Statistical analysis code for analysis of CASTxB6 F2 mouse cross

1. Data cleaning and preprocessing

Peter Langfelder

March 22, 2011

In this document we detail the first step of our analysis of the CASTxB6 cross. We load the (already normalized) expression, trait (phenotype), and genotype data, and check that all mice were labeled correctly as female and male. We then save the expression for male and female mice separately. We start by loading and formatting the expression data.

```
# Load the WGCNA library
library(WGCNA)
# Here we work with liver. To run the code for adipose, replace "Liver" with "Adipose" below.
tissue = "Liver"
# This setting is important, do not leave out
options(stringsAsFactors = FALSE);
options(width = 109)
# Load expression data and isolate actual expression columns
fileName = spaste("../../Data-AllMouse/CxB_", tissue, "_F2_mlratio_UnAdj.csv.bz2");
edata = read.csv(bzfile(fileName));
dim(edata)
# See the first few rows and columns
edata[1:5, 1:8]
# Isolate expression columns
exprCols = substring(names(edata), 1, 3)=="F2_";
geneNames = edata$transcript_id
expr0 = t(as.matrix(edata[, exprCols]));
colnames(expr0) = geneNames;
```

We next load a table of cross direction for the individual mice, and the genotypes.

```
dirInfo = read.csv("../QTL/BxC-CrossDirection-correctedByGenotypeAndSource.csv");
file = bzfile("../../Data-AllMouse/CXB_Clinical_traits.csv.bz2");
tdata = read.csv(file);
tdata = tdata[!is.na(tdata$Mice_id), ];
collectGarbage();
```

Next we load the genotypes and do some basic re-formatting and a conversion from letter coding A/H/B to numerical coding 0/1/2.

```
# Look at genotypes
file = bzfile("../../Data-AllMouse/CXB_GENOTYPES_alpha.csv.bz2");
genoA = read.csv(file);
genoA[1:5, 1:10]
genoA0 = genoA[, substring(names(genoA), 1, 3)=="F2_"];
# Check genotype counts
table(c(as.matrix(genoA0)))
# Convert genotypes to numeric
genoA1 = as.matrix(genoA0);
genoA1[genoA0=="A"] = 1;
genoA1[genoA0=="H"] = 2;
genoA1[genoA0=="B"] = 3;
genoA2 = apply(genoA1, 2, as.numeric);
# Check that the conversion makes sense
table(genoA2)
```

We next check that all mice have good quality genotype data (not too many missing genotypes). It actually turns out that 4 mice have completely missing genotypes, so we remmove them.

```
nFinite = colSums(!is.na(genoA2))
table(nFinite);
# 4 mice have no genotype data - need to take them out
goodGenotypeMice = nFinite>1400;
genoMice = colnames(genoA1)[goodGenotypeMice];

# Go back to all the other data and take the mice out from other data as well
commonMice = intersect(genoMice, intersect(intersect(rownames(expr0), dirInfo$Mice_id), tdata$Mice_id));
keepInfo = dirInfo$Mice_id %in% commonMice;
keepTraits = tdata$Mice_id %in% commonMice;
keepExpr = rownames(expr0) %in% commonMice;
keepGeno = genoMice %in% commonMice;

apply(dirInfo[keepInfo & dirInfo$direction=="CxB", c(2:4)], 2, table)

table(keepInfo)
table(keepExpr)
table(keepTraits)
table(keepGeno)
exprMF = expr0[keepExpr, ];
dirInfo1 = dirInfo[match(rownames(exprMF), dirInfo$Mice_id), ];
```

We next load gene annotation and look for highly sex-dimorphic genes

```
# Load gene annotation.
file = bzfile(description = "../../Data-AllMouse/CXB_GeneAnnotation.csv.bz2");
annot = read.csv(file = file);
# Correlation and p-value of expression with mouse sex
sexCP = corAndPvalue(exprMF, as.numeric(factor(dirInfo1$sex)));
orders = list();
orders[[1]] = order(-sexCP$cor);
orders[[2]] = order(sexCP$cor);
```

We now plot the expressions of the highest sex-dimorphic genes in males and females.

```
# Take the 12 most differentially expressed genes on both sides
nBest = 12;
sizeGrWindow(9,9)
# Alternatively: plot into a file. Make sure a subdirectory (folder) called Plots exists in the current
# directory (folder).
# pdf(file="Plots/sexDifferentiallyExpressedProbes.pdf", wi=9, he=9);
par(mfrow = c(4,6))
par(mgp = c(1.9, 0.6, 0))
par(mar = c(3.2,3.2,3,1))
for (sign in 1:2) for (g in 1:nBest)
{
  verboseBoxplot(exprMF[, orders[[sign]][g]],
     dirInfo1$sex, xlab = "sex", ylab = "expression",
     main = paste(colnames(exprMF)[orders[[sign]][g]], "\n"),
     cex.lab = 1.2, cex.axis = 1.2, cex.main = 1.2)
}
# If plotting into a file, close it
dev.off();
```

The resulting plot is shown in Figure 1. The figure indicates that one mouse labeled male may in fact be a female, since the expressions of the sex-dimorphic genes are more female-like.

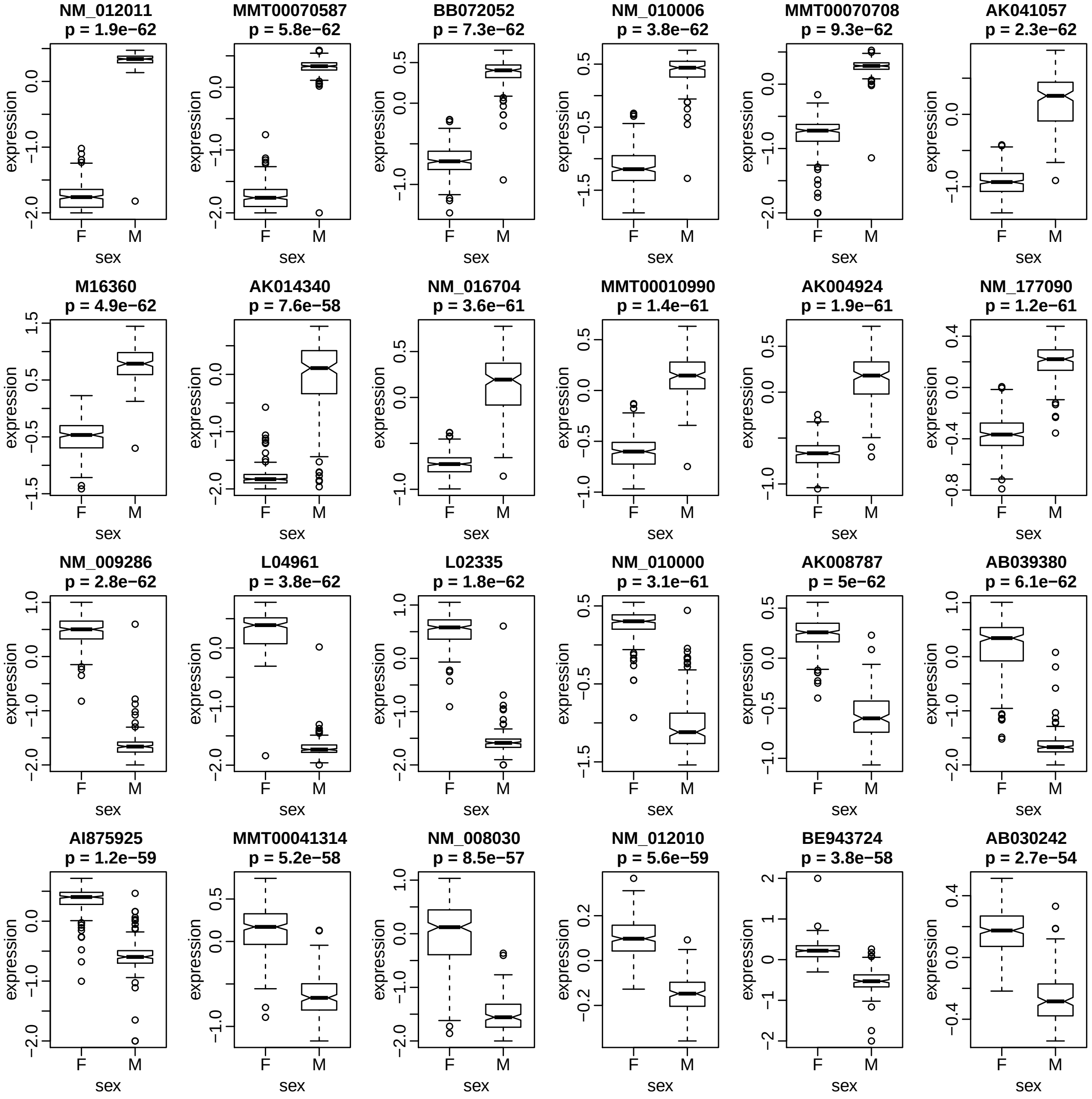

Figure 1: Expression of 12 strongest sex-dimorphic genes in each direction in male and female data. The figure indicates that one mouse labeled male may in fact be a female, since the expressions of the sex-dimorphic genes are more female-like.

We now identify the suspicious mouse sample.

```
miceLabels = dirInfo1$Mice_id;
gi = orders[[1]][1]
numSex = as.numeric(as.factor(dirInfo1$sex))
nSamples = nrow(exprMF);
suspMale = c(1:nSamples)[numSex==2][which.min(exprMF[numSex==2, gi])]
dirInfo1[suspMale, ]
# We get:
# Mice_id direction sex genotypeQuality
# 295 F2_366 CxB M good
# Find the X genotypes of the suspicious male. If it's not homzygous, it must be a female.
smGt = genoA[genoA$chro_number=="20", match(rownames(exprMF)[suspMale], names(genoA))]
table(smGt)
```

We get 44 B's and 20 H's. This proves that this ”male” is actually a female. We may as well check all other males for homozygous X chromosomes:

```
maleMice = dirInfo1$Mice_id[dirInfo1$sex=="M"];
maleGtCols = match(maleMice, names(genoA));
maleXGt = genoA[genoA$chro_number=="20", maleGtCols];
sum(maleXGt=="H", na.rm = TRUE);
table(apply(maleXGt=="H", 2, sum, na.rm = TRUE))
```

We confirm that there is one mouse with 20 H alleles (namely the one identified above), all other are homozygous. We now update the trait information and separate male and female expression data, saving each in its own .RData file.

```
dirInfoC = dirInfo;
dirInfoC$sex[dirInfo$Mice_id==rownames(exprMF)[suspMale]] = "F";
dirInfoC1 = dirInfoC[match(rownames(exprMF), dirInfoC$Mice_id), ];
# Form the expression variable
exprCxBFema = exprMF[ dirInfoC1$direction=="CxB" & dirInfoC1$sex=="F", ];
exprCxBMale = exprMF[ dirInfoC1$direction=="CxB" & dirInfoC1$sex=="M", ];
# Save the expression files
save(exprCxBFema, exprCxBMale, file = spaste("CxBOnly-", tissue, "-exprCxBFema-exprCxBMale.RData"));
# Also load the expression p-values and separate them by sex
file = bzfile(spaste("../../Data-AllMouse/CxB_", tissue, "_F2_pvalue_UnAdj.csv.bz2"));
pdata = read.csv(file);
dim(pdata)
pdata[1:5, 1:5]
exprCols = substring(names(pdata), 1, 3)=="F2_";
geneNames = pdata$transcript_id
p0 = t(as.matrix(pdata[, exprCols]));
colnames(p0) = geneNames;
all.equal(rownames(p0), rownames(expr0))
pMF = p0[keepExpr, ];
pCxBFema = pMF[ dirInfoC1$direction=="CxB" & dirInfoC1$sex=="F", ];
pCxBMale = pMF[ dirInfoC1$direction=="CxB" & dirInfoC1$sex=="M", ];
save(pCxBFema, pCxBMale, file = spaste("CxBOnly-", tissue, "-pCxBFema-pCxBMale.RData");
```

We now cluster the samples to check for possible microarray outliers. We perform the same procedure for the female and male data; we start with the females.

```
# Cluster samples using Euclidean distance and hierarchical clustering
dst = dist(exprCxBFema);
tree = flashClust(dst, method = "a");
if (tissue=="Liver") cutHeight = 65 else cutHeight = 35;
# Plot the tree
sizeGrWindow(13,9);
#pdf(file = spaste("Plots/", tissue, "-Female-outlierRemoval-dstTree.pdf"), w=18, h=9)
plot(tree, sub = "", xlab = "",
    main = paste("Female", tissue, "samples clustered on Euclidean distance"));
abline(h = cutHeight, col = "red");
#If plotting into a file, close it
dev.off();
# Cut the tree
labs = cutreeStatic(tree, cutHeight = cutHeight, minSize = 20);
keep = labs==1
# Restrict data to kept samples
keepMice = rownames(exprCxBFema)[keep];
removeMice = rownames(exprCxBFema)[!keep];
exprFemaOR = exprCxBFema[keep, ];
pValFemaOR = pCxBFema[keep, ];
# Save the cleaned data for further use
save(exprFemaOR, pValFemaOR,
    file = spaste("CxBOnly-", tissue, "-outliersRemoved-exprFemaOR-pValFemaOR.RData"));
# Save the list of samples that will be used for subsequent analyses
write.table(data.frame(keepMice),
           file = spaste("CxBOnly-", tissue, "-Female-outliersRemoved-retainedSamples.txt"), quote = FALSE,
           col.names=FALSE, row.names = FALSE)
```

The sample clustering tree and the cut line are shown in Figure 2.

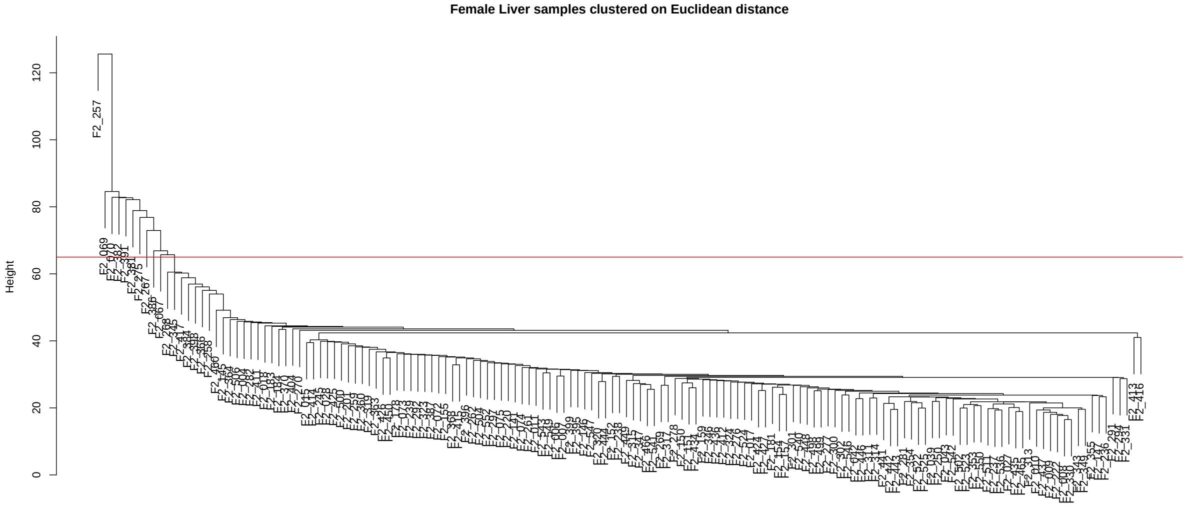

Figure 2: Hiearchical clustering tree of female samples. The red line indicates the cut above which samples are considered outliers.

We now perform the same procedure on the males.

```
# Outlier removal for male samples. Will do the same thing.
dstM = dist(exprCxBMale);
treeM = flashClust(dstM, method = "a");
if (tissue=="Liver") cutHeightM = 60 else cutHeightM = 35;
sizeGrWindow(13,9);
#pdf(file = spaste("Plots/", tissue, "-Male-outlierRemoval-dstTree.pdf"), w=16, h=9)
plot(treeM, sub = "", xlab = "",
    main = spaste("Male ", tissue, " samples clustered on Euclidean distance"));
abline(h = cutHeightM, col = "red");
#If plotting into a file, close it
dev.off();
# Cut the tree
labs = cutreeStatic(treeM, cutHeight = cutHeightM, minSize = 20);
keep = labs==1
# Restrict data to kept samples
keepMice = rownames(exprCxBMale)[keep];
removeMice = rownames(exprCxBMale)[!keep];
exprMaleOR = exprCxBMale[keep, ];
pValMaleOR = pCxBMale[keep, ];
# Save the cleaned data for further use
save(exprMaleOR, pValMaleOR,
    file = spaste("CxBOnly-", tissue, "-outliersRemoved-exprMaleOR-pValMaleOR.RData"));
# Save the list of samples that will be used for subsequent analyses
write.table(data.frame(keepMice), quote = FALSE,
           file = spaste("CxBOnly", tissue, "-Male-outliersRemoved-retainedSamples.txt",
           col.names=FALSE, row.names = FALSE)
```

The male sample clustering tree and the cut line are shown in Figure 3.

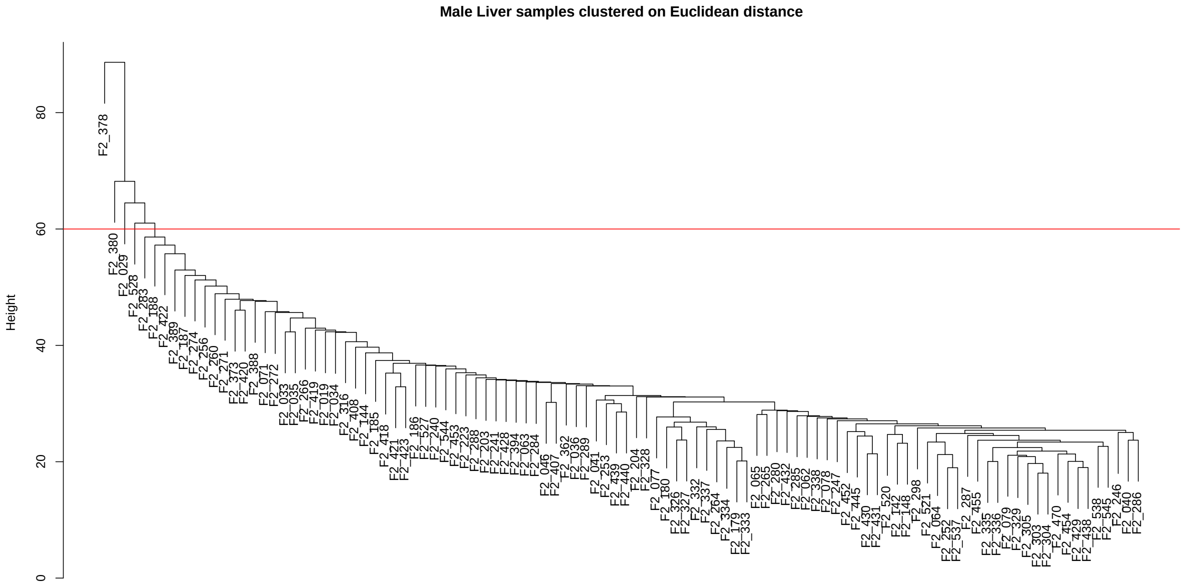

Figure 3: Hiearchical clustering tree of male samples. The red line indicates the cut above which samples are considered outliers.