Statistical analysis code for analysis of CASTxB6 F2 mouse cross

4. Standard analysis: associations of individual genes with HDL and other traits

Peter Langfelder

May 5, 2011

Contents

1 Preliminiaries

In this document we detail the standard analysis of association of individual genes with HDL and other traits. We use the robust biweight midcorrelation to quantify the associations. We start by setting up the R session and loading the pre-processed expression data.

```
# Load the WGCNA library
library(WGCNA)
# This setting is important, do not leave out
options(stringsAsFactors = FALSE);
options(width = 109)
# Set up number of data sets and convenience labels
nSets = 2;
setLabels = c("Female Liver", "Female Adipose");
shortLabels = c("Liver", "Adipose");
shortshortLabels = c("L", "A");
# Adjust the paths in the following to point to the expression files saved in part 1 (Data
Preprocessing) of the analysis
files = c("../CxBOnly-Liver-outliersRemoved-exprFemaOR-pValFemaOR.RData",
         "../CxBOnly-Adipose-outliersRemoved-exprFemaOR-pValFemaOR.RData");
express = list();
for (set in 1:nSets)
{
  x = load(file = files[set]);
  express[[set]] = list(data = exprFemaOR);
}
expr = express;
rm(express);
collectGarbage();
# Basic data statistics
exprSize = checkSets(expr);
nSamples = exprSize$nSamples;
collectGarbage()
```

We next load the clinical traits and restrict the traits to only those mice that have a valid expression sample (independently in each tissue).

```
rawTr = read.csv(file = bzfile("../../../Data-AllMouse/CXB_Clinical_traits.csv.bz2"))
```

```
numTraitInd = c(15:46, 48)
numTraits = vector(mode = "list", length = nSets);
# The following is relative to numTraitInd
selTraitInd = c(5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33)
selTraits = vector(mode = "list", length = nSets);
# restrict traits to samples with valid expression data
for (set in 1:nSets)
{
  mice = rownames(expr[[set]]$data);
  expr2tr = match(mice, rawTr$Mice_id);
  temp = rawTr[expr2tr, numTraitInd]
  rownames(temp) = rawTr$Mice_id[expr2tr];
  numTraits[[set]] = list(data = as.matrix(temp));
  selTraits[[set]] = list(data = as.matrix(temp[, selTraitInd]));
}
collectGarbage();
nSelTraits = length(selTraitInd)
tcInd = match("e_tc_mgdl", colnames(selTraits[[1]]$data));
hdlInd = match("e_hdl_mgdl", colnames(selTraits[[1]]$data));
```

There are a few obvious outliers in the data, such as a "monster mouse" with length nearly 30cm (one foot). We remove such obvious outliers.

```
# There's a length outlier:
for (set in 1:nSets)
{
   suspicious = numTraits[[set]]$data[,"length_cm"] > 20;
   numTraits[[set]]$data[suspicious,"length_cm"] = NA;
   selTraits[[set]]$data[suspicious,"length_cm"] = NA;
}
# ...and an e_fat outlier:
for (set in 1:nSets)
{
   suspicious = numTraits[[set]]$data[,"efat_g"] > 20;
   numTraits[[set]]$data[suspicious,"efat_g"] = NA;
   selTraits[[set]]$data[suspicious,"efat_g"] = NA;
}
```

2 Association of genes with traits

Here we calculate the association of all probes with the clinical traits and obtain the corresponding local estimates of False Discovery Rate (FDR).

```
bc = list();
q = list();
qvalue1 = function(...) { qvalue(...)$qvalues }
for (set in 1:nSets)
{
  bc[[set]] = bicorAndPvalue(expr[[set]]$data, selTraits[[set]]$data);
  bc[[set]]$q = apply(bc[[set]]$p, 2, qvalue1);
}
collectGarbage();
```

We next connect the probes to genes using the probe annotation file, and write out large tables of probe-trait associations in each tissue. Please adapt the path in the follwing code to point to the directory housing the annotation file.

```
file = bzfile(description = "../../../Data-AllMouse/CXB_GeneAnnotation.csv.bz2");
```

```
annot = read.csv(file = file);
nGenes = checkSets(expr)$nGenes;
nTraits = checkSets(selTraits)$nGenes;
GSTables = list();
for (set in 1:nSets)
{
  probes = colnames(expr[[set]]$data);
  expr2annot = match(probes, annot$sequence);
  LLID = annot$LocusLinkID[expr2annot];
  gene = annot$gene_symbol
  # produce and format the combined table
  res = rbind(bc[[set]]$bicor, bc[[set]]$p, bc[[set]]$q);
  dim(res) = c(nGenes, 3*nTraits);
  colnames(res) = spaste(c("cor.", "p.", "q."), rep(colnames(selTraits[[set]]$data), rep(3, nTraits)))
  GSTables[[set]] = data.frame(probe = probes, locusLinkID = LLID, geneSymbol = gene, res);
  # Write out the table.
  write.csv(GSTables[[set]], row.names = FALSE, quote = TRUE,
            file = bzfile(spaste("CxBOnly-Female-", shortLabels[set], "-marginalAnalysis.csv.bz2")));
}
```

We next print out numbers of probes that pass the thresholds of 0.05 for p, q, and Bonferroni corrected p-value. We also create indicator variables indicating for each probe whether it passes each of the thresholds. We use these indicators for the calculation of GO enrichment below.

```
collectGarbage();
hdlLabels = matrix(0, nGenes, 3 * nSets)
for (set in 1:nSets)
{
  pos = bc[[set]]$bicor[, hdlInd] > 0;
  printFlush(paste("=============", shortLabels[set], "======================"));
  printFlush(paste("Total # p<0.05:", sum(bc[[set]]$p[, hdlInd] < 0.05),
                   ", Total # q < 0.05:", sum(bc[[set]]$q[, hdlInd] < 0.05),
                   ", Total # p.Bonf < 0.05:", sum(bc[[set]]$p[, hdlInd] < 0.05/nGenes)));
  printFlush(paste("Positive # p<0.05:", sum(bc[[set]]$p[, hdlInd] < 0.05 & pos),
                   ", Positive # q < 0.05:", sum(bc[[set]]$q[, hdlInd] < 0.05 & pos),
                   ", Positive # p.Bonf < 0.05:", sum(bc[[set]]$p[, hdlInd] < 0.05/nGenes & pos)));
  printFlush(paste("Negative # p<0.05:", sum(bc[[set]]$p[, hdlInd] < 0.05 & !pos),
                   ", Negative # q < 0.05:", sum(bc[[set]]$q[, hdlInd] < 0.05 & !pos),
                   ", Negative # p.Bonf < 0.05:", sum(bc[[set]]$p[, hdlInd] < 0.05/nGenes & !pos)));

  hdlLabels[bc[[set]]$p[, hdlInd] < 0.05/nGenes, 3*(set-1) + 1] = 1;
  hdlLabels[bc[[set]]$p[, hdlInd] < 0.05/nGenes & pos, 3*(set-1) + 2] = 1;
  hdlLabels[bc[[set]]$p[, hdlInd] < 0.05/nGenes & !pos, 3*(set-1) + 3] = 1;
}
```

3 GO enrichment analysis of top genes associated with HDL

Here we calculate GO enrichment of the top genes associated with HDL. We calculate the enrichment for positively associated genes, negatively related genes, and combined positvely and negatively ("all") associated genes. We save the results in two formats, one with longer with more information and one shorter and more concise.

```
for (set in 1:nSets)
{
   # Get the LocusLinkIDs (Entrez codes) for genes represented by the probes
   expr2annot = match(colnames(expr[[set]]$data), annot$sequence);
   LLID = annot$LocusLinkID[expr2annot];
   table(is.na(LLID))
   fin = !is.na(LLID);
```

```
  # Restrict to proes with valid Entrez
  finLLID = LLID[fin];
  finLabels = hdlLabels[fin, c( (3*set-2):(3*set))];
  # Call the GO enrichment function. This may take some time, usually about 5-10 minutes.
  print(system.time ( {
       bt = GOenrichmentAnalysis(finLabels, finLLID, organism = "mouse",
                                 nBest = 20, nBiggest = 0,
                                 leaveOutLabel = 0);
             } ));
  # Isolate a summary enrichment table
  bt$setResults[[1]]$bestPTerms[[4]]$enrichment[, "module"] = "All";
  bt$setResults[[2]]$bestPTerms[[4]]$enrichment[, "module"] = "PositiveCor";
  bt$setResults[[3]]$bestPTerms[[4]]$enrichment[, "module"] = "NegativeCor";
  res = rbind(bt$setResults[[1]]$bestPTerms[[4]]$enrichment,
             bt$setResults[[2]]$bestPTerms[[4]]$enrichment,
             bt$setResults[[3]]$bestPTerms[[4]]$enrichment);
  # Write out the full results
  write.table(res, file = spaste("CxBOnly-Female-", shortLabels[[set]], "-HDLAssociated-GOenrichment.txt"),
             row.names = FALSE, sep = "\t", quote = FALSE);
  # Shorten the results and write a shorter version
  tab2 = res[, c(1, 3, 4, 6, 8, 11, 12, 13)];
  colnames(tab2)[colnames(tab2)=="module"] = "AssociationSign";
  colnames(tab2)[colnames(tab2)=="bkgrModSize"] = "nGenes";
  colnames(tab2)[colnames(tab2)=="fracOfBkgrModSize"] = "fractionInTerm";
  tab2[ tab2[, 1]=="All", 1] = "Both";
  write.table(tab2, file = spaste("CxBOnly-Female-", shortLabels[[set]],
                                  "-HDLAssociated-GOenrichment-shortened.txt"),
               row.names = FALSE, sep = "\t", quote = FALSE);
  # Print the results on-screen as well.
  res2 = res[, c(1, 2, 4, 6, 8, 12, 13)];
  res2[, c(3, 4)] = signif(apply(res2[, c(3,4)], 2, as.numeric), 2)
  rownames(res2) = NULL
  names(res2) = c("Mod", "Size", "Rnk", "p.Bonf", "fracModSz", "ont", "termName");
  terms = res2$termName;
  sterms = substring(terms, 1, 55);
  res2$termName = sterms;
  options(width = 109);
  print(res2)
}
```