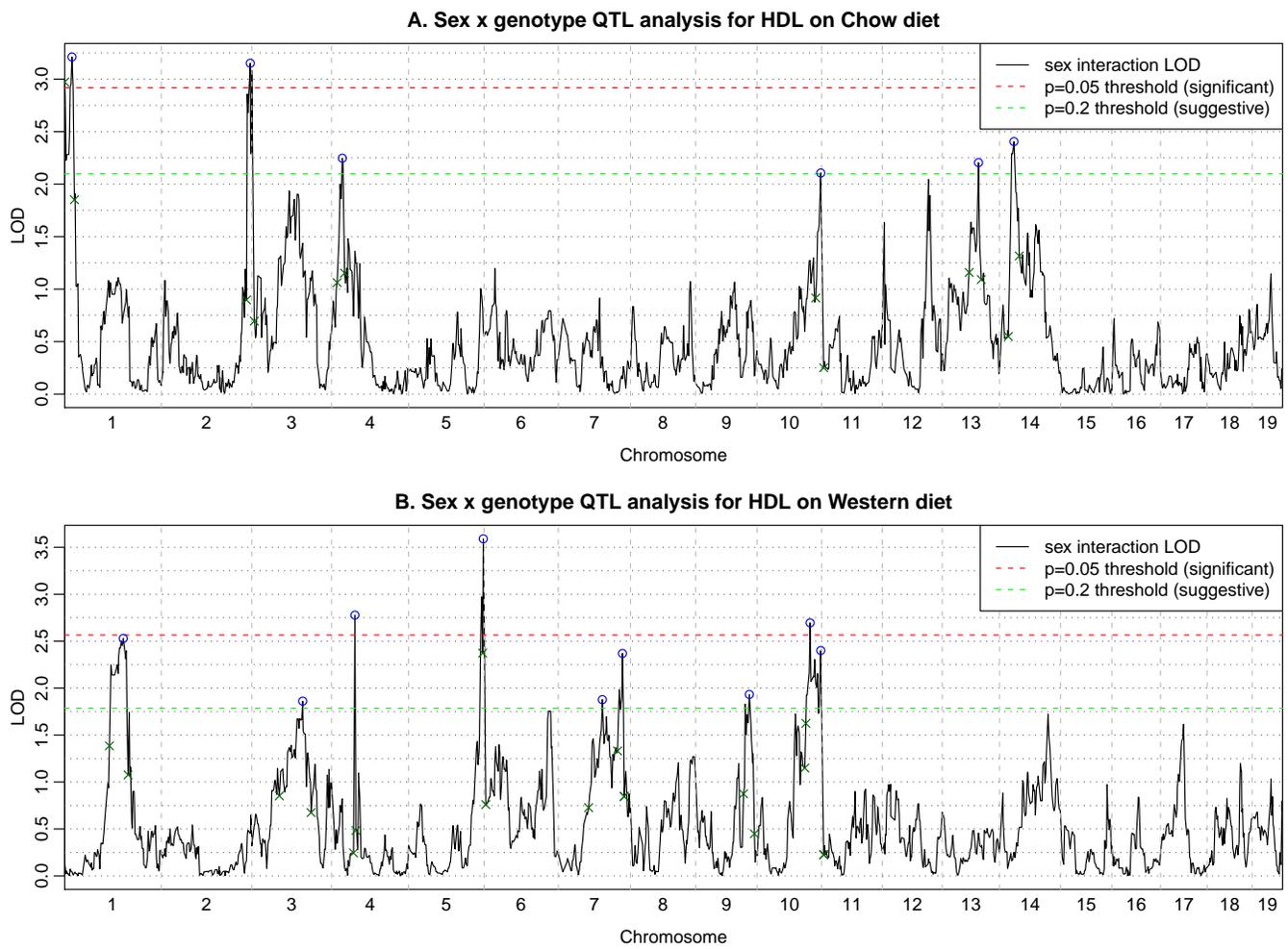


Figure 3: LOD scores of genotyped SNP markers for sex x genotype interaction for HDL on chow (A) and Western (B) diet.

We now output text tables of HDL levels by sex and genotype at the identified loci. The tables are written into plain text csv files and can be opened in standard spreadsheet software such as LibreOffice Calc or MS Excel.

```
# (Re-)define the convenience function
boxplot.stats1 = function(...)
{
  s = boxplot.stats(...);
  interval = (s$conf[2] - s$conf[1])/2
  spaste(signif(s$stats[3], 2), "(", signif(interval, 1), ")")
}
# Create tables
tables.int = list();
nPermutations = nrow(perm.sexInt);
for (tr in 1:2)
{
  nPeaks = length(suggestivePeakInfo.int[[tr]]$peakIndex);
  peakSNPnames = rownames(qtl)[suggestivePeakInfo.int[[tr]] $peakIndicator];

  hdl = traitsForQTL[, hdlCols.traits[tr]];
  tables.int[[tr]] = data.frame(matrix(0, nPeaks, 12));
  tables.int[[tr]][, 1] = peakSNPnames;
  tables.int[[tr]][, 2] = suggestivePeakInfo.int[[tr]] $peakChromo;
  tables.int[[tr]][, 3] = suggestivePeakInfo.int[[tr]] $peakBp;
  for (p in 1:nPeaks)
  {
    peakGt = geno[, -1][, suggestivePeakInfo.int[[tr]] $ peakIndex [p]];
    #means = tapply(hdl, peakGt, mean, na.rm = TRUE);
    #sd = tapply(hdl, peakGt, stdErr);
    #tables.int[[tr]][p, 4:6] = spaste(signif(means, 3), "(", signif(sd, 2), ")");

    stats = tapply(hdl, list(peakGt, sex), boxplot.stats1, do.out = FALSE);
    tables.int[[tr]][p, 4:9] = c(t(stats));
    #tables.int[[tr]][p, 7] = signif(kruskal.test(hdl, factor(peakGt))$p.value, 2);
    lod = lod.sexInt[suggestivePeakInfo.int[[tr]] $ peakIndex [p], hdlCols[tr]];
    tables.int[[tr]][p, 10] = signif(sexIntP.all[suggestivePeakInfo.int[[tr]] $ peakIndex [p], tr], 1);
    tables.int[[tr]][p, 11] = signif(lod, 2);
    lod.p = (sum(perm.sexInt[, hdlCols[tr]-2] > lod) + 1)/nPermutations
    tables.int[[tr]][p, 12] = lod.p
  }
  colnames(tables.int[[tr]]) = c("SNP", "Chr", "Bp",
    paste( rep(c("F", "M"), 3), rep(c("BB", "BC", "CC"), rep(2, 3))),
    "p-value", "LOD", "Perm. p-value");
# Save each table into a csv file.
fileName = spaste("qtl-anova-sexInt-For-", colnames(traitsForQTL)[hdlCols.traits[tr]], ".csv")
printFlush(paste("Saving table into file", fileName))
write.csv(tables.int[[tr]], file = fileName, row.names = FALSE, quote = FALSE);
}
```

The resulting tables are included as Supplementary Tables with our main manuscript.

## 5 eQTL of genes identified as consistently related to and causal for HDL

Here we generate eQTL LOD curves for 3 genes that association analysis (detailed in subsequent parts) identified as consistently associated with HDL, and causal testing identified them as causal for HDL. The three genes are *Wfdc2*, *Hdac3*, and *Acat2*. We first load the results of the eQTL analysis (detailed in a previous part) and gene annotation.

```
# Load results of eqtl analysis
load("eqtl-eqtl.RData");
# Load gene annotation
file = bzfile(description = "../../../Data-AllMouse/CXB_GeneAnnotation.csv.bz2");
annot = read.csv(file = file);
# The three genes:
genes = c("Wfdc2", "Hdac3", "Acat2");
# Check that each gene has only one probe:
sum(annot$gene_symbol==genes[1]);
sum(annot$gene_symbol==genes[2]);
sum(annot$gene_symbol==genes[3]);
genes2annot = match(genes, annot$gene_symbol)
probes = annot$sequence[genes2annot];
```

We now identify the nearest SNP and set up the plot.

```
cisSNP = nearestSNP(chromo, basePair, annot$Chromosome[genes2annot],
                    (annot$StartCoordinate[genes2annot] + annot$EndCoordinate[genes2annot])/2)
# Isolate the LOD score for the nearest marker for the three genes
eqtlCols = match(probes, colnames(eqtl));
cisEQTL = rep(0, length(genes))
for (g in 1:length(genes))
  cisEQTL[g] = eqtl[cisSNP$nearestSNP[g], eqtlCols[g]]
# Plot LOD curves of the genes and add HDL for comparison
colors = c("black", "blue", "green");
sizeGrWindow(10,8);
# Alternative plot in a file
#pdf(file = "Plots/causalGenes-eqtl-vs-HDL-qt1.pdf", wi = 10, he=8);
point.x = plotQTL(cbind(as.numeric(ql[, hdlCols[2]]), apply(eqtl[, eqtlCols], 2, as.numeric)),
                 chromo, basePair,
                 col = c("grey", colors),
                 lwd = 1, ylab = "LOD",
                 main = spaste("eQTL analysis for ", paste(genes, collapse = " and ")),
                 bpBasedSpacing = TRUE, gridColor = "grey", grid.lty = 2);
addGrid(linesPerTick = 2);
# Add markers for nearest SNP to each gene
points(x=point.x[cisSNP$nearestSNP], y = cisEQTL, col = colors, bg = colors, pch = 21)
legendClean("top", col = c(colors, "grey", colors), lty=c(1,1,1,1, NA, NA, NA),
           pch = c(NA, NA, NA, NA, 21, 21, 21),
           pt.bg = c(colors, NA, colors), bg = "grey95",
           legend = c(paste(genes, "eQTL"), "HDL QTL", paste("Physical location of", genes)));
# If plotting into a file, close it
dev.off()
```

The resulting plot is shown in Figure 4. The following code prints out the highest LOD scores for each gene, and the height of the *Acat2* peak on chromosome 10:

```
apply(eqtl[eqtlCols], 2, max)
max(eqtl[chromo==10, eqtlCols[3]])
```

### eQTL analysis for *Wfdc2* and *Hdac3* and *Acat2*

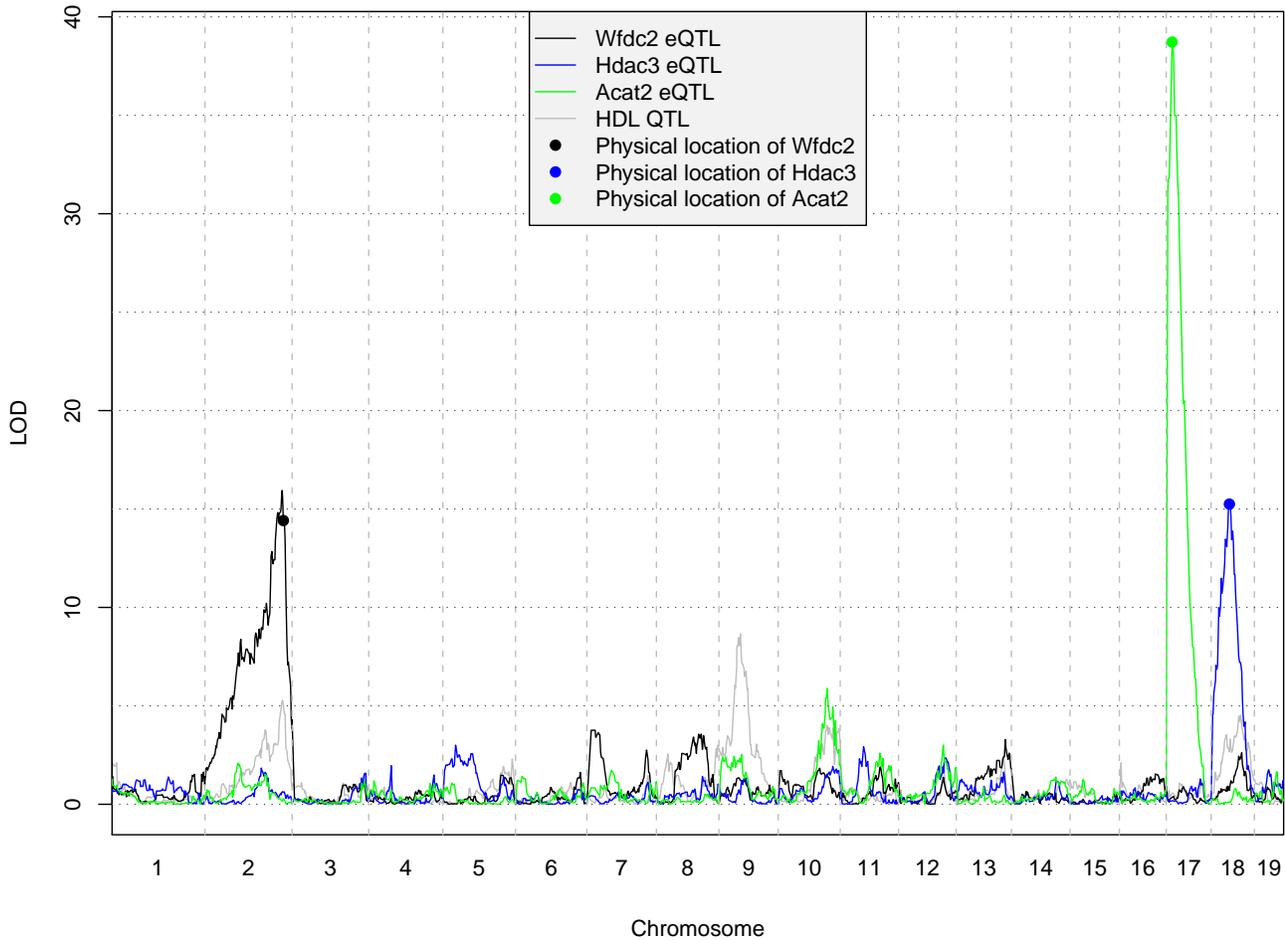


Figure 4: eQTL analysis of top 3 genes (*Wfdc2*, *Hdac3*, *Acat2*) that are consistently related to HDL and pass a causality test in the CASTxB6 data. The black, blue, and green lines show the LOD scores of *Wfdc2*, *Hdac3*, *Acat2*, respectively, at each of the nearly 1400 genotyped markers. All 3 genes have a strong cis-eQTL peak (their physical location is indicated by the filled circle on their respective LOD curves). For *Wfdc2* the cis-eQTL peak coincides with a QTL peak for HDL (grey line); for *Hdac3* the cis-eQTL peak is on the same chromosome (18) but some 30MB away from the HDL QTL peak. *Acat2* is located on chromosome 17 where HDL does not have peak, but the gene has a moderately strong trans-eQTL peak near the HDL QTL peak on chromosome 10.