Supplementary Appendix

This web page contains supplementary material, including supplementary methods and supplementary reports describing all data analyses, for the manuscript entitled *"Molecular Biomarkers of Residual Disease after Surgical Debulking of High-Grade Serous Ovarian Cancer"* by the ovarian cancer working group at MD Anderson Cancer Center.

All analyses were performed by Keith A. Baggerly, Shelley M. Herbrich, Susan L. Tucker, or Anna Unruh.

This page was last updated on Friday, March 28, 2013. The files posted here will not be changed after publication, allowing the web site to serve as permanent documentation of our analysis. Any changes will be posted on a separate page designed for addenda, errata, corrigenda and other adjustments.

Our analyses make use of raw data (e.g. Affymetrix CEL files) from a variety of sources. These files are not reproduced here, just links to where the data can be obtained.

- TCGA Ovarian Cancer Affymetrix CEL Files. We used version 1007 throughout; only the mage-tab folder appears to have been updated (to version 1008) as of May 7, 2013.
- TCGA Ovarian Cancer Clinical Data. These files are updated quite regularly, and we do not know where earlier versions can be found. The values we derived from the "clinical_patient_ov.txt" file for RD status, etc, are built in to the RData file we produced (available below).
- Tothill et al CEL Files and Clinical Information.
- Bonome et al CEL Files. Clinical information (e.g., optimal/suboptimal surgery outcome) is available from the individual pages, which are easily parseable using the GEOquery R package (see the "assemblingCCLEClinical" report below).
- CCLE CEL Files from the initial publication. Primary site and histology information are given in the component pages, and assembled using the GEOquery R package (see the "assemblingCCLEClinical" report below).

Supplementary Methods:

Data for validation of biomarker datasets. The first of these was from the study of Bonome et al. [11]. We downloaded CEL files (Affymetrix U133A arrays, N=195; 185 tumor samples and 10 normal ovary) from the Gene Expression Omnibus (GEO; GSE26712) on September 10, 2012. The samples in this study were lasercapture microdissected, and the surgical outcome recorded as optimal or suboptimal. These data were used to assess whether qualitative differences in gene expression observed in the first two datasets (TCGA and Tothill et al.) were present here as well. The other dataset was from the Cancer Cell Line Encyclopedia (CCLE) [12]. We downloaded CEL files (Affymetrix U133+2 arrays, N=917) described in the initial CCLE publication from GEO (GSE36133) on September 14, 2012. These data were used to determine whether differences in gene expression seen in tumor samples are present in ovarian cancer cell lines.

Quantitative RT-PCR analysis. Total RNA was extracted from the tumor tissues using the TRIzol® extraction method. RNA was then quantified using a nanodrop method and the 260/280 ratios were also checked to determine quality. RNA (1µg/sample) was reverse transcribed into cDNA using the Verso cDNA kit (Thermo Scientific, West Palm Beach, FL) according to the manufacturer's protocol.

qRT-PCR was performed on a 7500 PCR system (Applied Biosystems, Warrington, UK) using 1µL of cDNA for each sample. SYBR green (Applied Biosystems) was used to detect the products and 20pmoles of primer were used for the reaction. All reactions were carried out with 20µL of reaction mix and were performed in triplicate. We used the following primers: For *FABP4*, 5'-TGATGATCATGTTAGGTTTGGC-3' (forward) and 5'-TGGAAACTTGTCTCCAGTGAA-3' (reverse). For *ADH1B*, 5'- AGGGTAGAGGAGGCTGAAGA-3' (forward), 5'-

ACCTGCTTCACTCTGGGAAA-3' (reverse). The PCR reactions were run under the following conditions: 50°C for 2 minutes, 95°C for 15 minutes, followed by 40 cycles at 95°C for 1 minute each. All reactions were analyzed with the 7500 Applied Biosystems PCR software (v.2.0.5). The cycle threshold (Ct) values of the target genes were initially normalized to the Ct values of 18S rRNA and melt curves were checked to determine the specificity of the reactions.

Since initial examination of the qRT-PCR results showed that some gene-specific fluorescence thresholds automatically selected by the commercial PCR software were artificially low, resulting in overestimation of Ct values and underestimation of the amount of target RNA (see Supplementary Report: Problems with default PCR quantifications), we quantified the PCR samples with initial concentration estimates using the "window of linearity" method [S1]. This approach provides a simple, well-specific summary of initial amount that is independent of efficiency assumptions.

Supplementary References:

S1. Ruijter JM, Ramakers C, Hoogars WM, Karlen Y, Bakker O, van den Hoff MJ, Moorman AF. Amplification efficiency: linking baseline and bias in the analysis of quantitative PCR data. Nucleic Acids Res 2009;37:e45.

Supplementary Table 1: Probesets (N=47) and associated genes (N=38) having consistent differences in expression between residual disease (RD) and No-RD patients in the TCGA and Tothill data sets at a 10% false discovery rate in each data set.

Gene	Probeset
ADAM12	213790_at
ADH1B	209612_s_at
ADH1B	209613_s_at
ADIPOQ	207175_at
ALDH1A3	203180_at
ALDH5A1	203609_s_at
AQP1	209047_at
BCHE	205433_at
COL11A1	37892_at
COL16A1	204345_at
COL3A1	201852_x_at
COL5A1	203325_s_at
COL6A2	213290_at
COL8A1	214587_at
CRISPLD2	221541_at
CXCL12	203666_at
CXCL12	209687_at
CYR61	201289_at
DCN	201893_x_at
DCN	209335_at
DCN	211813_x_at
DCN	211896_s_at
ETV1	221911_at

FABP4	203980_at
FAP	209955_s_at
GADD45B	207574_s_at
GADD45B	209304_x_at
GADD45B	209305_s_at
GFPT2	205100_at
GREM1	218468_s_at
GREM1	218469_at
KCNE4	222379_at
LUM	201744_s_at
NBL1	201621_at
NBL1	37005_at
NFYA	204107_at
OMD	205907_s_at
PDGFD	219304_s_at
PDLIM3	209621_s_at
PDPN	221898_at
POLR1C	207515_s_at
PTGIS	208131_s_at
SVEP1	213247_at
TIMP3	201150_s_at
VGLL3	220327_at
VSIG4	204787_at
XYLT1	213725_x_at

Supplementary Reports:

Here is a list of the supplementary reports, which are provided in HTML format. These reports were produced using knitr, markdown and RStudio.

- Supplementary reports:
  - Assembling TCGA Expression Data, Assembling TCGA Clinical Data
  - Assembling Tothill Expression Data, Assembling Tothill Clinical Data
  - Assembling Bonome Expression Data, Assembling Bonome Clinical Data
  - Assembling CCLE Expression Data, Assembling CCLE Clinical Data
  - Filtering TCGA Samples
  - Filtering Tothill Samples
  - Overall Survival Curves in TCGA and Tothill, Showing Effects of RD
  - Identification of Genes Associated with RD in TCGA and Tothill (Biomarker Discovery)
  - Plots of Top 47 Probesets
  - Plots of *FABP4* versus *ADH1B* Expression for All Array Datasets
  - Problems with Default PCR Quantifications
  - PCR Quantifications for Validation Data
  - Showing Levels of *FABP4* and *ADH1B* Expression are Higher in Omental Samples than in Primary Ovarian Samples

    - Power Calculations for the Validation Study
    - Biomarker Validation based on PCR Results in the Validation Cohort
    - Script for Producing Figure 5
    - Plot of RD Incidence by *FABP4* Expression in the Validation Cohort
    - Correlations between *FABP4* and *ADH1B* Expression and Protein Levels Measured by RPPA in TCGA
- Zip files:
    - Rmd source files for all reports
    - RDataObjects

Our analysis source code relies on a number of software programs and auxiliary packages; we provide scripts, not stand-alone executables. Detailed descriptions of the packages (with version numbers) are listed in the individual reports. The pieces of software required to execute the source code can be obtained from the following locations:

- The main program and several package libraries from the R environment for statistical programming.
- Several package libraries from BioConductor.

Assembling an RMA Quantification Matrix for the TCGA Ovarian Data

Keith A. Baggerly

1 Executive Summary

1.1 Introduction

We want to produce an RData file with a matrix of robust multi-array average (RMA) expression values for the TCGA ovarian cancer samples profiled with Affymetrix HT_HG-U133A arrays.

1.2 Methods

We acquired the 14 gzipped tarballs containing the individual Level 1 data files (CEL files) from the TCGA open-access http page, https://tcga-data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/ov/cgcc/broad.mit.edu/ht_hg-u133a/transcriptome/, on September 2, 2012. According to the page, these files were last updated on June 24, 2011. At the same time, we also acquired the gzipped tarball with the MageTab (annotation) data.

Explicit lists of the batch and version numbers of the tarballs used are given in the text below.

We load the individual CEL files by folder, recording the folder (batch) information as we go, and use justRMA to compute RMA fits for the set. We extract the expression matrix and use the mapping information from the sample and data relationship format (sdrf) file in the MageTab folder to update the column names.

1.3 Results

We save tcgaSampleInfo, tcgaDataDirs, tcgaFiles, and tcgaExpression to the RData file “tcgaExpression.RData”.

In passing, we note that our quantifications match the reported TCGA Level 2 quantifications quite well (essentially to within roundoff error) when we restrict justRMA to the 594 (of 598) CEL files that are “used in analysis” per the sdrf file.

2 Libraries

We first load the libraries we will use in this report.

```
library(affy)
library(hthgu133acdf)
```

3 Specifying the Raw Data Location

Here, we specify the location of the data we acquired from TCGA on our local system. You will need to acquire these files and adjust this path before running this report yourself.

```
pathToTCGAData <- file.path("RawData", "TCGA", "CEL_Files")
```

4 The SDRF file from the MageTab Folder

We now load the sample description (sdrf) information.

```
sdrf <- read.table(file.path(pathToTCGAData, "broad.mit.edu_OV.HT_HG-U133A.mage-tab.1.1007.0",
    "broad.mit.edu_OV.HT_HG-U133A.sdrf.txt"), header = TRUE, sep = "\t")
dim(sdrf)
```

```
## [1] 598  33
```

```
sdrf[1, ]
```

```
##                  Extract.Name                          Protocol.REF
## 1 TCGA-29-1704-02A-01R-0808-01 broad.mit.edu:labeling:HT_HG-U133A:01
##          Labeled.Extract.Name  Label Term.Source.REF
## 1 TCGA-29-1704-02A-01R-0808-01 biotin   MGED Ontology
##                            Protocol.REF.1
## 1 broad.mit.edu:hybridization:HT_HG-U133A:01
##                                  Hybridization.Name
## 1 URGED_p_TCGA_Pop_May2011_HT_HG-U133A_96-HTA_D05_786990
##                               Array.Design.REF Term.Source.REF.1
## 1 Affymetrix.com:PhysicalArrayDesign:HT_HG-U133A          caArray
##                                 Protocol.REF.2
## 1 broad.mit.edu:image_acquisition:HT_HG-U133A:01
##                     Scan.Name
## 1 TCGA-29-1704-02A-01R-0808-01
##                                          Array.Data.File
## 1 URGED_p_TCGA_Pop_May2011_HT_HG-U133A_96-HTA_D05_786990.CEL
##                     Comment..TCGA.Archive.Name. Comment..TCGA.Data.Type.
## 1 broad.mit.edu_OV.HT_HG-U133A.Level_1.27.1007.0          Expression-Gene
##   Comment..TCGA.Data.Level. Comment..TCGA.Include.for.Analysis.
## 1                  Level 1                                   no
##                     Comment..md5. Protocol.REF.3 Normalization.Name
## 1 d47d925a83a2749fb09448736afa8a4b             ->                 ->
##   Derived.Array.Data.Matrix.File Comment..TCGA.Archive.Name..1
## 1                             ->                            ->
##   Comment..TCGA.Data.Type..1 Comment..TCGA.Data.Level..1
## 1                         ->                          ->
##   Comment..TCGA.Include.for.Analysis..1 Comment..md5..1 Protocol.REF.4
## 1                                    ->              ->              ->
##   Normalization.Name.1 Derived.Array.Data.Matrix.File.1
## 1                   ->                              ->
##   Comment..TCGA.Archive.Name..2 Comment..TCGA.Data.Type..2
## 1                           ->                          ->
##   Comment..TCGA.Data.Level..2 Comment..TCGA.Include.for.Analysis..2
## 1                          ->                                   ->
##   Comment..md5..2
## 1              ->
```

```
length(unique(sdrf[, 1]))
```

```
## [1] 597
```

There were 598 arrays run, but only 597 distinct samples; one sample was run twice. We now check which sample this was, and whether we care.

```
which(sdrf[, 1] == sdrf[which(duplicated(sdrf[, 1])), 1])
```

```
## [1]   1 207
```

```
as.character(sdrf[which(sdrf[, 1] == sdrf[which(duplicated(sdrf[, 1])), 1]),
    "Hybridization.Name"])
```

```
## [1] "URGED_p_TCGA_Pop_May2011_HT_HG-U133A_96-HTA_D05_786990"
## [2] "TARRE_p_MultiPlate_TCGA_SS_MA_Ref_HT_HG-U133A_96-HTA_F05_586106"
```

```
which(sdrf[, "Comment..TCGA.Include.for.Analysis."] == "no")
```

```
## [1]   1 157 171 207
```

```
as.character(sdrf[which(sdrf[, "Comment..TCGA.Include.for.Analysis."] == "no"),
    "Hybridization.Name"])
```

```
## [1] "URGED_p_TCGA_Pop_May2011_HT_HG-U133A_96-HTA_D05_786990"
## [2] "FEAST_p_TCGA_B20_21_Expression_HT_HG-U133A_96-HTA_F04_516474"
## [3] "FEAST_p_TCGA_B20_21_Expression_HT_HG-U133A_96-HTA_G06_516372"
## [4] "TARRE_p_MultiPlate_TCGA_SS_MA_Ref_HT_HG-U133A_96-HTA_F05_586106"
```

As it happens, four of the CEL files (including the two where the same sample was run) are excluded from later analyses. Spot checking the files in Batch 27 (where both of the duplicates were) shows these samples are present in the Level 1 but not in the Level 2 data. We restrict our quantification to just the 594 samples used.

5 Quantifying The CEL Files

5.1 Identifying the Data Directories

Next, we turn to the individual Level 1 data files (CEL files). These are stored in 14 folders, corresponding to run batches. Here, we identify the folders and sort them in rough chronological order. Since this is a “freeze” of what we use to generate our RData file, we hardcode the directories used.

```
tcgaDataDirs <- c("broad.mit.edu_OV.HT_HG-U133A.Level_1.9.1007.0", "broad.mit.edu_OV.HT_HG-
U133A.Level_1.11.1007.0",
    "broad.mit.edu_OV.HT_HG-U133A.Level_1.12.1007.0", "broad.mit.edu_OV.HT_HG-
U133A.Level_1.13.1007.0",
    "broad.mit.edu_OV.HT_HG-U133A.Level_1.14.1007.0", "broad.mit.edu_OV.HT_HG-
U133A.Level_1.15.1007.0",
    "broad.mit.edu_OV.HT_HG-U133A.Level_1.17.1007.0", "broad.mit.edu_OV.HT_HG-
U133A.Level_1.18.1007.0",
    "broad.mit.edu_OV.HT_HG-U133A.Level_1.19.1007.0", "broad.mit.edu_OV.HT_HG-
U133A.Level_1.21.1007.0",
    "broad.mit.edu_OV.HT_HG-U133A.Level_1.22.1007.0", "broad.mit.edu_OV.HT_HG-
U133A.Level_1.24.1007.0",
    "broad.mit.edu_OV.HT_HG-U133A.Level_1.27.1007.0", "broad.mit.edu_OV.HT_HG-
U133A.Level_1.40.1007.0")

nBatches <- length(tcgaDataDirs)

batchNumber <- strsplit(tcgaDataDirs, "\\.")
batchNumber <- unlist(lapply(batchNumber, function(x) {
    x[length(x) - 2]
}))
batchNumber <- as.numeric(batchNumber)
batchNumber
```

```
##  [1]  9 11 12 13 14 15 17 18 19 21 22 24 27 40
```

```
tcgaDataDirs <- tcgaDataDirs[order(batchNumber)]
tcgaDataDirs
```

```
##  [1] "broad.mit.edu_OV.HT_HG-U133A.Level_1.9.1007.0"
##  [2] "broad.mit.edu_OV.HT_HG-U133A.Level_1.11.1007.0"
##  [3] "broad.mit.edu_OV.HT_HG-U133A.Level_1.12.1007.0"
##  [4] "broad.mit.edu_OV.HT_HG-U133A.Level_1.13.1007.0"
##  [5] "broad.mit.edu_OV.HT_HG-U133A.Level_1.14.1007.0"
##  [6] "broad.mit.edu_OV.HT_HG-U133A.Level_1.15.1007.0"
##  [7] "broad.mit.edu_OV.HT_HG-U133A.Level_1.17.1007.0"
##  [8] "broad.mit.edu_OV.HT_HG-U133A.Level_1.18.1007.0"
##  [9] "broad.mit.edu_OV.HT_HG-U133A.Level_1.19.1007.0"
## [10] "broad.mit.edu_OV.HT_HG-U133A.Level_1.21.1007.0"
```

```
## [11] "broad.mit.edu_OV.HT_HG-U133A.Level_1.22.1007.0"
## [12] "broad.mit.edu_OV.HT_HG-U133A.Level_1.24.1007.0"
## [13] "broad.mit.edu_OV.HT_HG-U133A.Level_1.27.1007.0"
## [14] "broad.mit.edu_OV.HT_HG-U133A.Level_1.40.1007.0"
```

```
batchNumber <- sort(batchNumber)
```

5.2 Grabbing the CEL File Names

Next, we get all of the individual filenames contained in each folder.

```
tcgaFiles <- vector("list", length(batchNumber))
names(tcgaFiles) <- paste("Batch", batchNumber, sep = ".")
for (i1 in 1:nBatches) {
    tcgaFiles[[i1]] <- dir(file.path(pathToTCGAData, tcgaDataDirs[i1]), pattern = "CEL$")
}
unlist(lapply(tcgaFiles, length))
```

```
##  Batch.9 Batch.11 Batch.12 Batch.13 Batch.14 Batch.15 Batch.17 Batch.18
##       45       37       46       47       46       22       47       47
## Batch.19 Batch.21 Batch.22 Batch.24 Batch.27 Batch.40
##       47       46       47       46       24       51
```

```
nFiles <- sum(unlist(lapply(tcgaFiles, length)))
nFiles
```

```
## [1] 598
```

```
sampleBatch <- rep(batchNumber, times = unlist(lapply(tcgaFiles, length)))
```

There are 598 filenames, but (as noted above) these include samples not used in the analyses.

We list out the full paths in a character vector for feeding to justRMA.

```
celFileNames <- unlist(tcgaFiles)
celFileDirs <- rep(tcgaDataDirs, times = unlist(lapply(tcgaFiles, length)))
celFilePaths <- file.path(pathToTCGAData, celFileDirs, celFileNames)

unusedCELs <- as.character(sdrf[sdrf[, "Comment..TCGA.Include.for.Analysis."] ==
    "no", "Array.Data.File"])

celFilePathsReduced <- celFilePaths[-match(unusedCELs, celFileNames)]
```

5.3 Running justRMA

Now we use justRMA to summarize expression at the probeset level. We exclude the 4 CEL files not included for analysis per the sdrf file.

```
d1 <- date()
tcgaExpression <- justRMA(filenames = celFilePathsReduced)
tcgaExpression <- exprs(tcgaExpression)
d2 <- date()
c(d1, d2)
```

```
## [1] "Wed Nov 20 10:53:29 2013" "Wed Nov 20 10:57:17 2013"
```

The justRMA computation takes between 5 and 6 minutes on my MacBook Pro.

As an aside, we note that the RMA values computed here match the Level 2 values reported by TCGA quite well (to about 4 decimal places). Given that the group at the Broad is using a distinct implementation of RMA written for GenePattern, the differences are within roundoff error,

and should have no substantive effect on any analyses. This difference in coding may also explain why the row (probeset) ordering produced by justRMA differs from that reported in the Level 2 files.

6 Mapping CEL Names to Sample Barcodes

We now identify the sample barcodes using the sdrf file and parse them for more information.

```
barcodeRows <- match(celFileNames, as.character(sdrf[, "Array.Data.File"]))
sampleBarcodes <- as.character(sdrf[barcodeRows, "Extract.Name"])

sum(duplicated(sampleBarcodes))
```

```
## [1] 1
```

```
sampleBarcodes[sampleBarcodes == sampleBarcodes[duplicated(sampleBarcodes)]]
```

```
## [1] "TCGA-29-1704-02A-01R-0808-01" "TCGA-29-1704-02A-01R-0808-01"
```

```
celFileNames[sampleBarcodes == sampleBarcodes[duplicated(sampleBarcodes)]]
```

```
##                                                        Batch.2720
## "TARRE_p_MultiPlate_TCGA_SS_MA_Ref_HT_HG-U133A_96-HTA_F05_586106.CEL"
##                                                        Batch.2724
##          "URGED_p_TCGA_Pop_May2011_HT_HG-U133A_96-HTA_D05_786990.CEL"
```

```
unusedCELs
```

```
## [1] "URGED_p_TCGA_Pop_May2011_HT_HG-U133A_96-HTA_D05_786990.CEL"
## [2] "FEAST_p_TCGA_B20_21_Expression_HT_HG-U133A_96-HTA_F04_516474.CEL"
## [3] "FEAST_p_TCGA_B20_21_Expression_HT_HG-U133A_96-HTA_G06_516372.CEL"
## [4] "TARRE_p_MultiPlate_TCGA_SS_MA_Ref_HT_HG-U133A_96-HTA_F05_586106.CEL"
```

```
sampleBarcodes[duplicated(sampleBarcodes)] <- paste(sampleBarcodes[duplicated(sampleBarcodes)],
    "Rep", sep = ".")

sourceSite <- substr(sampleBarcodes, 6, 7)
patientID <- substr(sampleBarcodes, 9, 12)
sampleType <- substr(sampleBarcodes, 14, 15)
sampleTypeText <- rep("primaryTumor", nFiles)
sampleTypeText[sampleType == "02"] <- "recurrentTumor"
sampleTypeText[sampleType == "11"] <- "normalTissue"

sampleUsed <- rep("yes", nFiles)
sampleUsed[match(unusedCELs, celFileNames)] <- "no"
```

As noted above, one of the barcodes is used twice. To allow the barcodes to be used as sample IDs (rownames in a data frame), we add a suffix to the latter occurrence. Since the two CEL files for the same sample are among the four CEL files omitted from the analysis, the point is somewhat moot.

We now bundle these bits of information into a data frame.

```
tcgaSampleInfo <- data.frame(sourceSite = sourceSite, patientID = patientID,
    sampleType = sampleType, sampleTypeText = sampleTypeText, sampleBatch = sampleBatch,
    row.names = sampleBarcodes)
tcgaSampleInfo <- tcgaSampleInfo[sampleUsed == "yes", ]

tcgaSampleInfo[1:4, ]
```

```
##                              sourceSite patientID sampleType
## TCGA-13-0758-01A-01R-0362-01         13      0758         01
## TCGA-09-0364-01A-02R-0362-01         09      0364         01
## TCGA-13-0723-01A-02R-0362-01         13      0723         01
## TCGA-13-0757-01A-01R-0362-01         13      0757         01
##                              sampleTypeText sampleBatch
## TCGA-13-0758-01A-01R-0362-01   primaryTumor           9
## TCGA-09-0364-01A-02R-0362-01   primaryTumor           9
## TCGA-13-0723-01A-02R-0362-01   primaryTumor           9
## TCGA-13-0757-01A-01R-0362-01   primaryTumor           9
```

7 Saving RData

Now we save the relevant information to an RData object.

```
colnames(tcgaExpression) <- as.character(sdrf[match(colnames(tcgaExpression),
    as.character(sdrf[, "Array.Data.File"])), "Extract.Name"])

all(colnames(tcgaExpression) == rownames(tcgaSampleInfo))
```

```
## [1] TRUE
```

```
save(tcgaSampleInfo, tcgaDataDirs, tcgaFiles, tcgaExpression, file = file.path("RDataObjects",
    "tcgaExpression.RData"))
```

8 Appendix

8.1 File Location

```
getwd()
```

```
## [1] "/Users/slt/SLT WORKSPACE/EXEMPT/OVARIAN/Ovarian residual disease study 2012/RD
manuscript/Web page for paper/Webpage"
```

8.2 SessionInfo

```
sessionInfo()
```

```
## R version 3.0.2 (2013-09-25)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] hthgu133acdf_2.12.0  AnnotationDbi_1.22.6 affy_1.38.1
## [4] Biobase_2.20.1       BiocGenerics_0.6.0   knitr_1.5
##
## loaded via a namespace (and not attached):
##  [1] affyio_1.28.0         BiocInstaller_1.10.4  DBI_0.2-7
##  [4] evaluate_0.5.1        formatR_0.9           IRanges_1.18.4
##  [7] preprocessCore_1.22.0 RSQLite_0.11.4        stats4_3.0.2
## [10] stringr_0.6.2         tools_3.0.2           zlibbioc_1.6.0
```

Assembling Clinical Information for the TCGA Ovarian Data

by Keith A. Baggerly

1 Executive Summary

1.1 Introduction

We want to produce an RData file with the clinical information for the ovarian cancer samples profiled by TCGA.

1.2 Methods

We acquired the gzipped tarball containing the biotab clinical information from the open access TCGA http page, https://tcga-data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/ov/bcr/biotab/clin/, on Sep 14, 2012.

We load the "clinical_patient_ov" information into a data frame, and construct an R "Surv" object for overall survival.

We also construct a binary indicator vector for the presence or absence of residual disease (RD).

1.3 Results

We save tcgaClinical, tcgaOSYrs, and tcgaRD to the RData file "tcgaClinical.RData".

2 Libraries

We first load the libraries we will use in this report.

```
library(survival)
```

3 Specifying the Raw Data Location

Here, we specify the location of the data we acquired from TCGA on our local system. You will need to acquire these files and adjust this path before running this report yourself.

```
pathToTCGAData <- file.path("RawData", "TCGA", "Clinical")
```

4 Loading the Data

Here we simply load the table of clinical information.

```
tcgaClinical <- read.table(file.path(pathToTCGAData, "clinical_patient_ov.txt"),
    header = TRUE, sep = "\t", row.names = 1)
dim(tcgaClinical)
```

```
## [1] 576  36
```

```
tcgaClinical[1, ]
```

```
##             age_at_initial_pathologic_diagnosis
## TCGA-04-1331                                  78
##             anatomic_organ_subdivision
## TCGA-04-1331           [Not Available]
##                              bcr_patient_uuid date_of_form_completion
```

```
## TCGA-04-1331 6d10d4ee-6331-4bba-93bc-a7b64cc0b22a              2009-03-26
##              date_of_initial_pathologic_diagnosis days_to_birth
## TCGA-04-1331                           2004-00-00        -28848
##              days_to_death days_to_initial_pathologic_diagnosis
## TCGA-04-1331          1336                                   0
##              days_to_last_followup eastern_cancer_oncology_group
## TCGA-04-1331                  1224                [Not Available]
##                        ethnicity gender gynecologic_figo_staging_system
## TCGA-04-1331 NOT HISPANIC OR LATINO FEMALE                  [Not Available]
##                     histological_type          icd_10 icd_o_3_histology
## TCGA-04-1331 Serous Cystadenocarcinoma [Not Available]            8441/3
##              icd_o_3_site informed_consent_verified
## TCGA-04-1331        C56.9                       YES
##              initial_pathologic_diagnosis_method   jewish_origin
## TCGA-04-1331                      [Not Available] [Not Available]
##              karnofsky_performance_score lymphatic_invasion
## TCGA-04-1331             [Not Available]               YES
##              neoplasm_histologic_grade patient_id
## TCGA-04-1331                        G3       1331
##              performance_status_scale_timing person_neoplasm_cancer_status
## TCGA-04-1331                 [Not Available]                    WITH TUMOR
##              pretreatment_history  race  residual_tumor tissue_source_site
## TCGA-04-1331                   NO WHITE [Not Available]                  4
##              tumor_histologic_subtype tumor_residual_disease tumor_stage
## TCGA-04-1331       Cystadenocarcinoma               1-10 mm        IIIC
##              tumor_tissue_site venous_invasion vital_status
## TCGA-04-1331             OVARY             NO     DECEASED
```

5 Defining Overall Survival

Next, we define an R “Surv” object for overall survival (OS). We begin by looking at the recorded values for patient status.

```
table(tcgaClinical[, "vital_status"])
```

```
##
## [Not Available]        DECEASED          LIVING
##               4             297             275
```

```
tcgaClinical[1:15, "days_to_death"]
```

```
##  [1] 1336            1247            55              [Not Applicable]
##  [5] 61              [Not Applicable] [Not Applicable] 563
##  [9] 361             [Not Applicable] [Not Applicable] 1483
## [13] 656             1946            [Not Applicable]
## 275 Levels: [Not Applicable] [Not Available] 1000 1003 1007 1013 ... 976
```

```
tcgaClinical[1:15, "days_to_last_followup"]
```

```
##  [1] 1224            1247            55              1495
##  [5] 61              1418            [Not Available] 563
##  [9] 361             1992            1918            1483
## [13] 656             1946            1991
## 488 Levels: [Not Available] 0 1004 1007 1011 1013 1018 1024 1025 ... 999
```

```
tcgaClinical[1:15, "vital_status"]
```

```
##  [1] DECEASED DECEASED DECEASED LIVING   DECEASED LIVING   LIVING
##  [8] DECEASED DECEASED LIVING   LIVING   DECEASED DECEASED DECEASED
## [15] LIVING
## Levels: [Not Available] DECEASED LIVING
```

Vital status is available for almost all of the 576 patients. Checking the times available shows that days to death can exceed the days to last followup (entry 1), and that days to death is not available for patients still living, so we should use the former for deceased patients and the latter for living ones.

The above conclusions are based on a small sampling of the data. We perform the more extensive sanity checks for verification here.

```
daysToDeath <- as.numeric(as.character(tcgaClinical[, "days_to_death"]))
```

```
## Warning: NAs introduced by coercion
```

```
daysToLastFollowup <- as.numeric(as.character(tcgaClinical[, "days_to_last_followup"]))
```

```
## Warning: NAs introduced by coercion
```

```
vitalStatus <- as.character(tcgaClinical[, "vital_status"])

summary(daysToDeath - daysToLastFollowup)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0     0.0     0.0    24.2     0.0  1200.0     279
```

```
summary(daysToDeath[vitalStatus == "LIVING"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##      NA      NA      NA     NaN      NA      NA     275
```

```
daysToDeath[vitalStatus == "[Not Available]"]
```

```
## [1] NA NA NA NA
```

```
daysToLastFollowup[vitalStatus == "[Not Available]"]
```

```
## [1] NA NA  0 NA
```

The sanity checks are passed. We now assemble the Surv object.

```
daysToEvent <- rep(NA, nrow(tcgaClinical))
daysToEvent[vitalStatus == "LIVING"] <- daysToLastFollowup[vitalStatus == "LIVING"]
daysToEvent[vitalStatus == "DECEASED"] <- daysToDeath[vitalStatus == "DECEASED"]
eventStatus <- rep(NA, nrow(tcgaClinical))
eventStatus[vitalStatus == "LIVING"] <- "Censored"
eventStatus[vitalStatus == "DECEASED"] <- "Uncensored"

tcgaOSYrs <- Surv(daysToEvent/365, eventStatus == "Uncensored")
rownames(tcgaOSYrs) <- rownames(tcgaClinical)
```

6 Defining a Residual Disease Indicator

Now we summarize the Residual Disease (RD) information into a single indicator vector specifying if there is any RD (“RD”) or no RD (“No RD”). We begin by tabulating the information we have.

```
table(tcgaClinical[, "tumor_residual_disease"])
```

```
##
```

```
##        [Not Available]                >20 mm               1-10 mm
##                     62                  104                  254
##               11-20 mm No Macroscopic disease
##                     38                  118
```

We now define the indicator.

```
tcgaRD <- rep(NA, nrow(tcgaClinical))
tcgaRD[tcgaClinical[, "tumor_residual_disease"] == ">20 mm"] <- "RD"
tcgaRD[tcgaClinical[, "tumor_residual_disease"] == "1-10 mm"] <- "RD"
tcgaRD[tcgaClinical[, "tumor_residual_disease"] == "11-20 mm"] <- "RD"
tcgaRD[tcgaClinical[, "tumor_residual_disease"] == "No Macroscopic disease"] <- "No RD"
table(tcgaRD)
```

```
## tcgaRD
## No RD    RD
##   118   396
```

```
names(tcgaRD) <- rownames(tcgaClinical)
```

7 Saving RData

Now we save the relevant information to an RData object.

```
save(tcgaClinical, tcgaOSYrs, tcgaRD, file = file.path("RDataObjects", "tcgaClinical.RData"))
```

8 Appendix

8.1 File Location

```
getwd()
```

```
## [1] "\\\\mdadqsfs02/workspace/kabagg/RDPaper/Webpage/ResidualDisease"
```

8.2 Session Info

```
sessionInfo()
```

```
## R version 2.15.3 (2013-03-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] splines   stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] survival_2.37-4 knitr_1.2
##
## loaded via a namespace (and not attached):
## [1] digest_0.6.3   evaluate_0.4.3 formatR_0.7    stringr_0.6.2
## [5] tools_2.15.3
```

Assembling an RMA Quantification Matrix for the Tothill Ovarian Data

by Keith A. Baggerly

1 Executive Summary

1.1 Introduction

We want to produce an RData file with a matrix of RMA expression values for the ovarian cancer samples profiled by Tothill et al with Affymetrix U133+2 arrays.

1.2 Methods

We acquired a tarball of the 285 gzipped CEL files used from the Gene Expression Omnibus (GEO) page for GSE9891, http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE9891, on September 13, 2012.

We used justRMA to compute RMA fits, and used our previously assembled clinical information to map the GEO GSM ids to remap the column (sample) names.

1.3 Results

We save tothillExpression to the RData file "tothillExpression.RData".

2 Libraries

We first load the libraries we will use in this report.

```
library(affy)
library(hgu133plus2cdf)
```

3 Loading Clinical Information

Next, we load our previously assembled clinical information.

```
load(file.path("RDataObjects", "tothillClinical.RData"))
```

4 Specifying the Raw Data Location

Here, we specify the location of the data we acquired from GEO on our local system. You will need to acquire these files and adjust this path before running this report yourself.

```
pathToTothillData <- file.path("RawData", "Tothill", "CEL_Files")
```

5 Quantifying The CEL Files

First, we specify the CEL file paths in a character vector for passing to justRMA.

```
celFileNames <- dir(pathToTothillData, pattern = "^GSM")
celFilePaths <- file.path(pathToTothillData, celFileNames)
```

Now we use justRMA to summarize expression at the probeset level.

```
d1 <- date()
tothillExpression <- justRMA(filenames = celFilePaths, compress = TRUE)
tothillExpression <- exprs(tothillExpression)
d2 <- date()
c(d1, d2)
```

```
## [1] "Wed Nov 20 11:18:36 2013" "Wed Nov 20 11:22:03 2013"
```

```
dim(tothillExpression)
```

```
## [1] 54675   285
```

```
tothillExpression[1:3, 1:3]
```

```
##           GSM249714.CEL.gz GSM249715.CEL.gz GSM249716.CEL.gz
## 1007_s_at           10.037           10.591           10.291
## 1053_at              6.808            7.710            6.657
## 117_at               5.804            5.791            5.905
```

The justRMA computation takes about 4 minutes on my MacBook Pro.

6 Mapping CEL Names to Sample IDs

We now use the clinical information to replace the GEO GSM ids with the sample ids in the column names.

```
tempClinRows <- match(substr(colnames(tothillExpression), 1, 9), as.character(tothillClinical[,
    "GEO.ID"]))
tempNames <- rownames(tothillClinical)[tempClinRows]
tothillClinical[tempNames[1:3], ]
```

```
##          GEO.ID SampleID KMeansGroup ClinicalType HistologicSubtype
## X60120 GSM249714    60120           3          LMP               Ser
## X32117 GSM249715    32117           3          LMP               Ser
## X23066 GSM249716    23066           3          LMP               Ser
##        PrimarySite Stage Grade Age Status Pltx Tax Neo MosToRelapse
## X60120          OV    II     1  59     PF    N   N   N           37
## X32117          OV    II     1  26     PF    N   N   N            8
## X23066          OV   III     1  64     PF    N   N   N           18
##        MosToDeath ResidDisease ArraySite
## X60120         37          nil        OV
## X32117          8          nil        OV
## X23066         18          nil        OV
```

```
colnames(tothillExpression)[1:3]
```

```
## [1] "GSM249714.CEL.gz" "GSM249715.CEL.gz" "GSM249716.CEL.gz"
```

```
colnames(tothillExpression) <- tempNames
tothillExpression[1:3, 1:3]
```

```
##          X60120 X32117 X23066
## 1007_s_at 10.037 10.591 10.291
## 1053_at    6.808  7.710  6.657
## 117_at     5.804  5.791  5.905
```

7 Saving RData

Now we save the relevant information to an RData object.

```
save(tothillExpression, file = file.path("RDataObjects", "tothillExpression.RData"))
```

8 Appendix

8.1 File Location

```
getwd()
```

```
## [1] "/Users/slt/SLT WORKSPACE/EXEMPT/OVARIAN/Ovarian residual disease study 2012/RD
manuscript/Web page for paper/Webpage"
```

8.2 SessionInfo

```
sessionInfo()
```

```
## R version 3.0.2 (2013-09-25)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] hgu133plus2cdf_2.12.0 AnnotationDbi_1.22.6  affy_1.38.1
## [4] Biobase_2.20.1        BiocGenerics_0.6.0    knitr_1.5
##
## loaded via a namespace (and not attached):
##  [1] affyio_1.28.0         BiocInstaller_1.10.4  DBI_0.2-7
##  [4] evaluate_0.5.1        formatR_0.9           IRanges_1.18.4
##  [7] preprocessCore_1.22.0 RSQLite_0.11.4        stats4_3.0.2
## [10] stringr_0.6.2         tools_3.0.2           zlibbioc_1.6.0
```

9 References

> [1] Tothill RW, Tinker AV, George J, Brown R, Fox SB, Lade S, Johnson DS, Trivett MK, Etemadmoghadam D, Locandro B, Traficante N, Fereday S, Hung JA, Chiew YE, Haviv I; Australian Ovarian Cancer Study Group, Gertig D, DeFazio A, Bowtell DD. Novel molecular subtypes of serous and endometrioid ovarian cancer linked to clinical outcome. *Clin Cancer Res*, **14(16):5198-208, 2008.**

Assembling Clinical Information for the Tothill Ovarian Data

by Keith A. Baggerly

1 Executive Summary

1.1 Introduction

We want to produce an RData file with the clinical information for the ovarian cancer samples profiled by Tothill et al.

1.2 Methods

We acquired clinical annotation from two sources on Sep 13, 2012: "clinical_anns.csv" from the Gene Expression Omnibus (GEO) page for GSE9891, http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE9891, and p.27-30 of the supplementary data pdf for Tothill et al., http://clincancerres.aacrjournals.org/content/14/16/5198/suppl/DC1.

A csv file of this annotation, together with an extra column specifying the GEO GSM ID for each sample, is stored in RawData as tothillClinical.csv.

We load the clinical information into a data frame, and construct R "Surv" objects for overall and progression (relapse) free survival.

We also construct a binary indicator vector for the presence or absence of residual disease (RD).

1.3 Results

We save tothillClinical, tothillOSMos, tothillPFSMos, and tothillRD to the RData file "tothillClinical.RData".

2 Options and Libraries

We first load the options and libraries we will use in this report.

```
library(survival)
```

3 Loading the Data

Here we simply load the table of clinical information.

```
tothillClinical <- read.table(file.path("RawData", "Tothill", "Clinical", "tothillClinical.csv"),
    header = TRUE, sep = ",")
dim(tothillClinical)
```

```
## [1] 285  17
```

```
tothillClinical[1:3, ]
```

```
##      GEO.ID SampleID KMeansGroup ClinicalType HistologicSubtype
## 1 GSM249839       49           5          MAL              Ser
## 2 GSM250001      129           1          MAL              Ser
## 3 GSM250000      146          NC          MAL              Ser
##   PrimarySite Stage Grade Age Status Pltx Tax Neo MosToRelapse MosToDeath
## 1          OV   III     3  56      D    Y   N   N            7          8
## 2          OV   III     3  65      D    Y   N   N            7         15
## 3          OV   III     3  56     PF    Y   N   N          166        166
##   ResidDisease ArraySite
## 1           <1        OV
```

```
## 2           >1        PE
## 3           >1        OV
```

```
rownames(tothillClinical) <- paste("X", tothillClinical[, "SampleID"], sep = "")
```

4 Defining Overall and Progression-Free Survival

Next, we define R “Surv” objects for overall survival (OS) and progression-free survival (PFS). We begin by looking at the recorded values for patient status.

```
table(tothillClinical[, "Status"])
```

```
##
##       D  D*  PF   R
##   3 111   2  92  77
```

According to the supplementary information table, D = Dead, D* = Dead of Other Causes, PF = Alive Progression-Free, and R = Alive and Relapsed.

Next, we define indicator vectors for OS and PFS. We begin with OS.

```
tothillOSStatus <- rep(NA, nrow(tothillClinical))
tothillOSStatus[tothillClinical$Status == "D"] <- "Uncensored"
tothillOSStatus[tothillClinical$Status == "D*"] <- "Uncensored"
tothillOSStatus[tothillClinical$Status == "PF"] <- "Censored"
tothillOSStatus[tothillClinical$Status == "R"] <- "Censored"
table(tothillOSStatus)
```

```
## tothillOSStatus
##   Censored Uncensored
##        169        113
```

Next we deal with PFS.

```
tothillPFStatus <- rep(NA, nrow(tothillClinical))
tothillPFStatus[tothillClinical$Status == "D"] <- "Uncensored"
tothillPFStatus[tothillClinical$Status == "D*"] <- "Uncensored"
tothillPFStatus[tothillClinical$Status == "PF"] <- "Censored"
tothillPFStatus[tothillClinical$Status == "R"] <- "Uncensored"
table(tothillPFStatus)
```

```
## tothillPFStatus
##   Censored Uncensored
##         92        190
```

Now we create the Surv objects.

```
tothillOSMos <- Surv(tothillClinical[, "MosToDeath"], tothillOSStatus == "Uncensored")
rownames(tothillOSMos) <- rownames(tothillClinical)

tothillPFSMos <- Surv(tothillClinical[, "MosToRelapse"], tothillPFStatus ==
    "Uncensored")
rownames(tothillPFSMos) <- rownames(tothillClinical)
```

5 Defining a Residual Disease Indicator

Now we summarize the Residual Disease (RD) information into a single indicator vector specifying if there is any RD (“RD”) or no RD (“No RD”). We begin by tabulating the information we have.

```
table(tothillClinical[, "ResidDisease"])
```

```
##
##            <1            >1 macro size NK           nil            NK
##            76            70            18            84            37
```

According to the supplementary information from Tothill et al., “macro size NK” = macroscopic disease size unknown (but there is some), and “NK” = residual disease unknown.

We now define the indicator.

```
tothillRD <- rep(NA, nrow(tothillClinical))
tothillRD[tothillClinical[, "ResidDisease"] == "<1"] <- "RD"
tothillRD[tothillClinical[, "ResidDisease"] == ">1"] <- "RD"
tothillRD[tothillClinical[, "ResidDisease"] == "macro size NK"] <- "RD"
tothillRD[tothillClinical[, "ResidDisease"] == "nil"] <- "No RD"
table(tothillRD)
```

```
## tothillRD
## No RD    RD
##    84   164
```

```
names(tothillRD) <- rownames(tothillClinical)
```

6 Saving RData

Now we save the relevant information to an RData object.

```
save(tothillClinical, tothillOSMos, tothillPFSMos, tothillRD, file = file.path("RDataObjects",
    "tothillClinical.RData"))
```

7 Appendix

7.1 File Location

```
getwd()
```

```
## [1] "\\\\mdadqsfs02/workspace/kabagg/RDPaper/Webpage/ResidualDisease"
```

7.2 SessionInfo

```
sessionInfo()
```

```
## R version 2.15.3 (2013-03-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] splines   stats     graphics  grDevices utils     datasets  methods
## [8] base
```

```
##
## other attached packages:
## [1] survival_2.37-4 knitr_1.2
##
## loaded via a namespace (and not attached):
## [1] digest_0.6.3   evaluate_0.4.3 formatR_0.7    stringr_0.6.2
## [5] tools_2.15.3
```

8 References

[1] Tothill RW, Tinker AV, George J, Brown R, Fox SB, Lade S, Johnson DS, Trivett MK, Etemadmoghadam D, Locandro B, Traficante N, Fereday S, Hung JA, Chiew YE, Haviv I; Australian Ovarian Cancer Study Group, Gertig D, DeFazio A, Bowtell DD. Novel molecular subtypes of serous and endometrioid ovarian cancer linked to clinical outcome. *Clin Cancer Res*, **14(16):5198-208, 2008.**

Assembling an RMA Quantification Matrix for the Bonome Ovarian Data

by Keith A. Baggerly

1 Executive Summary

1.1 Introduction

We want to produce an RData file with a matrix of RMA expression values for the ovarian cancer samples profiled by Bonome et al. with Affymetrix U133A arrays.

1.2 Methods

We acquired a tarball of the 195 gzipped CEL files (185 tumor samples and 10 normal ovarian samples) used from the Gene Expression Omnibus (GEO) page for GSE26712, http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE26712, on Sep 10, 2012.

We used justRMA to compute RMA fits, and used our previously assembled clinical information to map the GEO GSM ids to remap the column (sample) names.

1.3 Results

We save bonomeExpression to the RData file “bonomeExpression.RData”.

2 Libraries

We first load the libraries we will use in this report.

```
library(affy)
library(hgu133acdf)
```

3 Loading Clinical Information

Next, we load our previously assembled clinical information.

```
load(file.path("RDataObjects", "bonomeClinical.RData"))
```

4 Specifying the Raw Data Location

Here, we specify the location of the data we acquired from GEO on our local system. You will need to acquire these files and adjust this path before running this report yourself.

```
pathToBonomeData <- file.path("RawData", "Bonome", "CEL_Files")
```

5 Quantifying The CEL Files

First, we specify the CEL file paths in a character vector for passing to justRMA.

```
celFileNames <- dir(pathToBonomeData, pattern = "^GSM")
celFilePaths <- file.path(pathToBonomeData, celFileNames)
```

Now we use justRMA to summarize expression at the probeset level.

```
d1 <- date()
bonomeExpression <- justRMA(filenames = celFilePaths, compress = TRUE)
bonomeExpression <- exprs(bonomeExpression)
d2 <- date()
c(d1, d2)
```

```
## [1] "Wed Jun 12 14:35:55 2013" "Wed Jun 12 14:38:55 2013"
```

```
dim(bonomeExpression)
```

```
## [1] 22283   195
```

```
bonomeExpression[1:3, 1:3]
```

```
##           GSM657519_HOSE2237.CEL.gz GSM657520_HOSE2008.CEL.gz
## 1007_s_at                     8.693                     8.425
## 1053_at                       5.100                     5.071
## 117_at                        5.084                     5.868
##           GSM657521_HOSE2061.CEL.gz
## 1007_s_at                     8.570
## 1053_at                       5.099
## 117_at                        5.238
```

The justRMA computation takes about 2 minutes on my MacBook Pro.

6 Mapping CEL Names to Sample IDs

We now use the clinical information to replace the GEO GSM ids with the sample ids in the column names.

```
tempClinRows <- match(substr(colnames(bonomeExpression), 1, 9), as.character(bonomeClinical[,
    "GEO.ID"]))
tempNames <- rownames(bonomeClinical)[tempClinRows]
bonomeClinical[tempNames[1:3], ]
```

```
##             GEO.ID SampleID SurgeryOutcome Status SurvivalYears
## HOSE2237 GSM657519 HOSE2237                                  NA
## HOSE2008 GSM657520 HOSE2008                                  NA
## HOSE2061 GSM657521 HOSE2061                                  NA
```

```
colnames(bonomeExpression)[1:3]
```

```
## [1] "GSM657519_HOSE2237.CEL.gz" "GSM657520_HOSE2008.CEL.gz"
## [3] "GSM657521_HOSE2061.CEL.gz"
```

```
colnames(bonomeExpression) <- tempNames
bonomeExpression[1:3, 1:3]
```

```
##           HOSE2237 HOSE2008 HOSE2061
## 1007_s_at    8.693    8.425    8.570
## 1053_at      5.100    5.071    5.099
## 117_at       5.084    5.868    5.238
```

7 Saving RData

Now we save the relevant information to an RData object.

```
save(bonomeExpression, file = file.path("RDataObjects", "bonomeExpression.RData"))
```

8 Appendix

8.1 File Location

```
getwd()
```

```
## [1] "\\\\mdadqsfs02/workspace/kabagg/RDPaper/Webpage/ResidualDisease"
```

8.2 SessionInfo

```
sessionInfo()
```

```
## R version 2.15.3 (2013-03-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] hgu133acdf_2.11.0    AnnotationDbi_1.20.7 affy_1.36.1
## [4] Biobase_2.18.0       BiocGenerics_0.4.0   knitr_1.2
##
## loaded via a namespace (and not attached):
##  [1] affyio_1.26.0         BiocInstaller_1.8.3   DBI_0.2-7
##  [4] digest_0.6.3          evaluate_0.4.3        formatR_0.7
##  [7] IRanges_1.16.6        parallel_2.15.3       preprocessCore_1.20.0
## [10] RSQLite_0.11.4        stats4_2.15.3         stringr_0.6.2
## [13] tools_2.15.3          zlibbioc_1.4.0
```

9 References

> [1] Bonome T, Levine DA, Shih J, Randonovich M, Pise-Masison CA, Bogomolniy F, Ozbun L, Brady J, Barrett JC, Boyd J, Birrer MJ. A gene signature predicting for survival in suboptimally debulked patients with ovarian cancer. *Cancer Res*, **68(13)**:5478-86, 2008.

Assembling Clinical Information for the Bonome Ovarian Data

by Keith A. Baggerly

1 Executive Summary

1.1 Introduction

We want to produce an RData file with the clinical information for the ovarian cancer samples profiled by Bonome et al. on U133A arrays.

1.2 Methods

We acquired clinical annotation from the Gene Expression Omnibus (GEO) pages descending from the main page GSE26712, http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE26712, on Sep 12, 2012. This includes the GEO GSM id for each sample, the sample ID, Surgery Outcome (DOD, AWD, NED), and Survival in Years.

A csv file of this annotation is stored in RawData as bonomeClinical.csv.

We load the clinical information into a data frame, and construct an R “Surv”“ object for overall survival.

1.3 Results

We save bonomeClinical and bonomeOSYrs to the RData file "bonomeClinical.RData”.

2 Libraries

We first load the libraries we will use in this report.

```
library(survival)
```

```
## Loading required package: splines
```

3 Loading the Data

Here we simply load the table of clinical information.

```
bonomeClinical <- read.table(file.path("RawData", "Bonome", "Clinical", "bonomeClinical.csv"),
    header = TRUE, sep = ",")
dim(bonomeClinical)
```

```
## [1] 195   5
```

```
bonomeClinical[1:3, ]
```

```
##      GEO.ID SampleID SurgeryOutcome Status SurvivalYears
## 1 GSM657519 HOSE2237                                  NA
## 2 GSM657520 HOSE2008                                  NA
## 3 GSM657521 HOSE2061                                  NA
```

```
rownames(bonomeClinical) <- as.character(bonomeClinical[, "SampleID"])
```

4 Defining Overall Survival

Next, we define an R “Surv”“ object for overall survival (OS) We begin by looking at the recorded values for patient status.

```
table(bonomeClinical[, "Status"])
```

```
##
##     AWD DOD NED
##  10  24 129  32
```

Here, AWD = Alive with Disease, DOD = Dead of Disease, and NED = Alive with no Evidence of Disease.

Next, we define an indicator vectors for OS.

```
bonomeOSStatus <- rep(NA, nrow(bonomeClinical))
bonomeOSStatus[bonomeClinical[, "Status"] == "AWD"] <- "Censored"
bonomeOSStatus[bonomeClinical[, "Status"] == "DOD"] <- "Uncensored"
bonomeOSStatus[bonomeClinical[, "Status"] == "NED"] <- "Censored"
table(bonomeOSStatus)
```

```
## bonomeOSStatus
##   Censored Uncensored
##         56        129
```

Now we create the Surv object.

```
bonomeOSYrs <- Surv(bonomeClinical[, "SurvivalYears"], bonomeOSStatus == "Uncensored")
rownames(bonomeOSYrs) <- rownames(bonomeClinical)
```

5 Saving RData

Now we save the relevant information to an RData object.

```
save(bonomeClinical, bonomeOSYrs, file = file.path("RDataObjects", "bonomeClinical.RData"))
```

6 Appendix

6.1 File Location

```
getwd()
```

```
## [1] "\\\\mdadqsfs02/workspace/kabagg/RDPaper/Webpage/ResidualDisease"
```

6.2 SessionInfo

```
sessionInfo()
```

```
## R version 2.15.3 (2013-03-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
```

```
## attached base packages:
## [1] splines   stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] survival_2.37-4 knitr_1.2
##
## loaded via a namespace (and not attached):
## [1] digest_0.6.3   evaluate_0.4.3 formatR_0.7    stringr_0.6.2
## [5] tools_2.15.3
```

7 References

[1] Bonome T, Levine DA, Shih J, Randonovich M, Pise-Masison CA, Bogomolniy F, Ozbun L, Brady J, Barrett JC, Boyd J, Birrer MJ. A gene signature predicting for survival in suboptimally debulked patients with ovarian cancer. *Cancer Res*, **68(13)**:5478-86, 2008.

Assembling an RMA Quantification Matrix for the CCLE Data

Keith A. Baggerly

1 Executive Summary

1.1 Introduction

We want to produce an RData file with a matrix of RMA expression values for the cancer cell lines profiled as part of the Cancer Cell Line Encylcopedia (CCLE) on Affymetrix U133+2 arrays.

1.2 Methods

We acquired a tarball of the 917 gzipped CEL files used from the Gene Expression Omnibus (GEO) page for GSE36133, http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE36133, on Sep 14, 2012. (Warning - this file is over 4G, so it may not download properly to a 32-bit machine.)

We used justRMA to compute RMA fits, and used our previously assembled clinical information to map the GEO GSM ids to remap the column (sample) names.

1.3 Results

We save ccleExpression to the RData file “ccleExpression.RData”.

2 Libraries

We first load the libraries we will use in this report.

```
library(affy)
library(hgu133plus2cdf)
```

3 Loading Clinical Information

Next, we load our previously assembled clinical information.

```
load(file.path("RDataObjects", "ccleClinical.RData"))
```

4 Specifying the Raw Data Location

Here, we specify the location of the data we acquired from GEO on our local system. You will need to acquire these files and adjust this path before running this report yourself.

```
pathToCCLEData <- file.path("RawData", "CCLE", "CEL_Files")
```

5 Quantifying The CEL Files

First, we specify the CEL file paths in a character vector for passing to justRMA.

```
celFileNames <- dir(pathToCCLEData, pattern = "^GSM")
celFilePaths <- file.path(pathToCCLEData, celFileNames)
```

Now we use justRMA to summarize expression at the probeset level.

```
d1 <- date()
ccleExpression <- justRMA(filenames = celFilePaths, compress = TRUE)
ccleExpression <- exprs(ccleExpression)
d2 <- date()
```

```
c(d1, d2)
```

```
## [1] "Thu Jun 13 07:53:42 2013" "Thu Jun 13 08:08:35 2013"
```

```
dim(ccleExpression)
```

```
## [1] 54675   917
```

```
ccleExpression[1:3, 1:3]
```

```
##           GSM886835.CEL.gz GSM886836.CEL.gz GSM886837.CEL.gz
## 1007_s_at            8.400            7.699           10.638
## 1053_at             10.062            9.331           10.577
## 117_at               4.257            3.966            3.905
```

The justRMA computation takes about 40 minutes on my MacBook Pro; the sheer volume of the data makes this challenging.

6 Mapping CEL Names to Sample IDs

We now use the clinical information to replace the GEO GSM ids with the sample ids in the column names.

```
tempClinRows <- match(substr(colnames(ccleExpression), 1, 9), as.character(ccleClinical[,
    "GEO.ID"]))
tempNames <- rownames(ccleClinical)[tempClinRows]
ccleClinical[tempNames[1:3], ]
```

```
##           GEO.ID sourceName            primarySite    histology
## 1321N1 GSM886835      ECACC central_nervous_system       glioma
## 143B   GSM886836       ATCC                   bone osteosarcoma
## 22Rv1  GSM886837       ATCC              prostate    carcinoma
##            subtype
## 1321N1 astrocytoma
## 143B
## 22Rv1
```

```
colnames(ccleExpression)[1:3]
```

```
## [1] "GSM886835.CEL.gz" "GSM886836.CEL.gz" "GSM886837.CEL.gz"
```

```
colnames(ccleExpression) <- tempNames
ccleExpression[1:3, 1:3]
```

```
##           1321N1  143B  22Rv1
## 1007_s_at  8.400 7.699 10.638
## 1053_at   10.062 9.331 10.577
## 117_at     4.257 3.966  3.905
```

7 Saving RData

Now we save the relevant information to an RData object.

```
save(ccleExpression, file = file.path("RDataObjects", "ccleExpression.RData"))
```

8 Appendix

8.1 File Location

```
getwd()
```

```
## [1] "/workspace/kabagg/RDPaper/Webpage/ResidualDisease"
```

8.2 SessionInfo

```
sessionInfo()
```

```
## R version 2.15.1 (2012-06-22)
## Platform: x86_64-unknown-linux-gnu (64-bit)
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=C                 LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] hgu133plus2cdf_2.10.0 AnnotationDbi_1.22.5  affy_1.34.0
## [4] Biobase_2.16.0        BiocGenerics_0.6.0    markdown_0.5.3
## [7] knitr_0.9
##
## loaded via a namespace (and not attached):
##  [1] affyio_1.24.0         BiocInstaller_1.4.9   DBI_0.2-6
##  [4] digest_0.6.3          evaluate_0.4.3        formatR_0.7
##  [7] IRanges_1.18.0        preprocessCore_1.18.0 RSQLite_0.11.3
## [10] stats4_2.15.1         stringr_0.6.2         tools_2.15.1
## [13] zlibbioc_1.2.0
```

9 References

> [1] Barretina J, Caponigro G, Stransky N, Venkatesan K et al. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, **483(7391)**:603-7, 2012. PMID: 22460905.

Assembling Clinical Information for the CCLE Data

by Keith A. Baggerly

1 Executive Summary

1.1 Introduction

We want to produce an RData file with the clinical (annotation) information for the cancer cell lines profiled as part of the Cancer Cell Line Encylcopedia (CCLE).

1.2 Methods

We use GEOquery to parse the annotation information for the 917 cell lines posted at the Gene Expression Omnibus (GEO) as part of GSE36133: http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE36133. We use GEOquery to extract the annotation information contained in the individual GSM files, including cell line name, GSM sample id, site of primary tumor, histology, and histological subtype (when applicable).

We save these results both as a data frame and a csv file.

1.3 Results

We save ccleClinical to the RData file "ccleClinical.RData", and also export the table to ccleClinical.csv in RawData.

2 Libraries

We first load the options and libraries we will use in this report.

```
library(GEOquery)
```

3 Loading the Data

Here we simply use the GEOquery package to download the annotation information (and posted quantifications) directly from GEO. Since the quantifications are based on a nonstandard CDF file, we prefer to build our own from the CEL files. Since the number of CEL files is large, GEO partitions the results into component series files – each contains info on at most 255 entries, so there are 4 files for the CCLE data.

```
d1 <- date()
ccleFromGEO <- getGEO("GSE36133")
d2 <- date()
c(d1, d2)
```

```
## [1] "Wed Jun 12 14:54:36 2013" "Wed Jun 12 14:55:22 2013"
```

```
length(ccleFromGEO)
```

```
## [1] 4
```

```
names(ccleFromGEO)
```

```
## [1] "GSE36133_series_matrix-1.txt.gz" "GSE36133_series_matrix-2.txt.gz"
## [3] "GSE36133_series_matrix-3.txt.gz" "GSE36133_series_matrix-4.txt.gz"
```

```
class(ccleFromGEO)
```

```
## [1] "list"
```

```
class(ccleFromGEO[[1]])
```

```
## [1] "ExpressionSet"
## attr(,"package")
## [1] "Biobase"
```

Obtaining the data takes about 30 seconds on my MacBook Pro using a high-speed home DSL connection. Judging timing here is a bit tricky, in that it relies on the speed of your internet connection as well as your computer's processing power. We now have a list of ExpressionSet objects to work with.

4 Extracting the Annotation

Since what we really want is the annotation, we need to extract the phenoData from each ExpressionSet and look at the pData from each phenoData object.

4.1 Identifying Annotation Fields of Interest

Before simply bundling the annotation across all files, we examine the results for a few files to see which fields are actually informative.

We first look at the information supplied for a single file.

```
annotBlock1 <- pData(phenoData(ccleFromGEO[[1]]))
dim(annotBlock1)
```

```
## [1] 255  37
```

```
colnames(annotBlock1)
```

```
##  [1] "title"                  "geo_accession"
##  [3] "status"                 "submission_date"
##  [5] "last_update_date"       "type"
##  [7] "channel_count"          "source_name_ch1"
##  [9] "organism_ch1"           "characteristics_ch1"
## [11] "characteristics_ch1.1"  "characteristics_ch1.2"
## [13] "treatment_protocol_ch1" "growth_protocol_ch1"
## [15] "molecule_ch1"           "extract_protocol_ch1"
## [17] "label_ch1"              "label_protocol_ch1"
## [19] "taxid_ch1"              "hyb_protocol"
## [21] "scan_protocol"          "description"
## [23] "data_processing"        "platform_id"
## [25] "contact_name"           "contact_email"
## [27] "contact_laboratory"     "contact_department"
## [29] "contact_institute"      "contact_address"
## [31] "contact_city"           "contact_state"
## [33] "contact_zip/postal_code" "contact_country"
## [35] "contact_web_link"       "supplementary_file"
## [37] "data_row_count"
```

```
annotBlock1[1, ]
```

```
##           title geo_accession               status submission_date
## GSM886835 1321N1     GSM886835 Public on Mar 20 2012     Mar 06 2012
##           last_update_date type channel_count source_name_ch1 organism_ch1
## GSM886835      Mar 20 2012  RNA             1          ECACC Homo sapiens
##                            characteristics_ch1 characteristics_ch1.1
## GSM886835 primary site: central_nervous_system     histology: glioma
```

```
##                     characteristics_ch1.2 treatment_protocol_ch1
## GSM886835 histology subtype1: astrocytoma                   None
##
growth_protocol_ch1
## GSM886835 Cells lines were cultured following growth recommendations from the cell line source
vendors. Cells were incubated at 37 ÂºC at 5% CO2 until 70% confluency was reached.  Pellets were
harvested 48 hours post media change, flash frozen, and stored at -80 ÂºC until nucleic acid
extraction.
##           molecule_ch1
## GSM886835    total RNA
##
extract_protocol_ch1
## GSM886835 RNA was isolated from frozen cell pellets, containing an average cell count of 4.5
million cells, via Trizol (Invitrogen) extraction following the manufacturers instructions. Samples
were quantified using both the Nanodrop 8000 spectrophotometer (ThermoScientific) and via Agilent
2100 Bioanalyzer.
##           label_ch1
## GSM886835    Biotin
##
label_protocol_ch1
## GSM886835 The Affymetrix One-Cycle Target labeling assay was used to synthesize hybridization
target using 1 mg of total RNA. Labeled cRNA hybridization target was produced using a 3-day, semi-
automated workflow
##           taxid_ch1
## GSM886835      9606
##
hyb_protocol
## GSM886835 Fragmentation of the biotin-labeled cRNA prepared the sample for hybridization on the
GeneChip Human U133 2.0 (PN 900467). The washing and staining was performed on the GeneChip
Fluidics station using standard Affymetrix protocols outlined in the Expression Analysis Technical
Manual, 2001, Affymetrix.
##                                                        scan_protocol
## GSM886835 The GeneChipÂ® Scanner 3000 was used for scanning the arrays.
##                               description
## GSM886835 Gene expression data from the CCLE
##
data_processing
## GSM886835 The data were analyzed with 2.14.0, using the packages affyio_1.22.0 and affy_1.32.0,
and RMA as normalization method. rma() was used with default options (normalize=TRUE,
background=TRUE) and with the package hgu133plus2hsentrezgcdf_15.0.0 from Brainarray for the CDF
file.
##           platform_id      contact_name              contact_email
## GSM886835    GPL15308 Nicolas,,Stransky stransky@broadinstitute.org
##           contact_laboratory contact_department contact_institute
## GSM886835      Levi Garraway     Cancer Program   Broad Institute
##              contact_address contact_city contact_state
## GSM886835 7 Cambridge center    Cambridge            MA
##           contact_zip/postal_code contact_country
## GSM886835                   02118            USA
##                     contact_web_link
## GSM886835 www.broadinstitute.org/ccle
##
supplementary_file
## GSM886835
ftp://ftp.ncbi.nlm.nih.gov/pub/geo/DATA/supplementary/samples/GSM886nnn/GSM886835/GSM886835.CEL.gz
##           data_row_count
## GSM886835          18926
```

There's quite a bit of annotation here, but most of it isn't unique to the given cell line, and is thus of less interest to us. We compare annotations for the first two files to see which bits change.

```
annotBlock1[1, ] == annotBlock1[2, ]
```

```
##           title geo_accession status submission_date last_update_date type
## GSM886835 FALSE         FALSE   TRUE            TRUE             TRUE TRUE
##           channel_count source_name_ch1 organism_ch1 characteristics_ch1
## GSM886835          TRUE           FALSE         TRUE                FALSE
```

```
##           characteristics_ch1.1 characteristics_ch1.2
## GSM886835                 FALSE                 FALSE
##           treatment_protocol_ch1 growth_protocol_ch1 molecule_ch1
## GSM886835                   TRUE                TRUE         TRUE
##           extract_protocol_ch1 label_ch1 label_protocol_ch1 taxid_ch1
## GSM886835                 TRUE      TRUE               TRUE      TRUE
##           hyb_protocol scan_protocol description data_processing
## GSM886835         TRUE          TRUE        TRUE            TRUE
##           platform_id contact_name contact_email contact_laboratory
## GSM886835        TRUE         TRUE          TRUE               TRUE
##           contact_department contact_institute contact_address
## GSM886835               TRUE              TRUE            TRUE
##           contact_city contact_state contact_zip/postal_code
## GSM886835         TRUE          TRUE                   TRUE
##           contact_country contact_web_link supplementary_file
## GSM886835            TRUE             TRUE              FALSE
##           data_row_count
## GSM886835           TRUE
```

```
sum(annotBlock1[1, ] != annotBlock1[2, ])
```

```
## [1] 7
```

There are 7 fields whose values change, but two of these (geo_accession and supplementary_file) reflect the fact that the GSM number is different, and this information is already in the row names. This leaves title (the cell line name), source_name_ch1 (where the cell line came from), characteristics_ch1 (the organ location of the primary tumor), characteristics_ch1.1 (the tumor histology), and characteristics_ch1.2 (the histologic subtype, if applicable). We extract these fields for our annotation table.

4.2 Grabbing Interesting Columns

Now we grab the columns of interest from each ExpressionSet, convert them to character matrices, and bind them together into a single object.

```
annotBlock2 <- pData(phenoData(ccleFromGEO[[2]]))
annotBlock3 <- pData(phenoData(ccleFromGEO[[3]]))
annotBlock4 <- pData(phenoData(ccleFromGEO[[4]]))
keyColumns <- c("title", "source_name_ch1", "characteristics_ch1", "characteristics_ch1.1",
    "characteristics_ch1.2")
allAnnot <- rbind(as.matrix(annotBlock1[, keyColumns]), as.matrix(annotBlock2[,
    keyColumns]), as.matrix(annotBlock3[, keyColumns]), as.matrix(annotBlock4[,
    keyColumns]))
dim(allAnnot)
```

```
## [1] 917   5
```

```
allAnnot[1:3, ]
```

```
##           title    source_name_ch1 characteristics_ch1
## GSM886835 "1321N1" "ECACC"         "primary site: central_nervous_system"
## GSM886836 "143B"   "ATCC"          "primary site: bone"
## GSM886837 "22Rv1"  "ATCC"          "primary site: prostate"
##           characteristics_ch1.1     characteristics_ch1.2
## GSM886835 "histology: glioma"       "histology subtype1: astrocytoma"
## GSM886836 "histology: osteosarcoma" ""
## GSM886837 "histology: carcinoma"    ""
```

We have extracted the information desired.

5 Rearranging the Annotation in a Data Frame

While we have all of the information we want, it's not yet arranged the way we want it. We'd prefer to use the cell line names as row names, as

opposed to the GEO ids, and several parts of the text strings (e.g., “primary site:”) appear redundant.

Here we clean up the data and reorder the columns.

```
GEO.ID <- rownames(allAnnot)
cellLineNames <- allAnnot[, "title"]
sourceName <- allAnnot[, "source_name_ch1"]
primarySite <- allAnnot[, "characteristics_ch1"]
histology <- allAnnot[, "characteristics_ch1.1"]
subtype <- allAnnot[, "characteristics_ch1.2"]

table(sourceName)
```

```
## sourceName
## Academic Lab         ATCC         DSMZ        ECACC        HSRRB
##           10          423          209           48          112
##         ICLC         KCLB     NCI/DCTD        RIKEN
##            6           43            7           59
```

```
table(substr(primarySite, 1, 14))
```

```
##
## primary site:
##           917
```

```
primarySite <- substr(primarySite, 15, nchar(primarySite))

table(substr(histology, 1, 11))
```

```
##
## histology:
##         917
```

```
histology <- substr(histology, 12, nchar(histology))

table(substr(subtype, 1, 20))
```

```
##
##                    histology subtype1:
##                  237                  680
```

```
subtype <- substr(subtype, 21, nchar(subtype))

ccleClinical <- data.frame(GEO.ID = GEO.ID, sourceName = sourceName, primarySite = primarySite,
    histology = histology, subtype = subtype, row.names = cellLineNames)

ccleClinical[1:3, ]
```

```
##           GEO.ID sourceName            primarySite    histology
## 1321N1 GSM886835      ECACC central_nervous_system       glioma
## 143B   GSM886836       ATCC                 bone osteosarcoma
## 22Rv1  GSM886837       ATCC              prostate    carcinoma
##            subtype
## 1321N1 astrocytoma
## 143B
## 22Rv1
```

6 Saving RData and csv Files

Now we save the relevant information to an RData object and to a csv file; the latter for use when we don't trust our internet connection.

```
save(ccleClinical, file = file.path("RDataObjects", "ccleClinical.RData"))

write.csv(ccleClinical, file = file.path("RawData", "CCLE", "Clinical", "ccleClinical.csv"))
```

7 Appendix

7.1 File Location

```
getwd()
```

```
## [1] "\\\\mdadqsfs02/workspace/kabagg/RDPaper/Webpage/ResidualDisease"
```

7.2 SessionInfo

```
sessionInfo()
```

```
## R version 2.15.3 (2013-03-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] GEOquery_2.24.1    Biobase_2.18.0     BiocGenerics_0.4.0
## [4] knitr_1.2
##
## loaded via a namespace (and not attached):
## [1] digest_0.6.3   evaluate_0.4.3 formatR_0.7    RCurl_1.95-4.1
## [5] stringr_0.6.2  tools_2.15.3   XML_3.96-1.1
```

8 References

[1] Barretina J, Caponigro G, Stransky N, Venkatesan K et al. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, **483(7391)**:603-7, 2012. PMID: 22460905.

Filtering Samples from the TCGA Data to Focus on RD

by Keith A. Baggerly

1 Executive Summary

1.1 Introduction

We have data from TCGA for 594 ovarian samples, but these include normal samples, recurrences, and cell lines. We don't want to include these samples in our comparisons. We want to identify the high-grade serous ovarian tumors with residual disease (RD) information to focus the question more precisely.

1.2 Methods

Starting with the previously assembled tables of clinical information and expression (for the sample names), we examine the various columns and see which clinical features would justify exclusion from the set being examined.

We consider

- Type of Sample, excluding normal and recurrent disease.
- RD status, excluding samples with no RD information.
- Array Site, excluding samples not coming from the ovary (OV) or peritoneum (PE).
- Neoadjuvant Treatment, excluding samples from patients who received chemotherapy before sample acquisition.
- Grade, excluding samples not Grade 2, 3, or 4.
- Duplication, excluding all but the first occurrenceof any samples remaining deriving from the same patient.

We use these rules to build up a data frame with two columns: sampleUse (Used or Unused), and whyExcluded. We also construct vectors mapping the samples assayed to the clinical information, and the RD status of the samples (as opposed to patients).

1.3 Results

We exclude 103 of the 594 samples for various reasons. Of the 491 that remain, 378 are RD and 113 are No RD.

We save tcgaFilteredSamples, tcgaSampleClinicalMapping, and tcgaSampleRD to the RData file “tcgaFilteredSamples.RData”.

2 Libraries

We first load the libraries we will use in this report.

3 Loading the Data

Here we simply load the previously assembled clinical information and expression matrices, and skim the first line of the clinical information to see what variables exist for filtering the samples.

```
load(file.path("RDataObjects", "tcgaClinical.RData"))
load(file.path("RDataObjects", "tcgaExpression.RData"))
tcgaClinical[1, ]
```

```
##             age_at_initial_pathologic_diagnosis
## TCGA-04-1331                                  78
##             anatomic_organ_subdivision
## TCGA-04-1331           [Not Available]
##                                bcr_patient_uuid date_of_form_completion
## TCGA-04-1331 6d10d4ee-6331-4bba-93bc-a7b64cc0b22a              2009-03-26
##             date_of_initial_pathologic_diagnosis days_to_birth
## TCGA-04-1331                           2004-00-00        -28848
##             days_to_death days_to_initial_pathologic_diagnosis
```

```
## TCGA-04-1331         1336                                    0
##              days_to_last_followup eastern_cancer_oncology_group
## TCGA-04-1331                  1224                [Not Available]
##                       ethnicity gender gynecologic_figo_staging_system
## TCGA-04-1331 NOT HISPANIC OR LATINO FEMALE                 [Not Available]
##                     histological_type          icd_10 icd_o_3_histology
## TCGA-04-1331 Serous Cystadenocarcinoma [Not Available]            8441/3
##              icd_o_3_site informed_consent_verified
## TCGA-04-1331        C56.9                       YES
##              initial_pathologic_diagnosis_method   jewish_origin
## TCGA-04-1331                      [Not Available] [Not Available]
##              karnofsky_performance_score lymphatic_invasion
## TCGA-04-1331             [Not Available]               YES
##              neoplasm_histologic_grade patient_id
## TCGA-04-1331                        G3       1331
##              performance_status_scale_timing person_neoplasm_cancer_status
## TCGA-04-1331                [Not Available]                     WITH TUMOR
##              pretreatment_history  race  residual_tumor tissue_source_site
## TCGA-04-1331                   NO WHITE [Not Available]                  4
##              tumor_histologic_subtype tumor_residual_disease tumor_stage
## TCGA-04-1331       Cystadenocarcinoma               1-10 mm        IIIC
##              tumor_tissue_site venous_invasion vital_status
## TCGA-04-1331             OVARY             NO     DECEASED
```

4 Filtering Samples Used

We now walk through the various criteria, and seeing what these imply for inclusion of the various samples. Our default assumption is that all samples are used.

```
sampleUse <- rep("Used", ncol(tcgaExpression))
names(sampleUse) <- colnames(tcgaExpression)

whyExcluded <- rep("", ncol(tcgaExpression))
names(whyExcluded) <- colnames(tcgaExpression)
```

We also define a mapping between the samples run and the patients from which they were derived, to let us go from the expression data (on samples) to the clinical data (on patients) and vice-versa.

```
sampleClinicalMapping <- match(substr(colnames(tcgaExpression), 1, 12), rownames(tcgaClinical))
names(sampleClinicalMapping) <- names(sampleUse)
```

4.1 Type of Sample

First, we check the type of sample. We want to focus on primary tumors, not normal samples or recurrences.

```
table(tcgaSampleInfo[, "sampleTypeText"])
```

```
##
##   normalTissue   primaryTumor recurrentTumor
##              8            569             17
```

```
sampleUse[tcgaSampleInfo[, "sampleTypeText"] == "normalTissue"] <- "Unused"
sampleUse[tcgaSampleInfo[, "sampleTypeText"] == "recurrentTumor"] <- "Unused"

whyExcluded[tcgaSampleInfo[, "sampleTypeText"] == "normalTissue"] <-
paste(whyExcluded[tcgaSampleInfo[,
    "sampleTypeText"] == "normalTissue"], "-normalTissue-", sep = "")
whyExcluded[tcgaSampleInfo[, "sampleTypeText"] == "recurrentTumor"] <-
paste(whyExcluded[tcgaSampleInfo[,
    "sampleTypeText"] == "recurrentTumor"], "-recurrentTumor-", sep = "")

table(sampleUse)
```

```
## sampleUse
## Unused   Used
##     25    569
```

4.2 Residual Disease

Now we check residual disease status, and exclude samples with no information.

```
tcgaSampleRD <- rep("", ncol(tcgaExpression))
names(tcgaSampleRD) <- colnames(tcgaExpression)
sampleClinicalMapping <- match(substr(colnames(tcgaExpression), 1, 12), rownames(tcgaClinical))
names(sampleClinicalMapping) <- names(tcgaSampleRD)
tcgaSampleRD <- tcgaRD[sampleClinicalMapping]
names(tcgaSampleRD) <- names(sampleClinicalMapping)

table(tcgaSampleRD, useNA = "ifany")
```

```
## tcgaSampleRD
## No RD    RD  <NA>
##   121   401    72
```

```
sampleUse[is.na(tcgaSampleRD)] <- "Unused"
whyExcluded[is.na(tcgaSampleRD)] <- paste(whyExcluded[is.na(tcgaSampleRD)],
    "-No RD Info-", sep = "")

table(sampleUse)
```

```
## sampleUse
## Unused   Used
##     88    506
```

4.3 Array Site

Next, we look at the site the sample was taken from (the “tissue site”). We want tumors from the ovary or the peritoneum.

```
table(tcgaClinical[, "tumor_tissue_site"])
```

```
##
##            OMENTUM              OVARY PERITONEUM (OVARY)
##                  2                572                  2
```

```
badSamples <- which(is.element(sampleClinicalMapping, which(tcgaClinical[, "tumor_tissue_site"] ==
    "OMENTUM")))

sampleUse[badSamples] <- "Unused"

whyExcluded[badSamples] <- paste(whyExcluded[badSamples], "-Not OV or PE-",
    sep = "")

table(sampleUse)
```

```
## sampleUse
## Unused   Used
##     90    504
```

4.4 Neoadjuvant Chemo

Next, we look at whether the patients received neoadjuvant chemotherapy. We want to focus on chemo-naive tumors.

```
table(tcgaClinical[, "pretreatment_history"])
```

```
##
##  NO YES
## 574   2
```

```
badSamples <- which(is.element(sampleClinicalMapping, which(tcgaClinical[, "pretreatment_history"]
==
    "YES")))

sampleUse[badSamples] <- "Unused"

whyExcluded[badSamples] <- paste(whyExcluded[badSamples], "-NeoAdj Chemo-",
    sep = "")

table(sampleUse)
```

```
## sampleUse
## Unused   Used
##     90    504
```

4.5 Grade

Next, we look at grade. We want only Grade 2 or higher samples.

```
table(tcgaClinical[, "neoplasm_histologic_grade"])
```

```
##
## [Not Available]              G1              G2              G3
##               4               6              69             486
##              G4              GB              GX
##               1               1               9
```

```
badSamples <- which(is.element(sampleClinicalMapping, which(tcgaClinical[,
"neoplasm_histologic_grade"] ==
    "[Not Available]")))

sampleUse[badSamples] <- "Unused"

whyExcluded[badSamples] <- paste(whyExcluded[badSamples], "-Grade NA-", sep = "")

badSamples <- which(is.element(sampleClinicalMapping, which(tcgaClinical[,
"neoplasm_histologic_grade"] ==
    "G1")))

sampleUse[badSamples] <- "Unused"

whyExcluded[badSamples] <- paste(whyExcluded[badSamples], "-Grade 1-", sep = "")

badSamples <- which(is.element(sampleClinicalMapping, which(tcgaClinical[,
"neoplasm_histologic_grade"] ==
    "GB")))

sampleUse[badSamples] <- "Unused"

whyExcluded[badSamples] <- paste(whyExcluded[badSamples], "-Grade GB-", sep = "")

badSamples <- which(is.element(sampleClinicalMapping, which(tcgaClinical[,
"neoplasm_histologic_grade"] ==
    "GX")))

sampleUse[badSamples] <- "Unused"
```

```
whyExcluded[badSamples] <- paste(whyExcluded[badSamples], "-Grade GX-", sep = "")

table(sampleUse)
```

```
## sampleUse
## Unused   Used
##    102    492
```

4.6 Check Samples Used for Duplicates

Finally, we check to see if any of the samples remaining in the “Used” category appear more than once, and if so, which ones.

```
namesOfSamplesUsed <- names(sampleUse)[sampleUse == "Used"]
which(duplicated(substr(namesOfSamplesUsed, 1, 12)))
```

```
## [1] 443
```

```
namesOfSamplesUsed[which(duplicated(substr(namesOfSamplesUsed, 1, 12)))]
```

```
## [1] "TCGA-23-1023-01R-01R-0808-01"
```

```
which(substr(namesOfSamplesUsed, 1, 12) == "TCGA-23-1023")
```

```
## [1]  98 443
```

```
namesOfSamplesUsed[which(substr(namesOfSamplesUsed, 1, 12) == "TCGA-23-1023")]
```

```
## [1] "TCGA-23-1023-01A-02R-0434-01" "TCGA-23-1023-01R-01R-0808-01"
```

```
sampleUse["TCGA-23-1023-01R-01R-0808-01"] <- "Unused"
whyExcluded["TCGA-23-1023-01R-01R-0808-01"] <- "-duplicate sample-"

table(sampleUse)
```

```
## sampleUse
## Unused   Used
##    103    491
```

One duplicate sample remains. We arbitrarily keep just the first one.

4.7 Final Tally

Now we see how many RD and No RD samples remain.

```
table(sampleUse, tcgaSampleRD, useNA = "ifany")
```

```
##          tcgaSampleRD
## sampleUse No RD  RD <NA>
##    Unused     8  23   72
##    Used     113 378    0
```

There are 491 samples left, 113 from patients with No RD, and 378 from patients with RD.

5 Building the Data Frame

Now we bundle the assembled information into a data frame for later use.

```
tcgaFilteredSamples <- data.frame(sampleUse = sampleUse, whyExcluded = whyExcluded,
    row.names = colnames(tcgaExpression))
```

6 Saving RData

Now we save the relevant information to an RData object.

```
tcgaSampleClinicalMapping <- sampleClinicalMapping

save(tcgaFilteredSamples, tcgaSampleRD, tcgaSampleClinicalMapping, file =
file.path("RDataObjects",
    "tcgaFilteredSamples.RData"))
```

7 Appendix

7.1 File Location

```
getwd()
```

```
## [1] "/Users/slt/SLT WORKSPACE/EXEMPT/OVARIAN/Ovarian residual disease study 2012/RD
manuscript/Web page for paper/Webpage"
```

7.2 SessionInfo

```
sessionInfo()
```

```
## R version 3.0.2 (2013-09-25)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] knitr_1.5
##
## loaded via a namespace (and not attached):
## [1] evaluate_0.5.1 formatR_0.9    stringr_0.6.2  tools_3.0.2
```

Filtering Samples from the Tothill Data to Focus on RD

by Keith A. Baggerly

1 Executive Summary

1.1 Introduction

Tothill et al. profiled 285 ovarian tumor samples, but not all of the patients had the same type of disease, or had residual disease (RD) information recorded. We want to identify the high-grade serous ovarian tumors with RD information to focus the question more precisely.

1.2 Methods

Starting with the previously assembled table of clinical information, we examine the various columns and see which clinical features would justify exclusion from the set being examined.

We consider

- RD status, excluding samples with no RD information.
- Clinical Type, excluding low malignant potential (LMP) samples.
- Histologic Subtype, excluding non-serous (Adeno and Endo) samples.
- Array Site, excluding samples not coming from the ovary (OV) or peritoneum (PE).
- Neoadjuvant Treatment, excluding samples from patients who received chemotherapy before sample acquisition.
- Grade, excluding Grade 1 samples.

We use these rules to build up a data frame with two columns: sampleUse (Used or Unused), and whyExcluded.

1.3 Results

We exclude 96 of the 285 samples for various reasons. Of the 189 that remain, 139 are RD and 50 are No RD.

We save tothillFilteredSamples to the RData file “tothillFilteredSamples.RData”.

2 Libraries

We first load the libraries we will use in this report.

3 Loading the Data

Here we simply load the previously assembled clinical information.

```
load(file.path("RDataObjects", "tothillClinical.RData"))
tothillClinical[1:3, ]
```

```
##         GEO.ID SampleID KMeansGroup ClinicalType HistologicSubtype
## X49  GSM249839       49           5          MAL              Ser
## X129 GSM250001      129           1          MAL              Ser
## X146 GSM250000      146          NC          MAL              Ser
##      PrimarySite Stage Grade Age Status Pltx Tax Neo MosToRelapse
## X49           OV   III     3  56      D    Y   N   N            7
## X129          OV   III     3  65      D    Y   N   N            7
## X146          OV   III     3  56     PF    Y   N   N          166
##      MosToDeath ResidDisease ArraySite
## X49           8           <1        OV
## X129         15           >1        PE
## X146        166           >1        OV
```

4 Filtering Samples Used

We now walk through the various criteria, and seeing what these imply for inclusion of the various samples. Our default assumption is that all samples are used.

```
sampleUse <- rep("Used", nrow(tothillClinical))
names(sampleUse) <- rownames(tothillClinical)

whyExcluded <- rep("", nrow(tothillClinical))
names(whyExcluded) <- rownames(tothillClinical)
```

4.1 Residual Disease

First, we check residual disease status, and exclude patients with no information.

```
table(tothillClinical[, "ResidDisease"])
```

```
##
##           <1           >1 macro size NK          nil           NK
##           76           70           18           84           37
```

```
sampleUse[tothillClinical[, "ResidDisease"] == "NK"] <- "Unused"
whyExcluded[tothillClinical[, "ResidDisease"] == "NK"] <- paste(whyExcluded[tothillClinical[,
    "ResidDisease"] == "NK"], "-No RD Info-", sep = "")

table(sampleUse)
```

```
## sampleUse
## Unused   Used
##     37    248
```

4.2 Clinical Type

Next, we look at clinical type. Some of the samples are known to be of low malignant potential (LMP), and we don't want to use them.

```
table(tothillClinical[, "ClinicalType"])
```

```
##
## LMP MAL
##  18 267
```

```
sampleUse[tothillClinical[, "ClinicalType"] == "LMP"] <- "Unused"
whyExcluded[tothillClinical[, "ClinicalType"] == "LMP"] <- paste(whyExcluded[tothillClinical[,
    "ClinicalType"] == "LMP"], "-LMP-", sep = "")

table(sampleUse)
```

```
## sampleUse
## Unused   Used
##     53    232
```

4.3 Histologic Subtype

Next, we look at histologic subtype. We only want to keep serous (Ser) tumor samples.

```
table(tothillClinical[, "HistologicSubtype"])
```

```
##
## Adeno  Endo   Ser
##     1    20   264
```

```
sampleUse[tothillClinical[, "HistologicSubtype"] == "Adeno"] <- "Unused"
sampleUse[tothillClinical[, "HistologicSubtype"] == "Endo"] <- "Unused"

whyExcluded[tothillClinical[, "HistologicSubtype"] == "Adeno"] <-
paste(whyExcluded[tothillClinical[,
    "HistologicSubtype"] == "Adeno"], "-Adeno Subtype-", sep = "")
whyExcluded[tothillClinical[, "HistologicSubtype"] == "Endo"] <-
paste(whyExcluded[tothillClinical[,
    "HistologicSubtype"] == "Endo"], "-Endo Subtype-", sep = "")

table(sampleUse)
```

```
## sampleUse
## Unused   Used
##     68    217
```

4.4 Array Site

Next, we look at the site the sample was taken from (the “array site”“). We want tumors from the ovary or the peritoneum.

```
table(tothillClinical[, "ArraySite"])
```

```
##
##       BN       CO       FT       OM    Other       OV OV or OM       PE
##        1        4        2        2        3      200        1       71
##       UT
##        1
```

```
sampleUse[!is.element(tothillClinical[, "ArraySite"], c("OV", "PE"))] <- "Unused"
whyExcluded[!is.element(tothillClinical[, "ArraySite"], c("OV", "PE"))] <-
paste(whyExcluded[!is.element(tothillClinical[,
    "ArraySite"], c("OV", "PE"))], "-Not OV or PE-", sep = "")

table(sampleUse)
```

```
## sampleUse
## Unused   Used
##     77    208
```

4.5 Neoadjuvant Chemo

Next, we look at whether the patients received neoadjuvant chemotherapy. We want to focus on chemo-naive tumors.

```
table(tothillClinical[, "Neo"])
```

```
##
##       N   Y
##   3 264  18
```

```
sampleUse[tothillClinical[, "Neo"] == ""] <- "Unused"
sampleUse[tothillClinical[, "Neo"] == "Y"] <- "Unused"

whyExcluded[tothillClinical[, "Neo"] == ""] <- paste(whyExcluded[tothillClinical[,
    "Neo"] == ""], "-NeoAdj Unk-", sep = "")
whyExcluded[tothillClinical[, "Neo"] == "Y"] <- paste(whyExcluded[tothillClinical[,
```

```
    "Neo"] == ""], "-NeoAdj Chemo-", sep = "")

table(sampleUse)
```

```
## sampleUse
## Unused   Used
##     91    194
```

With respect to neoadjuvant chemo, we exlude patients who either received therapy or for whom this info is unavailable. This mostly reduces the number of RD samples.

4.6 Grade

Next, we look at grade. We want only Grade 2 or 3 samples.

```
table(tothillClinical[, "Grade"])
```

```
##
##   1   2   3
##  19  97 164
```

```
sampleUse[is.na(tothillClinical[, "Grade"])] <- "Unused"
sampleUse[tothillClinical[, "Grade"] == 1] <- "Unused"

whyExcluded[is.na(tothillClinical[, "Grade"])] <- paste(whyExcluded[is.na(tothillClinical[,
    "Grade"])], "-Grade NA-", sep = "")
whyExcluded[which(tothillClinical[, "Grade"] == 1)] <- paste(whyExcluded[which(tothillClinical[,
    "Grade"] == 1)], "-Grade 1-", sep = "")

table(sampleUse)
```

```
## sampleUse
## Unused   Used
##     96    189
```

4.7 Final Tally

Now we see how many RD and No RD samples remain.

```
table(sampleUse, tothillRD)
```

```
##          tothillRD
## sampleUse No RD  RD
##    Unused    34  25
##    Used      50 139
```

5 Building the Data Frame

Now we bundle the assembled information into a data frame for later use.

```
tothillFilteredSamples <- data.frame(sampleUse = sampleUse, whyExcluded = whyExcluded,
    row.names = rownames(tothillClinical))
```

6 Saving RData

Now we save the relevant information to an RData object.

```
save(tothillFilteredSamples, file = file.path("RDataObjects", "tothillFilteredSamples.RData"))
```

7 Appendix

7.1 File Location

```
getwd()
```

```
## [1] "/Users/slt/SLT WORKSPACE/EXEMPT/OVARIAN/Ovarian residual disease study 2012/RD
manuscript/Web page for paper/Webpage"
```

7.2 SessionInfo

```
sessionInfo()
```

```
## R version 3.0.2 (2013-09-25)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] knitr_1.5
##
## loaded via a namespace (and not attached):
## [1] evaluate_0.5.1 formatR_0.9    stringr_0.6.2  tools_3.0.2
```

8 References

[1] Tothill RW, Tinker AV, George J, Brown R, Fox SB, Lade S, Johnson DS, Trivett MK, Etemadmoghadam D, Locandro B, Traficante N, Fereday S, Hung JA, Chiew YE, Haviv I; Australian Ovarian Cancer Study Group, Gertig D, DeFazio A, Bowtell DD. Novel molecular subtypes of serous and endometrioid ovarian cancer linked to clinical outcome. *Clin Cancer Res*, **14(16):5198-208, 2008.**

Overall Survival Curves for TCGA and Tothill by RD Status

by Susan L. Tucker

```
opts_chunk$set(tidy = TRUE, message = TRUE)
```

1 Executive Summary

1.1 Introduction

The goal of this analysis is to produce Kaplan-Meier curves of overall survival (OS) by residual disease (RD) status for patients included in TCGA and Tothill et al.

1.2 Data & Methods

We use the RData objects containing clinical information created in previous reports (assembleTCGAClinical, assembleTothillClinical). Patients are filtered as described previously (filterTCGASamples, filterTothillSamples). Additional patients are excluded for whom survival information is missing.

Survival times are converted from months to years for the data of Tothill et al.

Kaplan-Meier plots are produced to illustrate OS in patient cohorts. OS is compared between groups using the log-rank test.

Comparisons considered are:

i) TCGA versus Tothill et al.

ii) Within each dataset by the RD categories provided in the original data sources.

iii) Within each dataset, any RD compared to no RD.

iv) Within each dataset, by FABP4 expression.

1.3 Results

Three patients are excluded from the filtered cohort of Tothill et al. because of missing survival information.

OS is essentially identical in TCGA and Tothill et al.

Within each data set, OS differs significantly by RD status, using both the RD categories provided or comparing any RD to no RD.

In each data set, OS is worse among the 25% of patients with the highest expression levels of FABP4. The difference reaches statistical significance in Tothill et al.

2 Loading & Filtration of Data

The data objects are loaded.

```
load(file.path("RDataObjects", "tcgaClinical.RData"))
load(file.path("RDataObjects", "tcgaFilteredSamples.RData"))
load(file.path("RDataObjects", "tcgaExpression.RData"))

load(file.path("RDataObjects", "tothillClinical.RData"))
load(file.path("RDataObjects", "tothillFilteredSamples.RData"))
load(file.path("RDataObjects", "tothillExpression.RData"))
```

Filtrations are applied to the TCGA data.

```
rownames(tcgaFilteredSamples)[1:2]
```

```
## [1] "TCGA-13-0758-01A-01R-0362-01" "TCGA-09-0364-01A-02R-0362-01"
```

```
rownames(tcgaClinical)[1:2]
```

```
## [1] "TCGA-04-1331" "TCGA-04-1332"
```

```
rownames(tcgaOSYrs)[1:2]
```

```
## [1] "TCGA-04-1331" "TCGA-04-1332"
```

```
colnames(tcgaExpression[, 1:2])
```

```
## [1] "TCGA-13-0758-01A-01R-0362-01" "TCGA-09-0364-01A-02R-0362-01"
```

```
tcgaSampleUseLong <- rownames(tcgaFilteredSamples[which(tcgaFilteredSamples[,
    "sampleUse"] == "Used"), ])
tcgaSampleUse <- substr(tcgaSampleUseLong, 1, 12)
length(tcgaSampleUse)
```

```
## [1] 491
```

```
length(unique(tcgaSampleUse))
```

```
## [1] 491
```

```
tcgaOSYrsUse <- tcgaOSYrs[tcgaSampleUse, ]
summary(tcgaOSYrsUse)
```

```
##       time           status
##  Min.   : 0.025   Min.   :0.000
##  1st Qu.: 0.936   1st Qu.:0.000
##  Median : 2.323   Median :1.000
##  Mean   : 2.652   Mean   :0.532
##  3rd Qu.: 3.760   3rd Qu.:1.000
##  Max.   :12.666   Max.   :1.000
```

```
tcgaClinUse <- tcgaClinical[tcgaSampleUse, ]
tcgaRDUse <- tcgaRD[tcgaSampleUse]
table(tcgaRDUse)
```

```
## tcgaRDUse
## No RD    RD
##   113   378
```

```
tcgaExpressionUse <- tcgaExpression[, tcgaSampleUseLong]
colnames(tcgaExpressionUse) <- tcgaSampleUse
```

Filtrations are applied to the data of Tothill et al. and survival times are converted from months to years.

```
rownames(tothillFilteredSamples)[1:2]
```

```
## [1] "X49"  "X129"
```

```
rownames(tothillClinical)[1:2]
```

```
## [1] "X49"  "X129"
```

```
rownames(tothillOSMos)[1:2]
```

```
## [1] "X49"  "X129"
```

```
colnames(tothillExpression[, 1:2])
```

```
## [1] "X60120" "X32117"
```

```
tothillSampleUseTmp <- rownames(tothillFilteredSamples[which(tothillFilteredSamples[,
    "sampleUse"] == "Used"), ])
length(tothillSampleUseTmp)
```

```
## [1] 189
```

```
summary(tothillOSMos[tothillSampleUseTmp, ])
```

```
##       time           status
##  Min.   :  0.0   Min.   :0.000
##  1st Qu.: 18.0   1st Qu.:0.000
##  Median : 27.0   Median :0.000
##  Mean   : 30.7   Mean   :0.455
##  3rd Qu.: 41.0   3rd Qu.:1.000
##  Max.   :166.0   Max.   :1.000
##  NA's   :3
```

```
tothillSampleUse <- intersect(tothillSampleUseTmp, rownames(tothillOSMos[!is.na(tothillOSMos[,
    1]), ]))
length(tothillSampleUse)
```

```
## [1] 186
```

```
tothillOSYrsUse <- tothillOSMos[tothillSampleUse, ]
tothillOSYrsUse[, 1] <- tothillOSYrsUse[, 1]/12
tothillClinUse <- tothillClinical[tothillSampleUse, ]
tothillRDUse <- tothillRD[tothillSampleUse]
table(tothillRDUse)
```

```
## tothillRDUse
## No RD    RD
##    50   136
```

```
tothillExpressionUse <- tothillExpression[, tothillSampleUse]
```

3 Analyses

Overall survival is compared in TCGA versus Tothill et al.

```
tmp <- rbind(tcgaOSYrsUse, tothillOSYrsUse)
```

```r
library(survival)
```

```
## Loading required package: splines
```

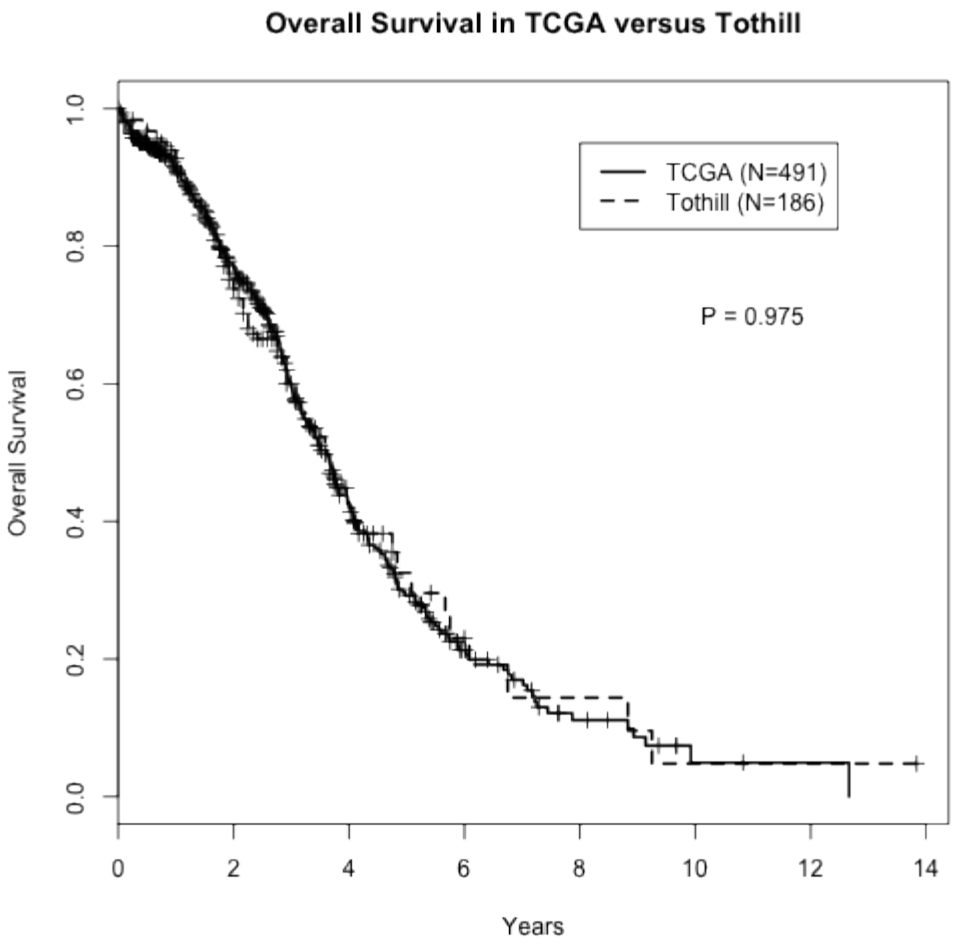
```r
osAll <- Surv(tmp[, 1], tmp[, 2] == 1)

cohort <- rep(2, dim(osAll)[1])
cohort[1:dim(tcgaOSYrsUse)[1]] <- 1
table(cohort)
```

```
## cohort
##   1   2
## 491 186
```

```r
fit <- survfit(osAll ~ cohort)
survdiff(osAll ~ cohort)
```

```
## Call:
## survdiff(formula = osAll ~ cohort)
##
##            N Observed Expected (O-E)^2/E (O-E)^2/V
## cohort=1 491      261    260.7  0.000242  0.000993
## cohort=2 186       86     86.3  0.000731  0.000993
##
##  Chisq= 0  on 1 degrees of freedom, p= 0.975
```

```r
plot(fit, lty = c(1, 2), xlab = "Years", ylab = "Overall Survival", lwd = 2,
    main = "Overall Survival in TCGA versus Tothill")
legend(x = 8, y = 0.95, legend = c("TCGA (N=491)", "Tothill (N=186)"), lty = c(1,
    2), lwd = 2)
text(11, 0.7, "P = 0.975")
```



Overall survival by residual disease status is plotted for the TCGA data.

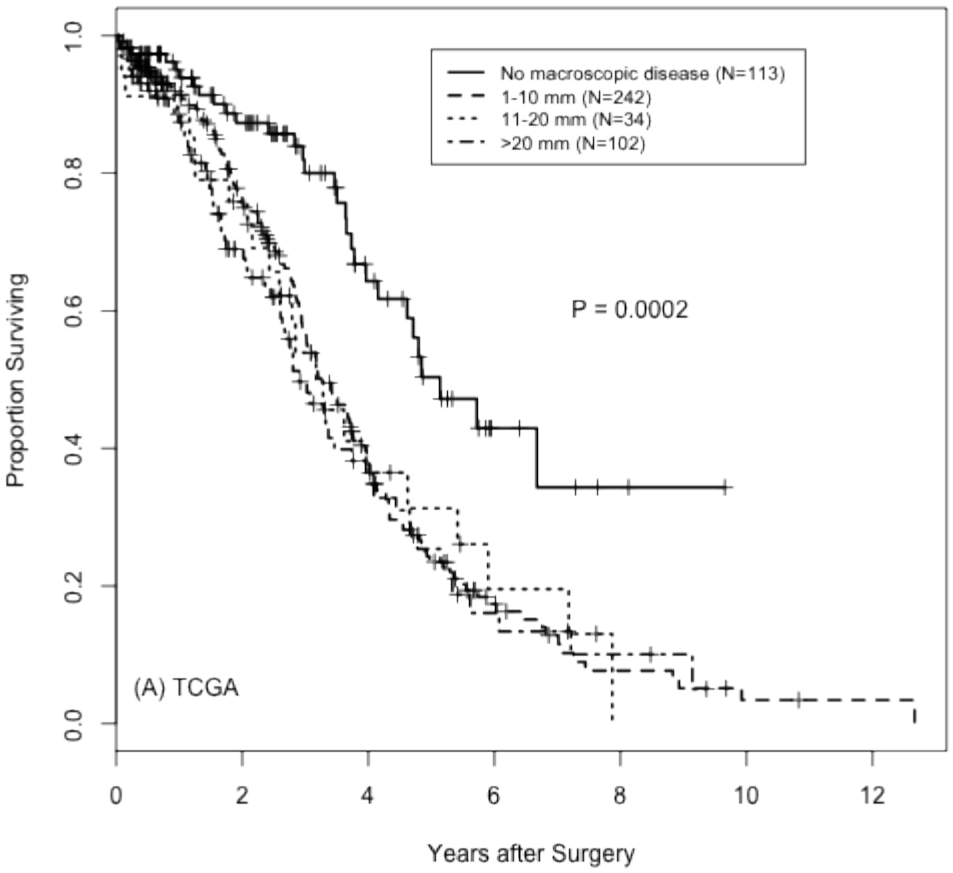
```
table(tcgaClinUse$tumor_residual_disease)
```

```
##
##        [Not Available]                >20 mm               1-10 mm
##                      0                   102                   242
##              11-20 mm No Macroscopic disease
##                     34                   113
```

```
tcgaGp <- rep(1, dim(tcgaOSYrsUse)[1])
tcgaGp[which(tcgaClinUse[, "tumor_residual_disease"] == "1-10 mm")] <- 2
tcgaGp[which(tcgaClinUse[, "tumor_residual_disease"] == "11-20 mm")] <- 3
tcgaGp[which(tcgaClinUse[, "tumor_residual_disease"] == ">20 mm")] <- 4

survTCGA <- Surv(tcgaOSYrsUse[, 1], tcgaOSYrsUse[, 2] == 1)
tcgaSurvFit <- survfit(survTCGA ~ tcgaGp)
survdiff(survTCGA ~ tcgaGp)
```

```
## Call:
## survdiff(formula = survTCGA ~ tcgaGp)
##
##            N Observed Expected (O-E)^2/E (O-E)^2/V
## tcgaGp=1 113       30     59.9    14.951    19.507
## tcgaGp=2 242      147    131.5     1.830     3.726
## tcgaGp=3  34       23     20.6     0.281     0.306
## tcgaGp=4 102       61     49.0     2.949     3.647
##
##  Chisq= 20.1  on 3 degrees of freedom, p= 0.000162
```

```
plot(tcgaSurvFit, lty = 1:4, xlab = "Years after Surgery", ylab = "Proportion Surviving",
    lwd = 2)
legend(x = 5, y = 0.98, legend = c("No macroscopic disease (N=113)", "1-10 mm (N=242)",
    "11-20 mm (N=34)", ">20 mm (N=102)"), lty = c(1:4), lwd = 2, cex = 0.8)
text(0.05, 0.05, "(A) TCGA", pos = 4)
text(7, 0.6, "P = 0.0002", pos = 4)
```



Overall survival by residual disease status is plotted for the data of Tothill et al.

```
table(tothillClinUse$ResidDisease)
```

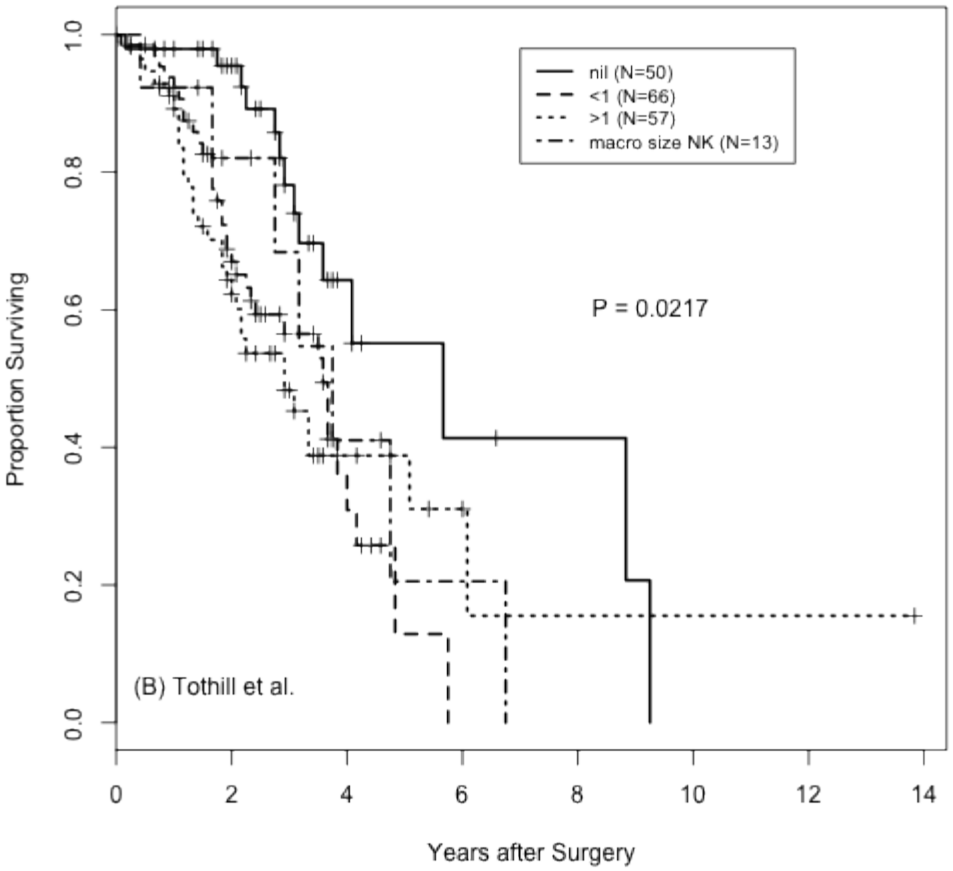
```
##
##           <1           >1 macro size NK          nil           NK
##           66           57           13           50            0
```

```
tothillGp <- rep(1, dim(tothillOSYrsUse)[1])
tothillGp[which(tothillClinUse[, "ResidDisease"] == "<1")] <- 2
tothillGp[which(tothillClinUse[, "ResidDisease"] == ">1")] <- 3
tothillGp[which(tothillClinUse[, "ResidDisease"] == "macro size NK")] <- 4

survTothill <- Surv(tothillOSYrsUse[, 1], tothillOSYrsUse[, 2] == 1)
tothillSurvFit <- survfit(survTothill ~ tothillGp)
survdiff(survTothill ~ tothillGp)
```

```
## Call:
## survdiff(formula = survTothill ~ tothillGp)
##
```

```
##              N Observed Expected (O-E)^2/E (O-E)^2/V
## tothillGp=1 50       14    26.97   6.23891   9.36475
## tothillGp=2 66       34    27.17   1.71681   2.64353
## tothillGp=3 57       31    25.10   1.38667   2.00101
## tothillGp=4 13        7     6.76   0.00872   0.00971
##
##  Chisq= 9.7  on 3 degrees of freedom, p= 0.0217
```

```
plot(tothillSurvFit, lty = 1:4, xlab = "Years after Surgery", ylab = "Proportion Surviving",
    lwd = 2)
legend(x = 7, y = 0.98, legend = c("nil (N=50)", "<1 (N=66)", ">1 (N=57)", "macro size NK
(N=13)"),
    lty = c(1:4), lwd = 2, cex = 0.8)
text(0.05, 0.05, "(B) Tothill et al.", pos = 4)
text(8, 0.6, "P = 0.0217", pos = 4)
```



For each data set, patients with any RD are compared to patients without RD. We do this first for the TCGA data.

```
table(tcgaRDUse)
```

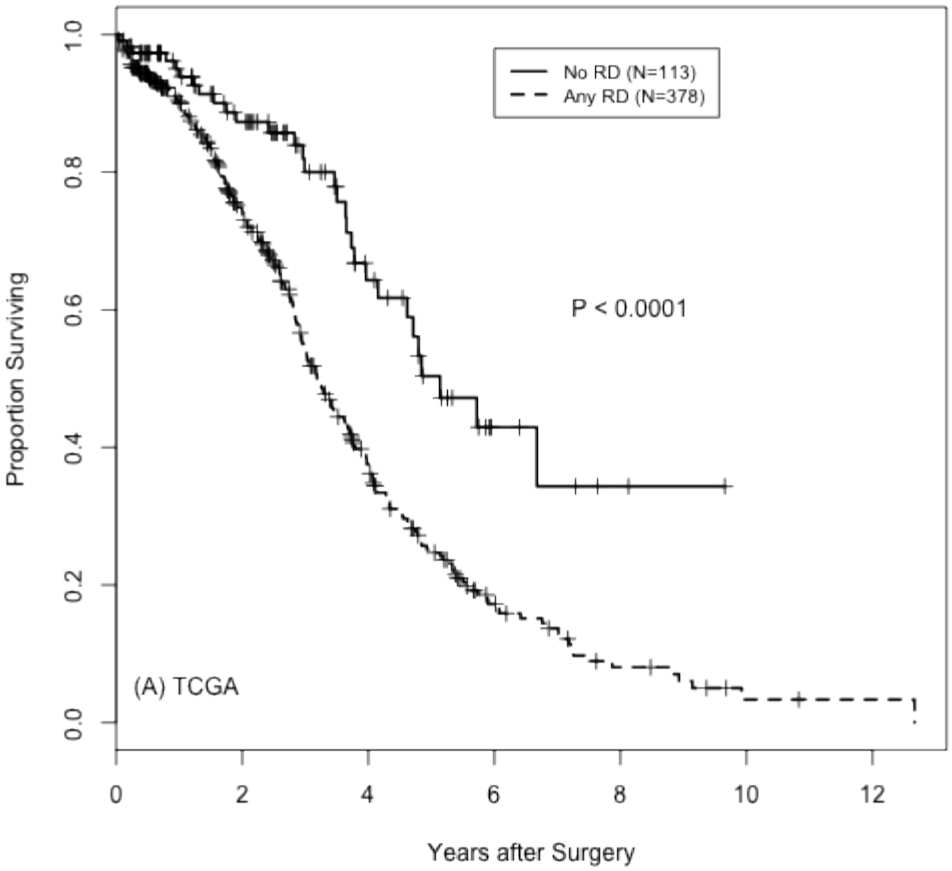
```
## tcgaRDUse
## No RD    RD
##   113   378
```

```
tcgaSurvFit <- survfit(survTCGA ~ tcgaRDUse)
survdiff(survTCGA ~ tcgaRDUse)
```

```
## Call:
```

```
## survdiff(formula = survTCGA ~ tcgaRDUse)
##
##                   N Observed Expected (O-E)^2/E (O-E)^2/V
## tcgaRDUse=No RD 113       30     59.9     14.95      19.5
## tcgaRDUse=RD    378      231    201.1      4.46      19.5
##
##  Chisq= 19.5  on 1 degrees of freedom, p= 1e-05
```

```
plot(tcgaSurvFit, lty = 1:4, xlab = "Years after Surgery", ylab = "Proportion Surviving",
    lwd = 2)
legend(x = 6, y = 0.98, legend = c("No RD (N=113)", "Any RD (N=378)"), lty = c(1:4),
    lwd = 2, cex = 0.8)
text(0.05, 0.05, "(A) TCGA", pos = 4)
text(7, 0.6, "P < 0.0001", pos = 4)
```



We next do this for the data of Tothill et al.

```
table(tothillRDUse)
```

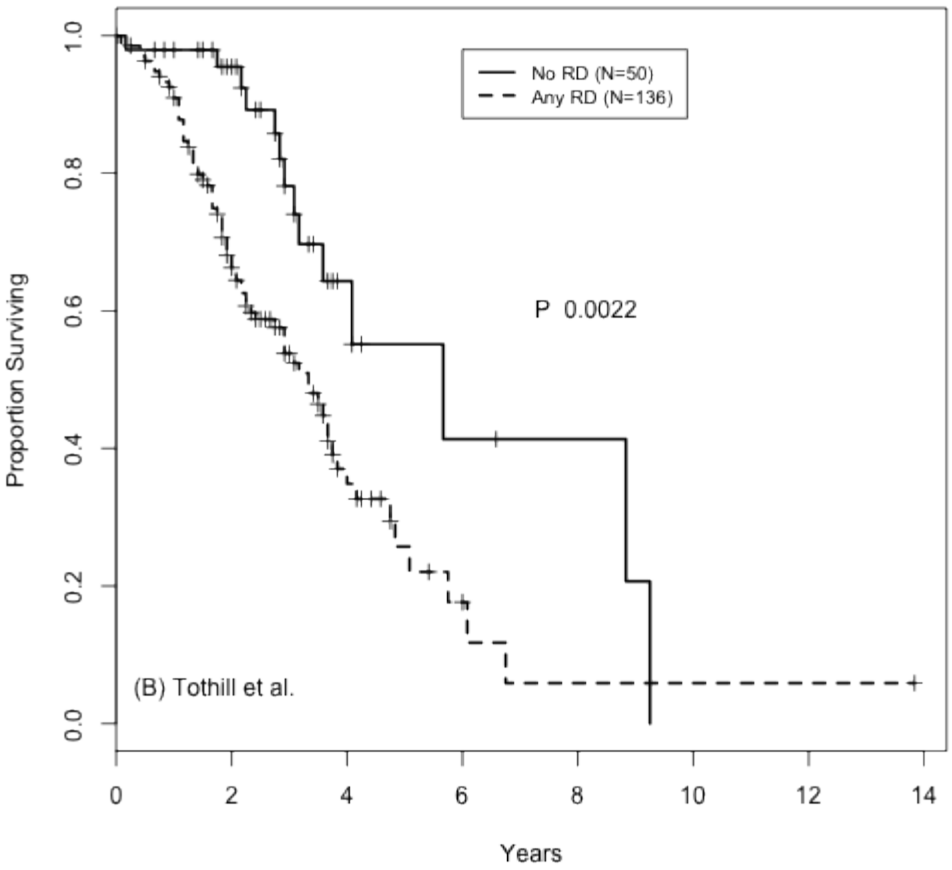
```
## tothillRDUse
## No RD    RD
##    50   136
```

```
tothillSurvFit <- survfit(survTothill ~ tothillRDUse)
survdiff(survTothill ~ tothillRDUse)
```

```
## Call:
## survdiff(formula = survTothill ~ tothillRDUse)
```

```
##
##                      N Observed Expected (O-E)^2/E (O-E)^2/V
## tothillRDUse=No RD  50       14       27      6.24      9.36
## tothillRDUse=RD    136       72       59      2.85      9.36
##
##  Chisq= 9.4  on 1 degrees of freedom, p= 0.00221
```

```
plot(tothillSurvFit, lty = 1:4, xlab = "Years", ylab = "Proportion Surviving",
    lwd = 2)
legend(x = 6, y = 0.98, legend = c("No RD (N=50)", "Any RD (N=136)"), lty = c(1:4),
    lwd = 2, cex = 0.8)
text(0.05, 0.05, "(B) Tothill et al.", pos = 4)
text(7, 0.6, "P  0.0022", pos = 4)
```



We also look at OS in each data set for patients with FABP4 in the top 25% compared to the lower 75%. We begin with TCGA.

```
probeNames <- rownames(tcgaExpressionUse)

library(hthgu133a.db)
```

```
## Loading required package: AnnotationDbi
## Loading required package: BiocGenerics
## Loading required package: parallel
##
## Attaching package: 'BiocGenerics'
##
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
```

```
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
##
## The following object is masked from 'package:stats':
##
##     xtabs
##
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, as.vector, cbind,
##     colnames, duplicated, eval, evalq, Filter, Find, get,
##     intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rep.int, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unlist
##
## Loading required package: Biobase
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
##
## Loading required package: org.Hs.eg.db
## Loading required package: DBI
```

```
geneNames <- unlist(mget(probeNames, hthgu133aSYMBOL))
probesFABP4 <- probeNames[which(geneNames == "FABP4")]
probesFABP4
```

```
## [1] "203980_at"
```

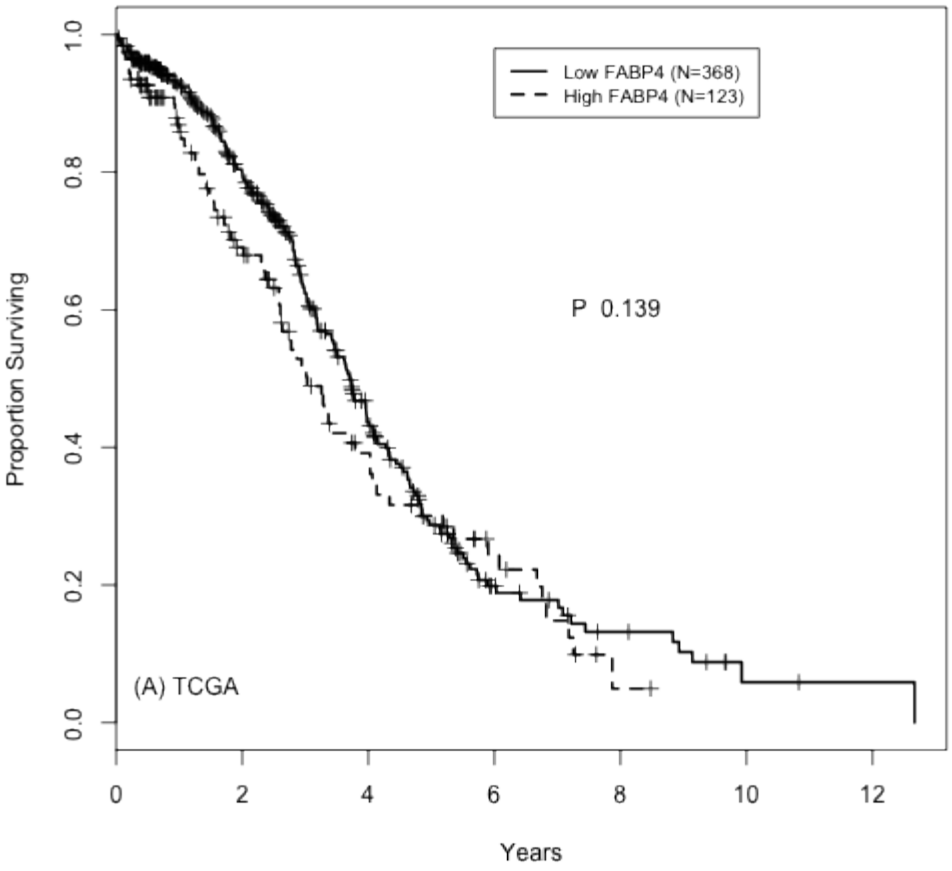
```
tcgaFABP4 <- tcgaExpressionUse[probesFABP4, ]

tcgaFABP4Gp <- rep(0, length(tcgaFABP4))
tcgaFABP4Gp[tcgaFABP4 > quantile(tcgaFABP4, probs = c(0.75))] <- 1
table(tcgaFABP4Gp)
```

```
## tcgaFABP4Gp
##   0   1
## 368 123
```

```
tcgaSurvFit <- survfit(survTCGA ~ tcgaFABP4Gp)
survdiff(survTCGA ~ tcgaFABP4Gp)
```

```
## Call:
## survdiff(formula = survTCGA ~ tcgaFABP4Gp)
##
##                 N Observed Expected (O-E)^2/E (O-E)^2/V
## tcgaFABP4Gp=0 368      190    200.1     0.505      2.19
## tcgaFABP4Gp=1 123       71     60.9     1.659      2.19
##
##  Chisq= 2.2  on 1 degrees of freedom, p= 0.139
```

```
plot(tcgaSurvFit, lty = 1:4, xlab = "Years", ylab = "Proportion Surviving",
    lwd = 2)
legend(x = 6, y = 0.98, legend = c("Low FABP4 (N=368)", "High FABP4 (N=123)"),
    lty = c(1, 2), lwd = 2, cex = 0.8)
text(0.05, 0.05, "(A) TCGA", pos = 4)
text(7, 0.6, "P  0.139", pos = 4)
```



We repeat, using the data of Tothill et al.

```
tothillFABP4 <- tothillExpressionUse[probesFABP4, ]

tothillFABP4Gp <- rep(0, length(tothillFABP4))
tothillFABP4Gp[tothillFABP4 > quantile(tothillFABP4, probs = c(0.75))] <- 1
table(tothillFABP4Gp)
```

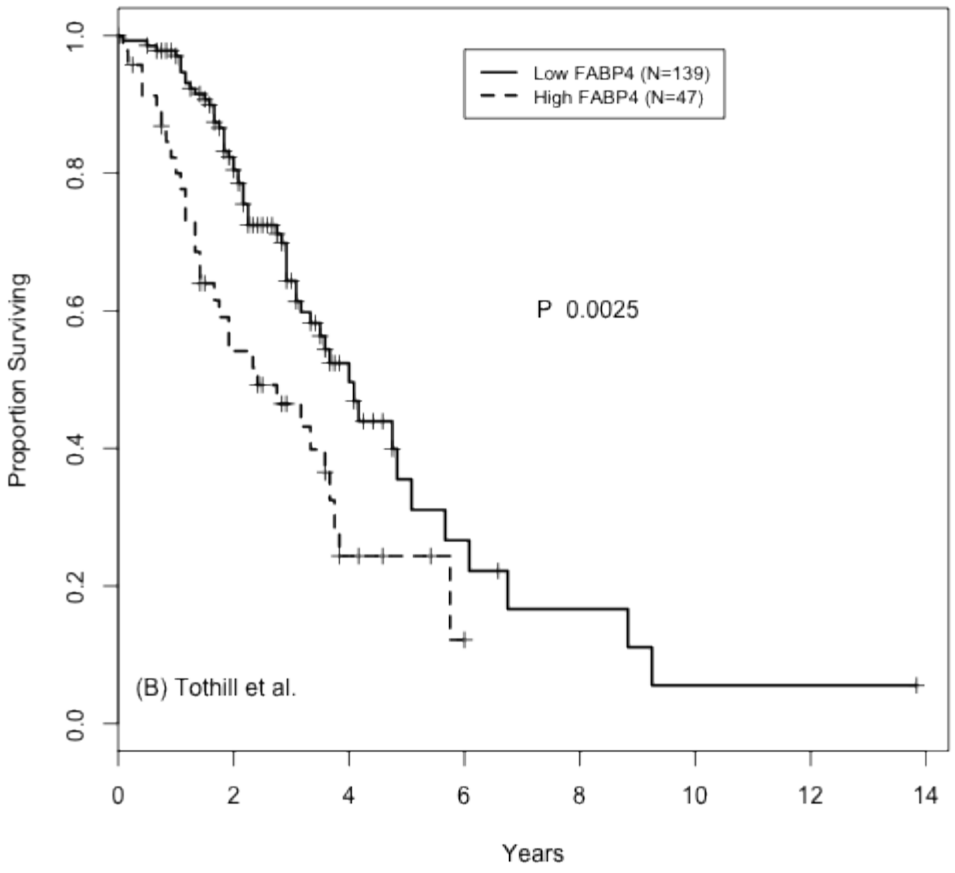
```
## tothillFABP4Gp
##   0   1
## 139  47
```

```
tothillSurvFit <- survfit(survTothill ~ tothillFABP4Gp)
survdiff(survTothill ~ tothillFABP4Gp)
```

```
## Call:
## survdiff(formula = survTothill ~ tothillFABP4Gp)
##
##                     N Observed Expected (O-E)^2/E (O-E)^2/V
## tothillFABP4Gp=0 139       56     67.4      1.92      9.17
## tothillFABP4Gp=1  47       30     18.6      6.97      9.17
##
##  Chisq= 9.2  on 1 degrees of freedom, p= 0.00246
```

```
plot(tothillSurvFit, lty = 1:2, xlab = "Years", ylab = "Proportion Surviving",
    lwd = 2)
legend(x = 6, y = 0.98, legend = c("Low FABP4 (N=139)", "High FABP4 (N=47)"),
    lty = c(1:2), lwd = 2, cex = 0.8)
```

```
text(0.05, 0.05, "(B) Tothill et al.", pos = 4)
text(7, 0.6, "P  0.0025", pos = 4)
```



4 Appendix

4.1 File Location

```
getwd()
```

```
## [1] "/Users/slt/SLT WORKSPACE/EXEMPT/OVARIAN/Ovarian residual disease study 2012/RD
manuscript/Web page for paper/Webpage"
```

4.2 SessionInfo

```
sessionInfo()
```

```
## R version 3.0.2 (2013-09-25)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  splines   stats     graphics  grDevices utils     datasets
## [8] methods   base
```

```
##
## other attached packages:
## [1] hthgu133a.db_2.10.1  org.Hs.eg.db_2.10.1  RSQLite_0.11.4
## [4] DBI_0.2-7            AnnotationDbi_1.24.0 Biobase_2.22.0
## [7] BiocGenerics_0.8.0   survival_2.37-7      knitr_1.5
##
## loaded via a namespace (and not attached):
## [1] evaluate_0.5.1 formatR_0.10   IRanges_1.20.6 stats4_3.0.2
## [5] stringr_0.6.2  tools_3.0.2
```

Flagging Genes (Probesets) Associated with RD in Both TCGA and Tothill

by Keith A. Baggerly

1 Executive Summary

1.1 Introduction

We want to identify genes whose expression shows a strong and similarly directed association with residual disease (RD) status in both the TCGA and Tothill datasets.

1.2 Methods

We load our previously assembled RData files for tcgaExpression, tcgaFilteredData, tothillExpression, tothillFilteredData, and tothillClinical.

We restrict our attention to probesets on both the TCGA and Tothill array platforms.

Then, using just the filtered sets of samples, we contrast RD and No RD samples within each dataset using two sample t-tests. For each probeset in each dataset, we identify the associated gene and record the mean expression level in the RD and No RD groups, the t-statistic, the raw p-value, and the false discovery rate (FDR) adjusted p-value.

We flag probsets that are significantly different in both TCGA and Tothill using (a) a 5% FDR cutoff and (b) a 10% FDR cutoff.

We plot the bivariate t-tests to look for structure, expression heatmaps for the selected probes to look for patterns, correlations between the probes chosen to look for coordinated behavior, and dot and density plots for individual probesets to identify other features. We write a convenience function to make generation of the dot and density plots easier.

1.3 Results

There are 22277 probesets common to the two platforms.

We flag 8 probesets using a 5% FDR cutoff in both datasets, and 47 probesets using a 10% FDR cutoff.

The bivariate plot of the t-statistics found is shown in Figure 1; a zoomed version highlighting the probesets overexpressed in RD samples is shown in Figure 2.

The expression heatmaps for TCGA and Tothill using the 5% FDR cutoffs are in Figures 3 and 4, respectively. The expression heatmaps for TCGA and Tothill using the 10% FDR cutoffs are in Figures 5 and 6, respectively.

Heatmaps of the pairwise 10% FDR probe correlations for TCGA and Tothill are shown in Figures 7 and 8.

Dot and density plots for all 47 probesets passing the 10% FDR filter are saved as “plotsOfTop47Probesets.pdf” in the Reports folder.

Dot and density plots for 6 probesets, corresponding to LUM, DCN, GADD45B, FABP4, ADH1B, and ADIPOQ are shown in Figures 9, 10, 11, 12, 13, and 14, respectively.

We save tcgaCommonUsed, tothillCommonUsed, keyProbesets05pct, keyGenes05pct, keyProbesets10pct, keyGenes10pct, nTCGANoRD, nTCGARD, nTothillNoRD, nTothillRD, plotProbesetResults, and rdTTestResults to the RData file “rdFlaggedGenes.RData”.

1.4 Conclusions

In both sets of expression heatmaps, there is evidence of a molecularly distinct subset of patients (about a third) with a higher chance of having RD. Expression levels for most of the genes identified are consistently higher in these patients.

For LUM, DCN, and GADD45B, which represent the bulk of the probsets showing elevation, what we see is an overall mean shift (values are trending higher) without a clear division point (above here, something's changed). For FABP4, ADH1B, and (to a lesser extent) ADIPOQ, we see a *qualitative* shift in a smaller subset – values for most samples are very low (effectively “off”“), but values for a subset of patients are very high

("on"").

A qualitative difference strikes us as more likely to survive a shift across assays than a mean offset, so we preferentially pursue FABP4 and ADH1B.

2 Libraries

We first load the libraries we will use in this report.

```
library(affy)
library(hthgu133a.db)
library(gplots)
```

3 Loading the Data

Here we simply load the previously assembled RData files.

clinical information and expression matrices, and skim the first line of the clinical information to see what variables exist for filtering the samples.

```
load(file.path("RDataObjects", "tcgaExpression.RData"))
load(file.path("RDataObjects", "tcgaFilteredSamples.RData"))

load(file.path("RDataObjects", "tothillExpression.RData"))
load(file.path("RDataObjects", "tothillFilteredSamples.RData"))
load(file.path("RDataObjects", "tothillClinical.RData"))
```

4 Rearranging Data

4.1 Selecting Common Probesets

We only want to examine probesets evaluated in both datasets.

```
commonProbesets <- intersect(rownames(tcgaExpression), rownames(tothillExpression))
```

4.2 Extracting RD and No RD Samples

Given the common probesets, we next get matrices of data for RD and No RD measurements for both TCGA and Tothill.

We begin with TCGA

```
tcgaCommonRD <- tcgaExpression[commonProbesets, names(tcgaSampleRD)[which((tcgaSampleRD ==
    "RD") & (tcgaFilteredSamples[, "sampleUse"] == "Used"))]]
dim(tcgaCommonRD)
```

```
## [1] 22277   378
```

```
tcgaCommonNoRD <- tcgaExpression[commonProbesets, names(tcgaSampleRD)[which((tcgaSampleRD ==
    "No RD") & (tcgaFilteredSamples[, "sampleUse"] == "Used"))]]
dim(tcgaCommonNoRD)
```

```
## [1] 22277   113
```

Next, we repeat the process for Tothill.

```
tothillCommonRD <- tothillExpression[commonProbesets, names(tothillRD)[which((tothillRD ==
    "RD") & (tothillFilteredSamples[, "sampleUse"] == "Used"))]]
```

```
dim(tothillCommonRD)
```

```
## [1] 22277   139
```

```
tothillCommonNoRD <- tothillExpression[commonProbesets, names(tothillRD)[which((tothillRD ==
    "No RD") & (tothillFilteredSamples[, "sampleUse"] == "Used"))]]
dim(tothillCommonNoRD)
```

```
## [1] 22277    50
```

4.3 Bundling

For later plots, it can be easier to rearrange things yet again.

```
tcgaCommonUsed <- cbind(tcgaCommonNoRD, tcgaCommonRD)
tothillCommonUsed <- cbind(tothillCommonNoRD, tothillCommonRD)
```

5 Contrasting RD with No RD: T-Tests

5.1 Running T-Tests

Our first comparisons involve simple two-sample t-tests. We perform these for TCGA first.

```
d1 <- date()
tcgaTVals <- rep(0, length(commonProbesets))
names(tcgaTVals) <- commonProbesets
tcgaPVals <- tcgaTVals
for (i1 in 1:length(commonProbesets)) {
    tempT <- t.test(tcgaCommonRD[i1, ], tcgaCommonNoRD[i1, ], var.equal = TRUE)
    tcgaTVals[i1] <- tempT[["statistic"]]
    tcgaPVals[i1] <- tempT[["p.value"]]
}
d2 <- date()
c(d1, d2)
```

```
## [1] "Wed Nov 20 11:29:41 2013" "Wed Nov 20 11:29:50 2013"
```

```
tcgaPValsAdj <- p.adjust(tcgaPVals, method = "fdr")
names(tcgaPValsAdj) <- commonProbesets
```

Then we repeat the process with Tothill.

```
d1 <- date()
tothillTVals <- rep(0, length(commonProbesets))
names(tothillTVals) <- commonProbesets
tothillPVals <- tothillTVals
for (i1 in 1:length(commonProbesets)) {
    tempT <- t.test(tothillCommonRD[i1, ], tothillCommonNoRD[i1, ], var.equal = TRUE)
    tothillTVals[i1] <- tempT[["statistic"]]
    tothillPVals[i1] <- tempT[["p.value"]]
}
d2 <- date()
c(d1, d2)
```

```
## [1] "Wed Nov 20 11:29:50 2013" "Wed Nov 20 11:29:59 2013"
```

```
tothillPValsAdj <- p.adjust(tothillPVals, method = "fdr")
```

```
names(tothillPValsAdj) <- commonProbesets
```

5.2 Checking for Overlap at an Extreme Cutoff

We now see which genes (if any) appear significant at an FDR of 5% in both datasets.

```
sum(tcgaPValsAdj < 0.05)
```

```
## [1] 149
```

```
sum(tothillPValsAdj < 0.05)
```

```
## [1] 81
```

```
sum((tothillPValsAdj < 0.05) & (tcgaPValsAdj < 0.05))
```

```
## [1] 8
```

```
sum((tothillPValsAdj < 0.1) & (tcgaPValsAdj < 0.1))
```

```
## [1] 47
```

```
keyProbesets05pct <- names(which((tothillPValsAdj < 0.05) & (tcgaPValsAdj <
    0.05)))
keyProbesets10pct <- names(which((tothillPValsAdj < 0.1) & (tcgaPValsAdj < 0.1)))
keyGenes05pct <- unlist(mget(keyProbesets05pct, hthgu133aSYMBOL))
keyGenes10pct <- unlist(mget(keyProbesets10pct, hthgu133aSYMBOL))
keyGenes05pct
```

```
## 201744_s_at   203666_at   203980_at 207574_s_at   209335_at 209613_s_at
##       "LUM"    "CXCL12"     "FABP4"   "GADD45B"       "DCN"     "ADH1B"
##   209687_at   221541_at
##    "CXCL12"  "CRISPLD2"
```

There are 8 probesets flagged at a common FDR of 5% (listed above), and 47 probesets flagged at a common FDR of 10%.

5.3 Building a Data Frame

Now we bundle our t-test results into a data frame for later reference, sorting the entries by mean fdr-adjusted p-value.

```
tcgaMeanRD <- apply(tcgaCommonRD, 1, mean)
tcgaMeanNoRD <- apply(tcgaCommonNoRD, 1, mean)
tothillMeanRD <- apply(tothillCommonRD, 1, mean)
tothillMeanNoRD <- apply(tothillCommonNoRD, 1, mean)
commonGeneSymbols <- unlist(mget(commonProbesets, hthgu133aSYMBOL))

rdTTestResults <- data.frame(row.names = rownames(tcgaCommonUsed), geneSymbol = commonGeneSymbols,
    tcgaMeanRD = tcgaMeanRD, tcgaMeanNoRD = tcgaMeanNoRD, tcgaTVals = tcgaTVals,
    tcgaPVals = tcgaPVals, tcgaPValsAdj = tcgaPValsAdj, tothillMeanRD = tothillMeanRD,
    tothillMeanNoRD = tothillMeanNoRD, tothillTVals = tothillTVals, tothillPVals = tothillPVals,
    tothillPValsAdj = tothillPValsAdj)
rdTTestResults <- rdTTestResults[order(tcgaPValsAdj + tothillPValsAdj), ]
```

As a check, we look at the results for the top 10 probesets by this ordering.

```
rdTTestResults[1:10, ]
```

```
##             geneSymbol tcgaMeanRD tcgaMeanNoRD tcgaTVals tcgaPVals
## 201744_s_at        LUM      9.057        8.084     4.992 8.318e-07
## 203666_at       CXCL12      4.400        3.921     4.134 4.202e-05
## 203980_at        FABP4      4.278        3.463     3.863 1.272e-04
## 209335_at          DCN      7.376        6.746     3.853 1.324e-04
## 209613_s_at      ADH1B      3.514        2.944     3.681 2.586e-04
## 221541_at     CRISPLD2      6.017        5.444     3.858 1.295e-04
## 209612_s_at      ADH1B      3.694        3.196     3.507 4.947e-04
## 207574_s_at    GADD45B      6.795        6.392     3.741 2.049e-04
## 209687_at       CXCL12      6.885        6.228     3.928 9.811e-05
## 211813_x_at        DCN      8.602        8.003     3.525 4.635e-04
##             tcgaPValsAdj tothillMeanRD tothillMeanNoRD tothillTVals
## 201744_s_at     0.002885        10.325           9.036        4.536
## 203666_at       0.020107         7.778           6.950        4.160
## 203980_at       0.034264         6.422           4.371        4.533
## 209335_at       0.034609         8.687           7.527        4.560
## 209613_s_at     0.044321         4.912           3.091        5.045
## 221541_at       0.034340         7.523           6.433        4.344
## 209612_s_at     0.057726         5.556           3.781        4.968
## 207574_s_at     0.039855         8.276           7.580        4.155
## 209687_at       0.028758         8.178           7.162        3.986
## 211813_x_at     0.056505        10.921           9.857        4.509
##             tothillPVals tothillPValsAdj
## 201744_s_at    1.024e-05        0.010629
## 203666_at      4.842e-05        0.022975
## 203980_at      1.036e-05        0.010629
## 209335_at      9.243e-06        0.010629
## 209613_s_at    1.067e-06        0.002970
## 221541_at      2.289e-05        0.016447
## 209612_s_at    1.521e-06        0.003765
## 207574_s_at    4.937e-05        0.022975
## 209687_at      9.644e-05        0.035837
## 211813_x_at    1.145e-05        0.010629
```

6 Plotting Data

Given the contrast results, we now plot the data in several ways to see if this clarifies aspects of the structure.

6.1 Bivariate t-value Plot

Our first check involves plotting the TCGA and Tothill t-statistics against each other, to see if there are clear outliers or disagreement with respect to sign. The initial plot, Figure 1, shows the vast majority of the probesets selected are more strongly expressed in RD samples. A zoom on the upper quadrant of this plot is shown in Figure 2.

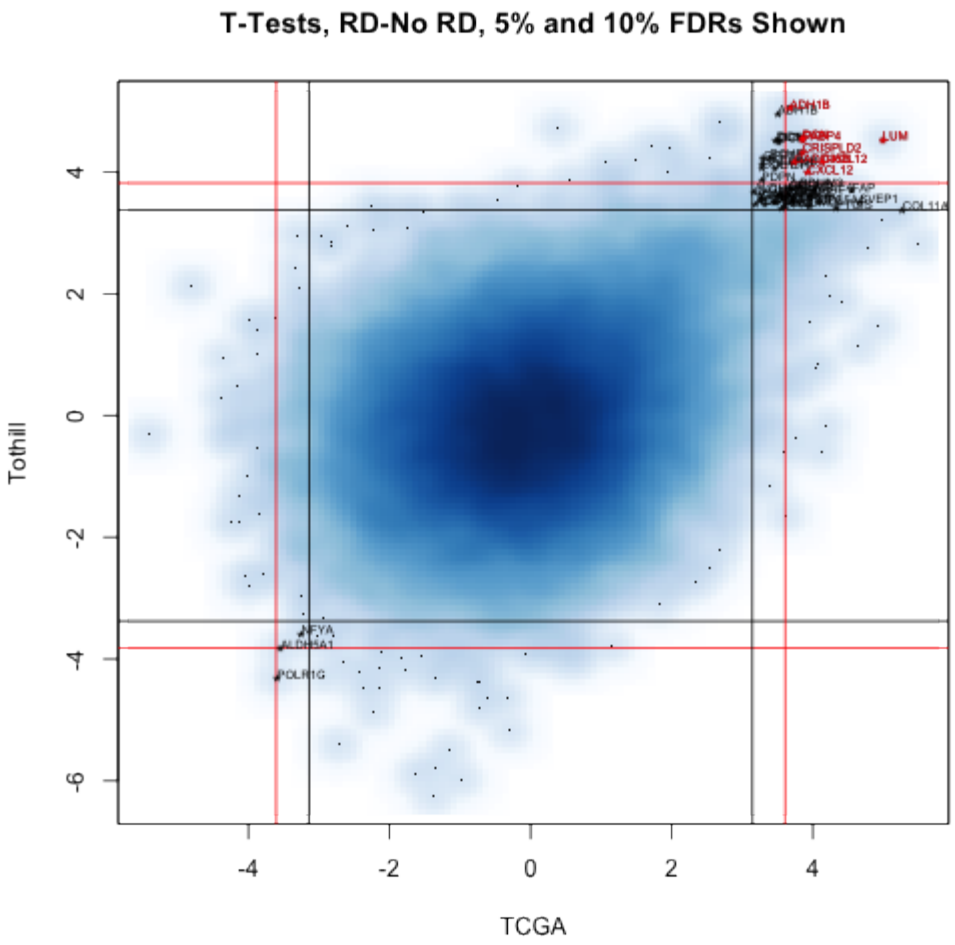


Figure 1: Bivariate plot of two-sample RD-No RD t-values for TCGA and Tothill. The vast majority of the probesets selected show higher expression in RD cases. Lumican (LUM) is the strongest overall.

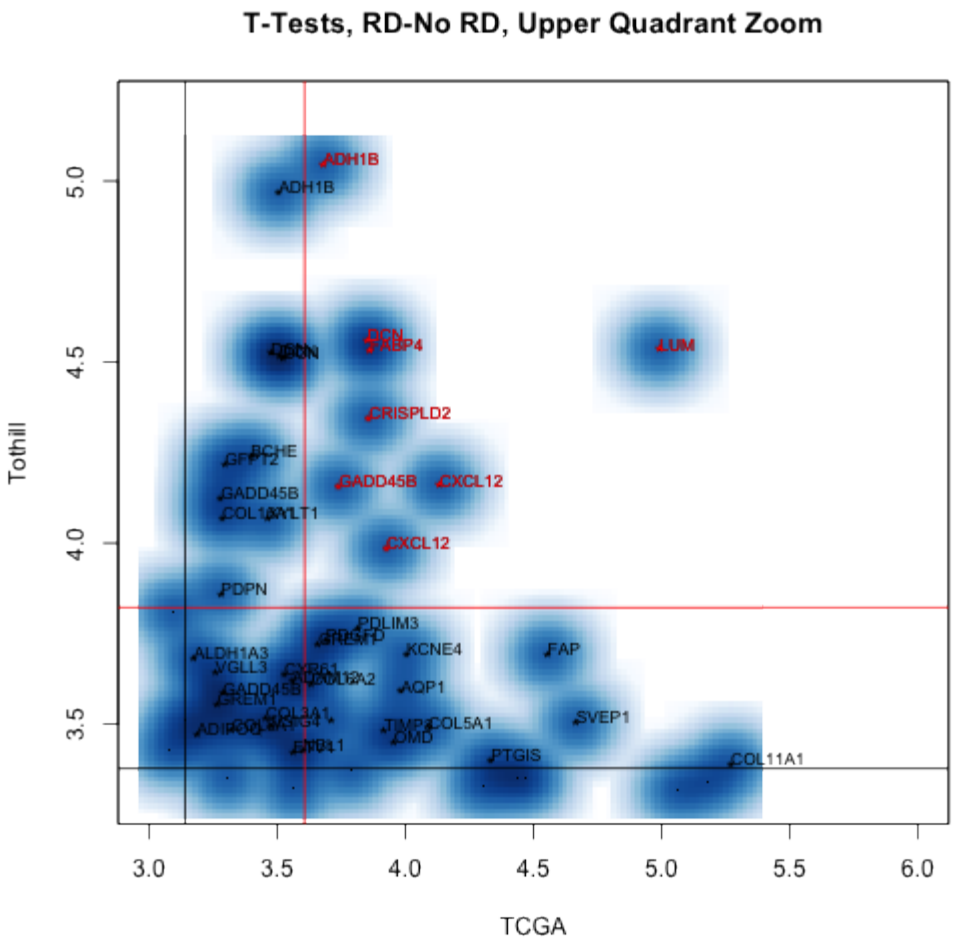


Figure 2: Zoom on the upper quadrant of the bivariate plot of two-sample RD-No RD t-values for TCGA and Tothill, to show the names more clearly.

6.2 Heatmaps of Probesets Flagged by 5% FDR

We want to see if the most clearly chosen probesets are flagging the same samples, and how clearly they divide RD from No RD cases. We check this first for the 8 probesets passing the 5% FDR filters. The TCGA heatmap is shown in Figure 3, and the Tothill heatmap is shown in Figure 4. The general story is the same in both; we see a clear cluster in which most of the probesets are concurrently elevated, and the patients in these clusters have much higher rates of RD. Of the 8 probesets examined, those for FABP4 and ADH1B stand out as telling the story most starkly, though the enrichment rates may not be much higher. We do not see a tight grouping of the No RD cases, but this is to be expected since some cases of RD will not be driven at the molecular level but rather by spatial positioning in the abdomen.

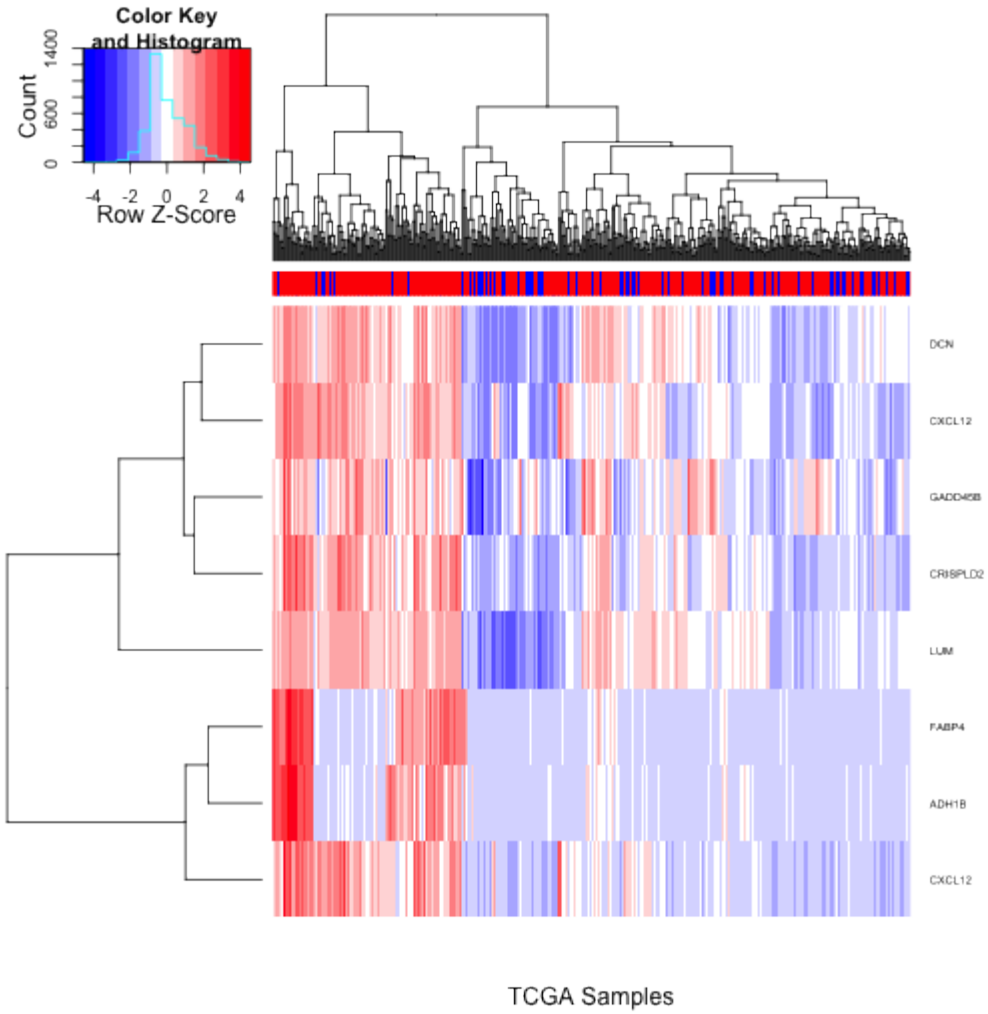


Figure 3: Heatmap of the TCGA Samples using just the 8 probesets passing the 5% FDR filter for both TCGA and Tothill. RD status (Red=RD, Blue=No RD) is indicated in the colorbar at top. There is a clear cluster at the left in which most of these genes are concurrently elevated; the density of RD cases is much higher in this group. Of the probesets shown, FABP4 and ADH1B stand out from the rest in that they show a much more marked "on/off" pattern.

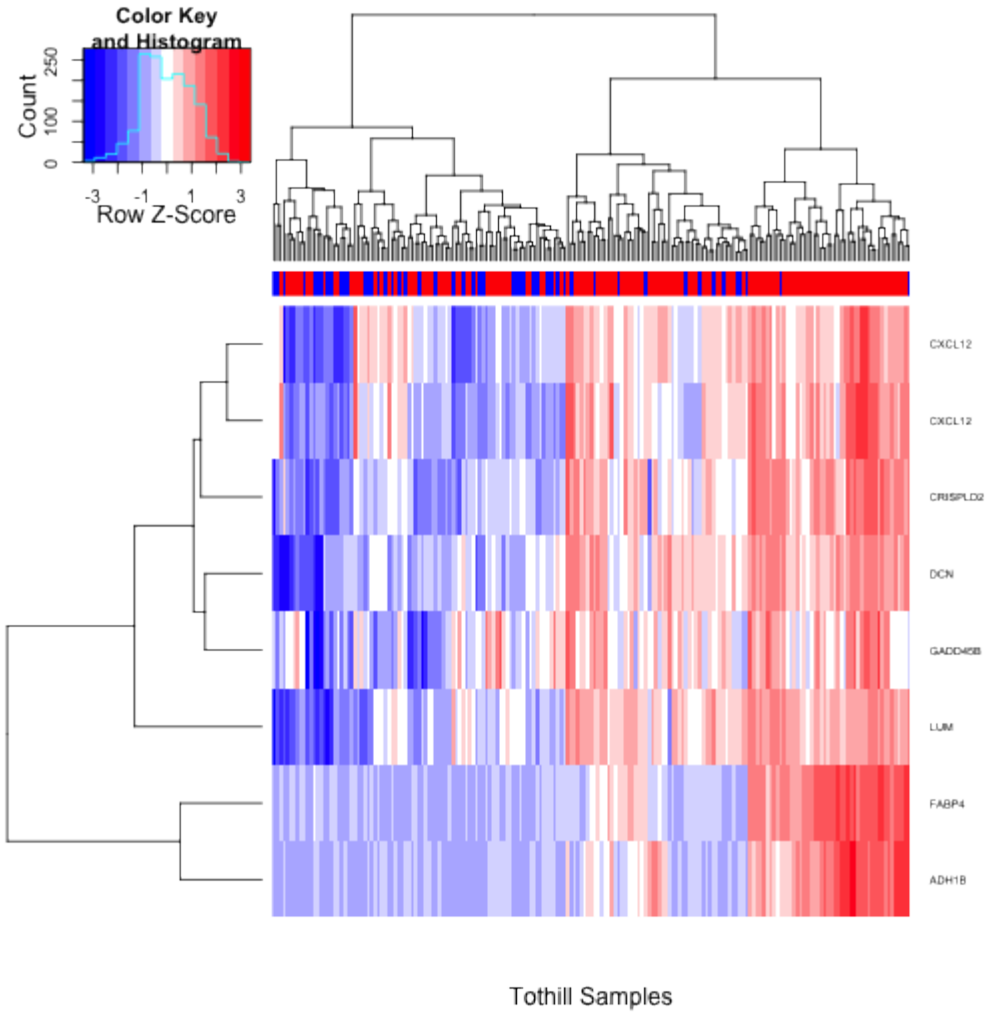


Figure 4: Heatmap of the Tothill Samples using just the 8 probesets passing the 5% FDR filter for both TCGA and Tothill. RD status (Red=RD, Blue=No RD) is indicated in the colorbar at top. The story essentially parallels that for the TCGA data. There is a clear cluster at the right in which most of these genes are concurrently elevated; the density of RD cases is much higher in this group. Of the probesets shown, FABP4 and ADH1B stand out from the rest in that they show a much more marked "on/off" pattern.

6.3 Heatmaps of Probesets Flagged by 10% FDR

Having examined the 5% FDR probesets, we now expand our view to encompass probesets passing a 10% FDR filter in both TCGA and Tothill. The TCGA heatmap is shown in Figure 5, and the Tothill heatmap is shown in Figure 6. One factor that becomes more apparent here is the broadly parallel pattern of overexpression seen for most of the probesets (FABP4 and ADH1B again stand out). This suggests there may be a common driver for many of them; possibly a "pathway" of some type.

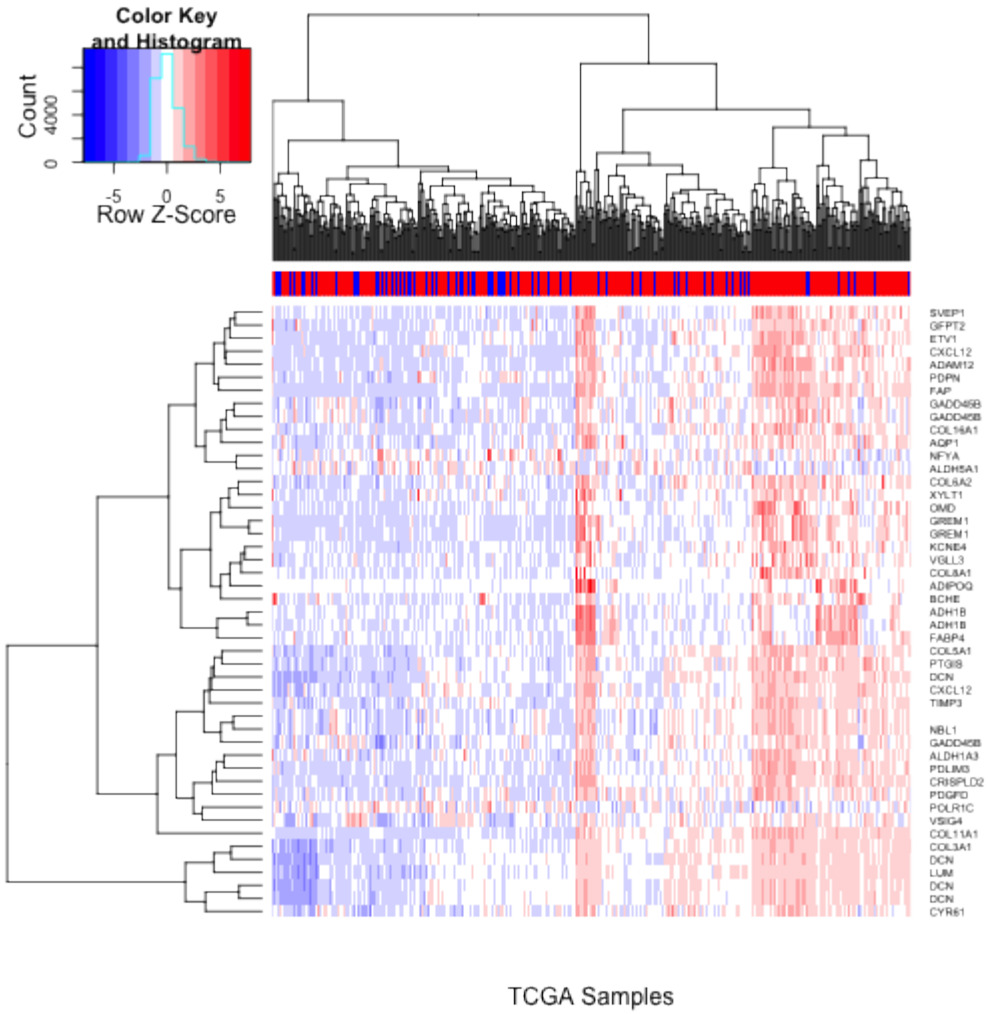


Figure 5: Heatmap of the TCGA Samples using the 47 probesets passing the 10% FDR filter for both TCGA and Tothill. RD status (Red=RD, Blue=No RD) is indicated in the colorbar at top. While FABP4, ADH1B, and, to a lesser extent ADIPOQ again stand out, the main visual impression is one of parallel expression for most of the probesets, suggesting some common underlying driving factor.

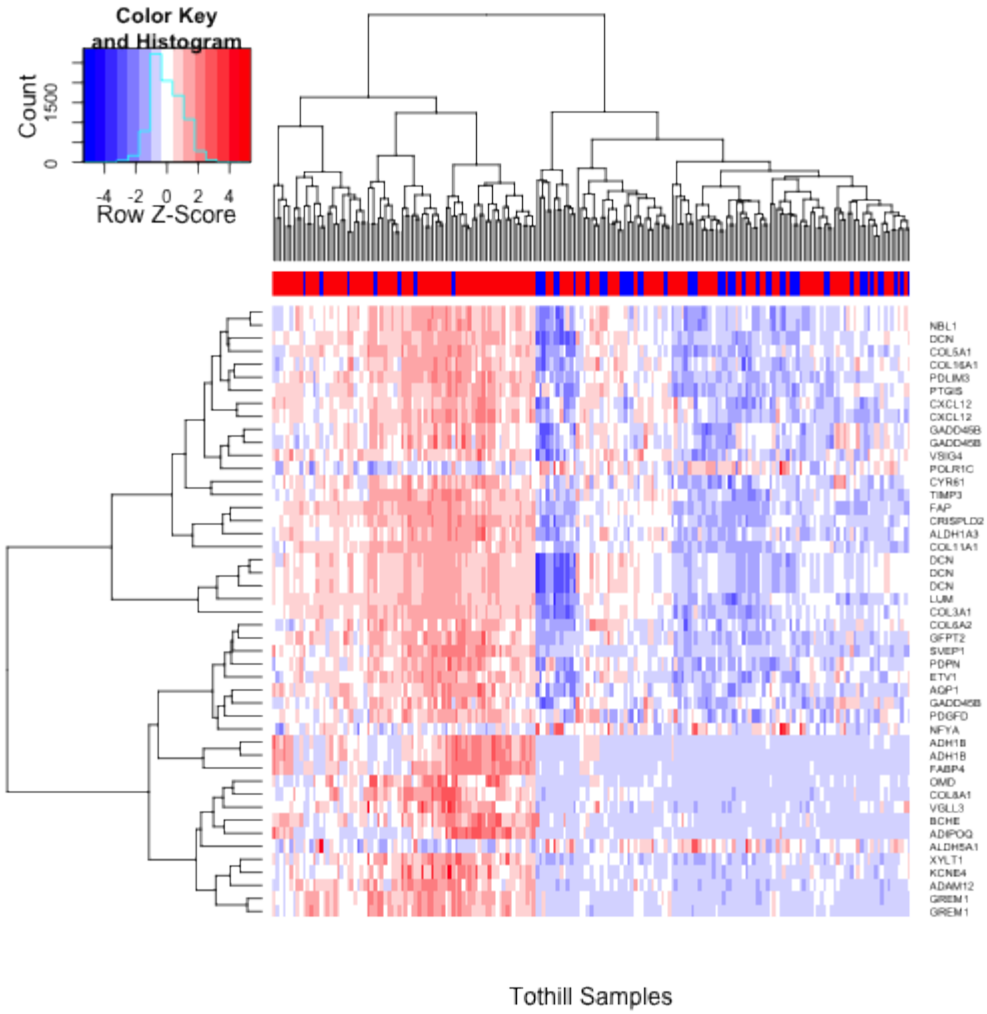


Figure 6: Heatmap of the Tothill Samples using the 47 probesets passing the 10% FDR filter for both TCGA and Tothill. RD status (Red=RD, Blue=No RD) is indicated in the colorbar at top. As with the TCGA data, While FABP4, and ADH1B again stand out, the main visual impression is one of parallel expression for most of the probesets, suggesting some common underlying driving factor.

6.4 Heatmaps of Correlation of Probesets Flagged by 10% FDR

Given the broad parallelism of expression seen in the heatmaps of probesets passing the 10% FDR filters, we want to check the correlation patterns between these probes. The TCGA heatmap is shown in Figure 7, and the Tothill heatmap is shown in Figure 8.

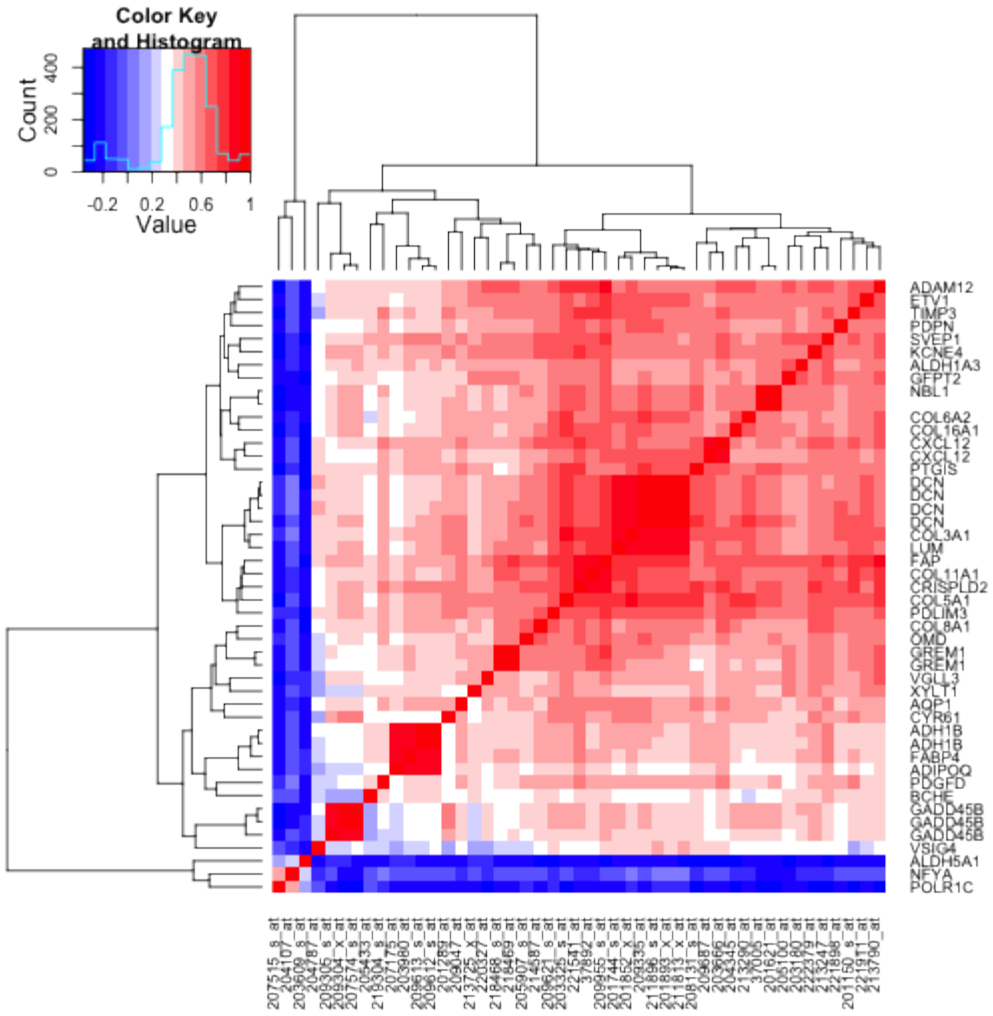


Figure 7: Correlations in the TCGA data between the 47 probesets selected by 10% FDR cutoffs. While the three probesets where expression declines drive the coloring most, the main story in terms of commonality may be the grouping of 26 probesets in the upper right, including lumican (LUM) and decorin (DCN). Cutting the probsets into three clusters overall, FABP4 and ADH1B reside in the second grouping of probesets at the right.

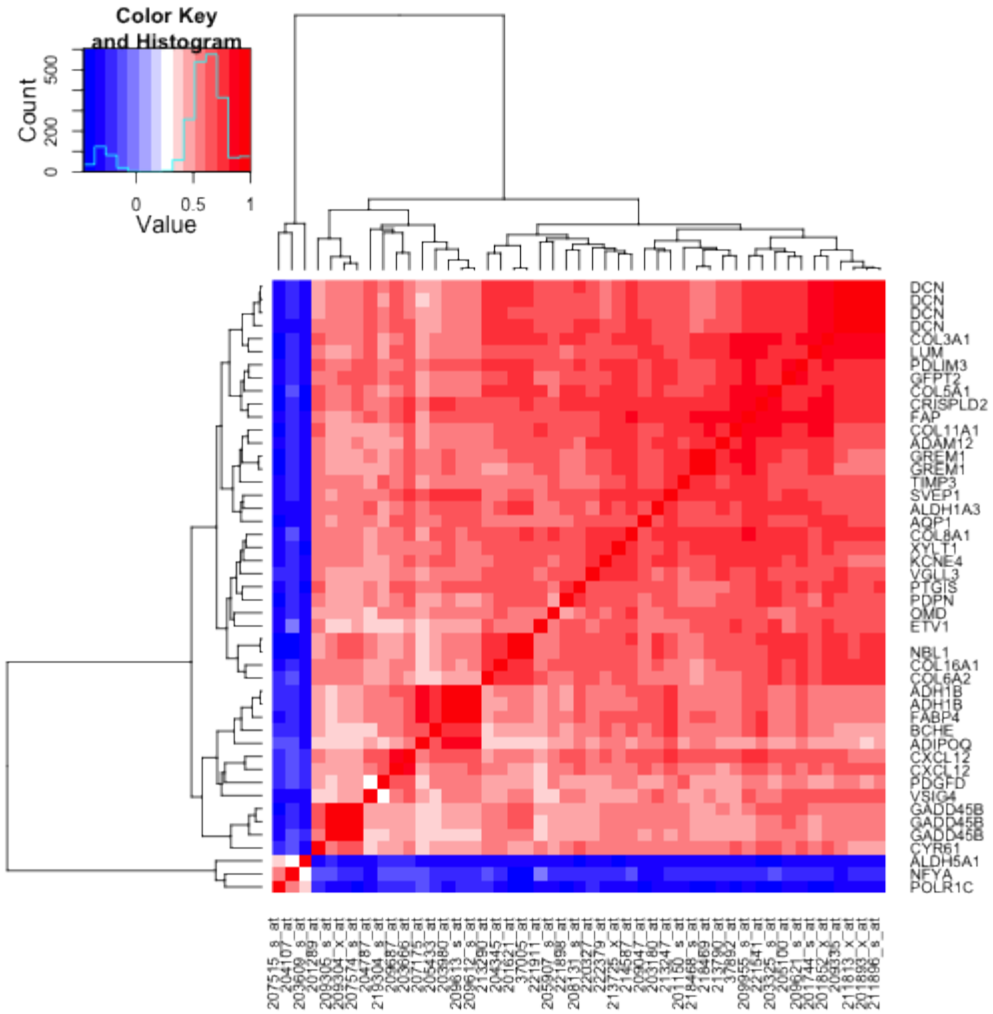


Figure 8: Correlations in the Tothill data between the 47 probesets selected by 10% FDR cutoffs. While the three probesets where expression declines drive the coloring most, the main story in terms of commonality may be the tight grouping of 31 probesets in the upper right, including lumican (LUM) and decorin (DCN). FABP4 and ADH1B reside in the second grouping of probesets at the right.

6.5 Density Plots and Dotplots

We know some of the probesets are of interest. Now we look at aspects of behavior of the individual probesets, specifically dotplots and density plots for each probeset by dataset.

First, we construct a generic function for plotting these results for a given probeset.

```
plotProbesetResults <- function(probesetID) {

    par(mfrow = c(2, 2))

    geneName <- unlist(mget(probesetID, hthgu133aSYMBOL))

    plot(tcgaCommonUsed[probesetID, ], col = c(rep("blue", nTCGANoRD), rep("red",
        nTCGARD)), xlab = "TCGA Samples", ylab = "Expression", main = paste("Expression of",
        geneName, "in TCGA"))
    abline(v = nTCGANoRD + 0.5)

    plot(tothillCommonUsed[probesetID, ], col = c(rep("blue", nTothillNoRD),
        rep("red", nTothillRD)), xlab = "Tothill Samples", ylab = "Expression",
        main = paste("Expression of", geneName, "in Tothill"))
    abline(v = nTothillNoRD + 0.5)

    tempDensTCGA <- density(tcgaCommonUsed[probesetID, ])
    tempDensTCGANoRD <- density(tcgaCommonNoRD[probesetID, ])
    tempDensTCGARD <- density(tcgaCommonRD[probesetID, ])
```

```
    plot(tempDensTCGA[["x"]], tempDensTCGA[["y"]], xlab = paste("Expression of",
        probesetID, "in TCGA"), ylab = "Density", type = "l", main = paste("Density of",
        probesetID, "in TCGA"))
    lines(tempDensTCGANoRD[["x"]], (nTCGANoRD/(nTCGANoRD + nTCGARD)) * tempDensTCGANoRD[["y"]],
        col = "blue")
    lines(tempDensTCGARD[["x"]], (nTCGARD/(nTCGANoRD + nTCGARD)) * tempDensTCGARD[["y"]],
        col = "red")

    tempDensTothill <- density(tothillCommonUsed[probesetID, ])
    tempDensTothillNoRD <- density(tothillCommonNoRD[probesetID, ])
    tempDensTothillRD <- density(tothillCommonRD[probesetID, ])

    plot(tempDensTothill[["x"]], tempDensTothill[["y"]], xlab = paste("Expression of",
        probesetID, "in Tothill"), ylab = "Density", type = "l", main = paste("Density of",
        probesetID, "in Tothill"))
    lines(tempDensTothillNoRD[["x"]], (nTothillNoRD/(nTothillNoRD + nTothillRD)) *
        tempDensTothillNoRD[["y"]], col = "blue")
    lines(tempDensTothillRD[["x"]], (nTothillRD/(nTothillNoRD + nTothillRD)) *
        tempDensTothillRD[["y"]], col = "red")

    par(mfrow = c(1, 1))

}
```

For reference, we produce a pdf file containing the results for all 47 probesets in our top list.

```
pdf(file = file.path("Reports", "plotsOfTop47Probesets.pdf"))

for (i1 in 1:length(keyProbesets10pct)) {
    plotProbesetResults(keyProbesets10pct[i1])
}

dev.off()
```

```
## pdf
##   2
```

We include results for a few selected genes here: LUM (201744_s_at), Figure 9, DCN (211896_s_at), Figure 10, GADD45B (207574_s_at), Figure 11, FABP4 (203980_at), Figure 12, ADH1B (209613_s_at), Figure 13, and ADIPOQ (207175_at), Figure 14. For some of the genes (ADH1B, DCN), mutliple probesets are available but the results appear qualitatively similar to the representative ones chosen.

For LUM, DCN, and GADD45B, which represent the bulk of the probsets showing elevation, what we see is an overall mean shift (values are trending higher) without a clear division point (above here, something's changed). For FABP4, ADH1B, and (to a lesser extent) ADIPOQ, we see a qualitative shift – values for most samples are very low (effectively “off”“), but values for a subset of patients are very high ("on”“). This type of qualitative difference strikes us as more likely to survive a shift across assays than a mean offset, so we will preferentially pursue FABP4 and ADH1B.

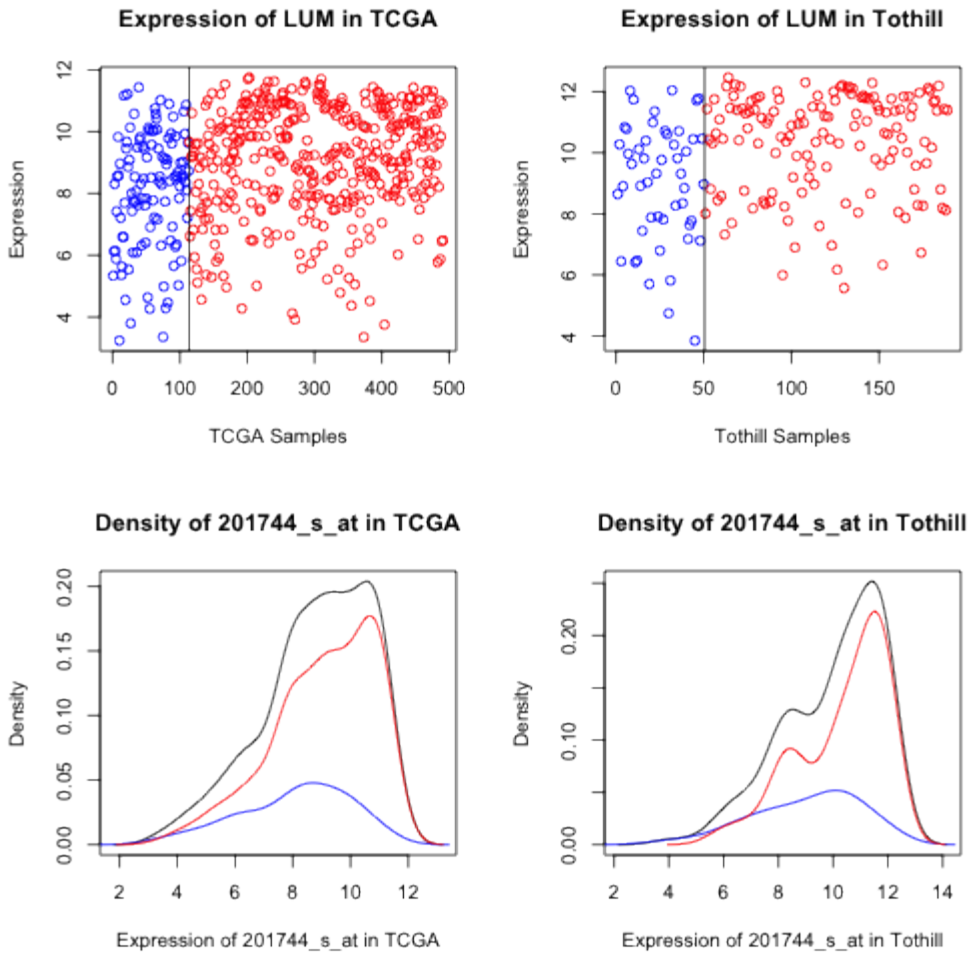


Figure 9: Dot and density plots for lumican (LUM) in TCGA and Tothill. Cases with No RD are blue, RD are red. While there is a clear mean shift (which drives the t-test results), there is not a clearly defined cutpoint.

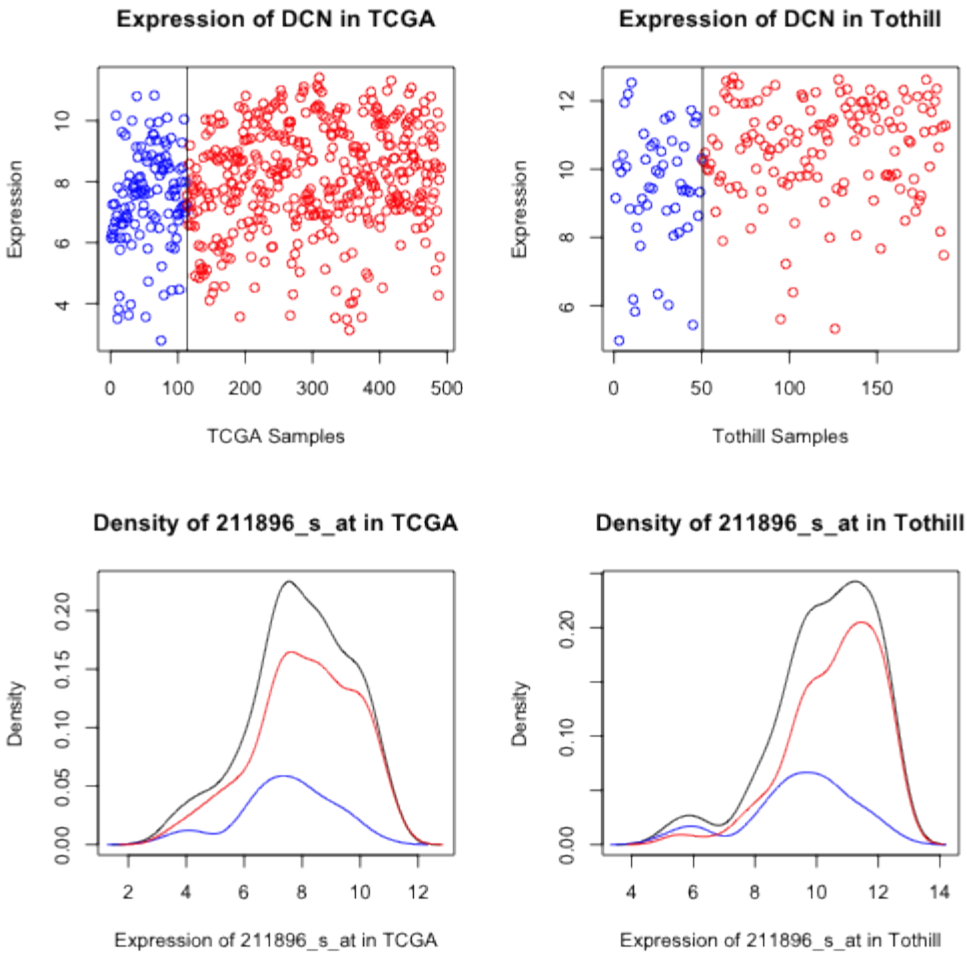


Figure 10: Dot and density plots for decorin (DCN) in TCGA and Tothill. Cases with No RD are blue, RD are red. While there is a clear mean shift (which drives the t-test results), there is not a clearly defined cutpoint.

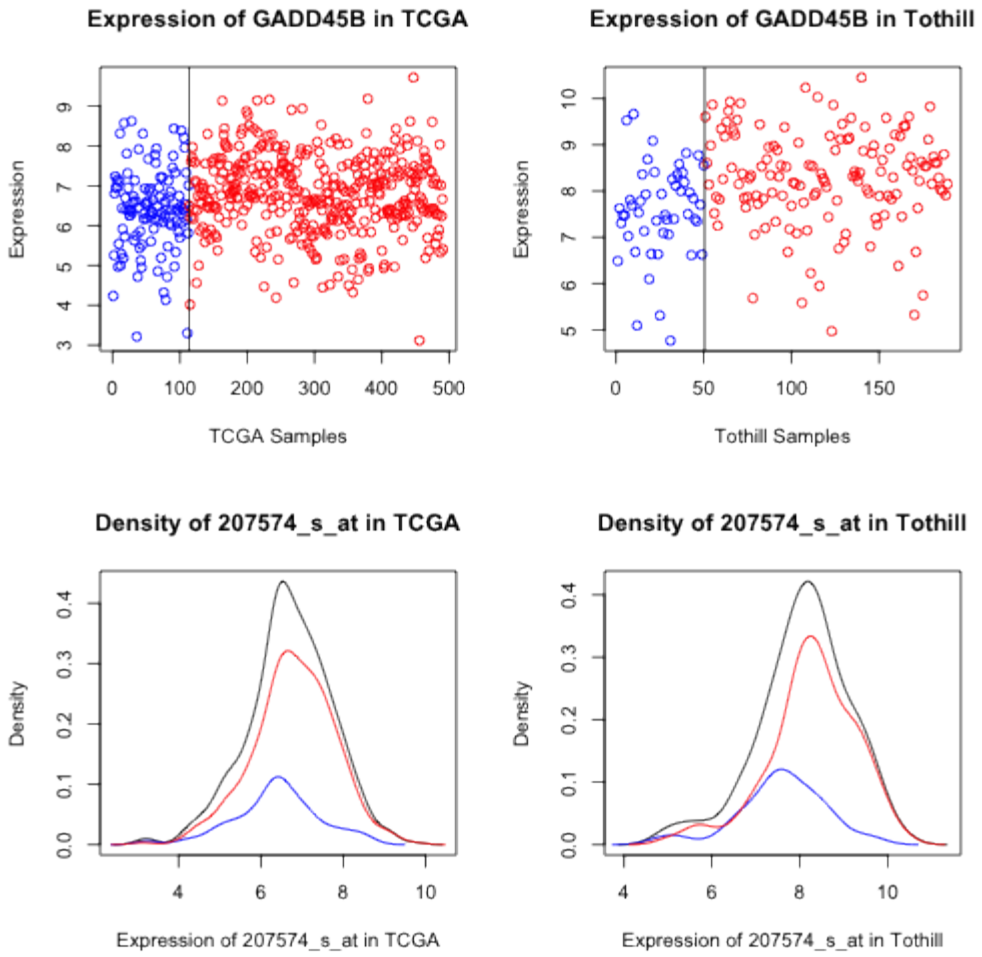


Figure 11: Dot and density plots for GADD45B in TCGA and Tothill. Cases with No RD are blue, RD are red. While there is a clear mean shift (which drives the t-test results), there is not a clearly defined cutpoint.

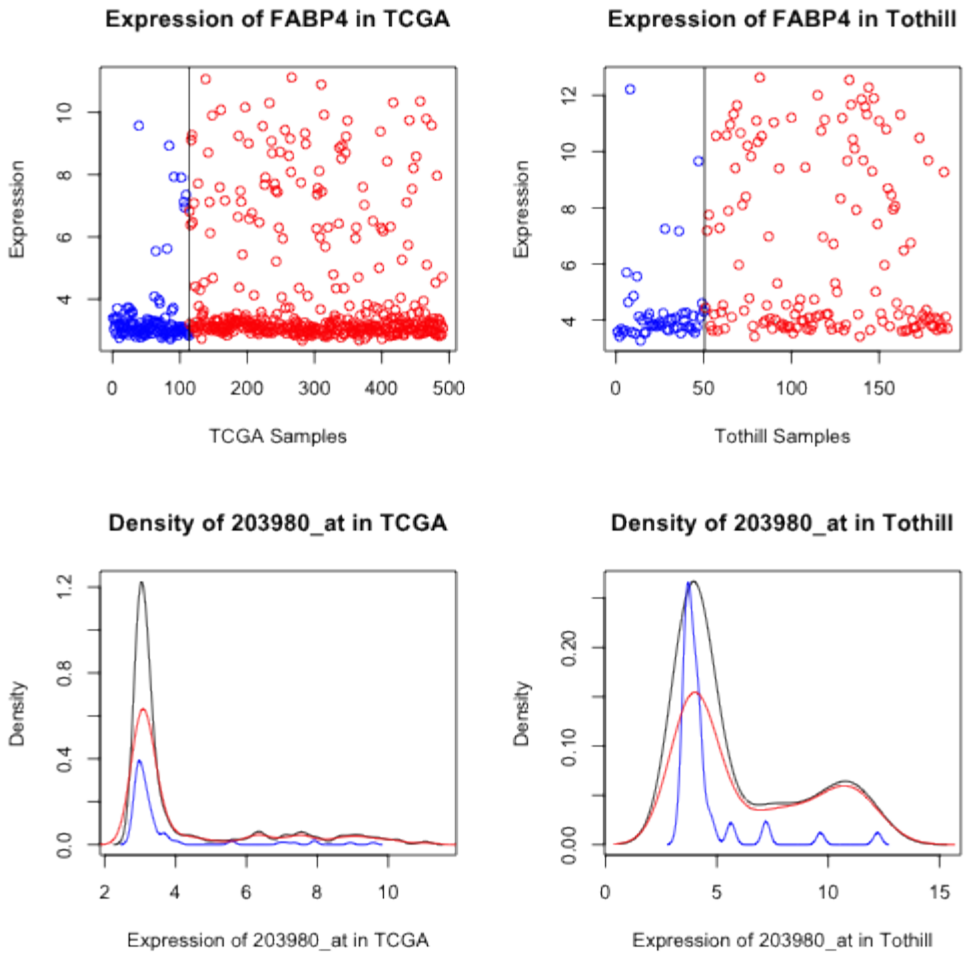


Figure 12: Dot and density plots for FABP4 in TCGA and Tothill. Cases with No RD are blue, RD are red. There is a qualitative shift in a subset of the patients.

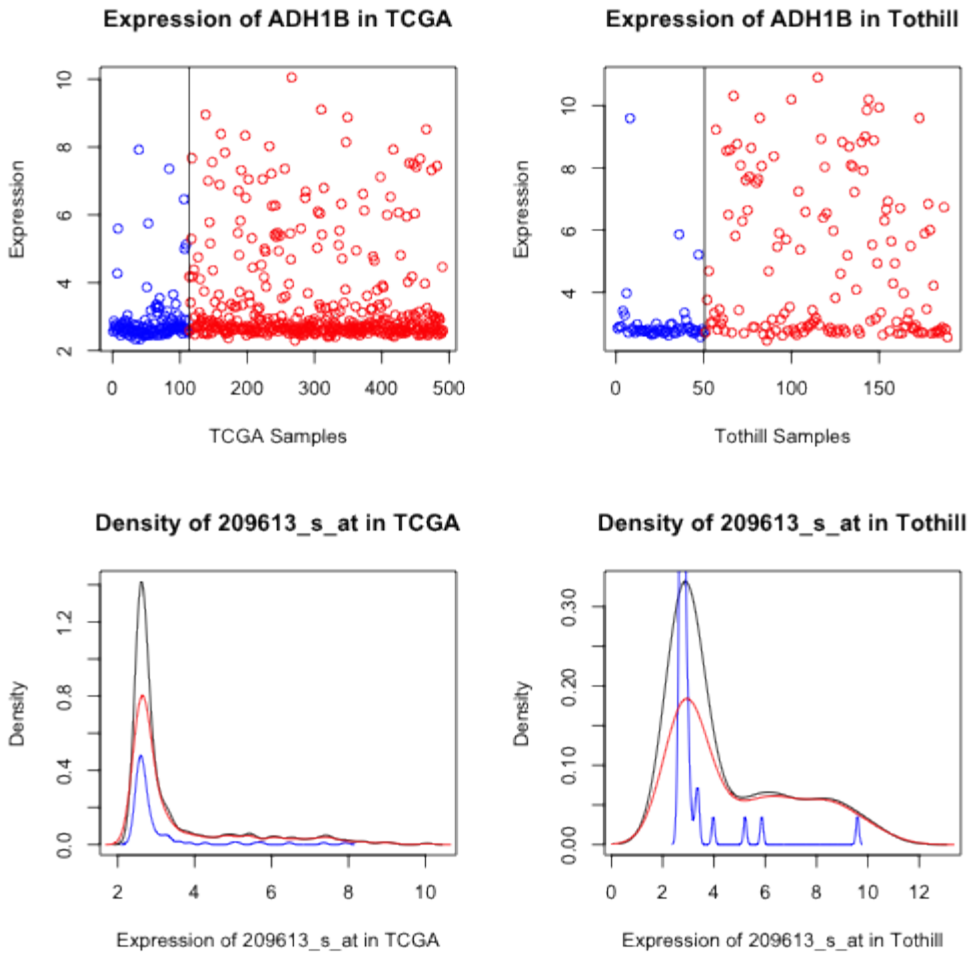


Figure 13: Dot and density plots for ADH1B in TCGA and Tothill. Cases with No RD are blue, RD are red. There is a qualitative shift in a subset of the patients.

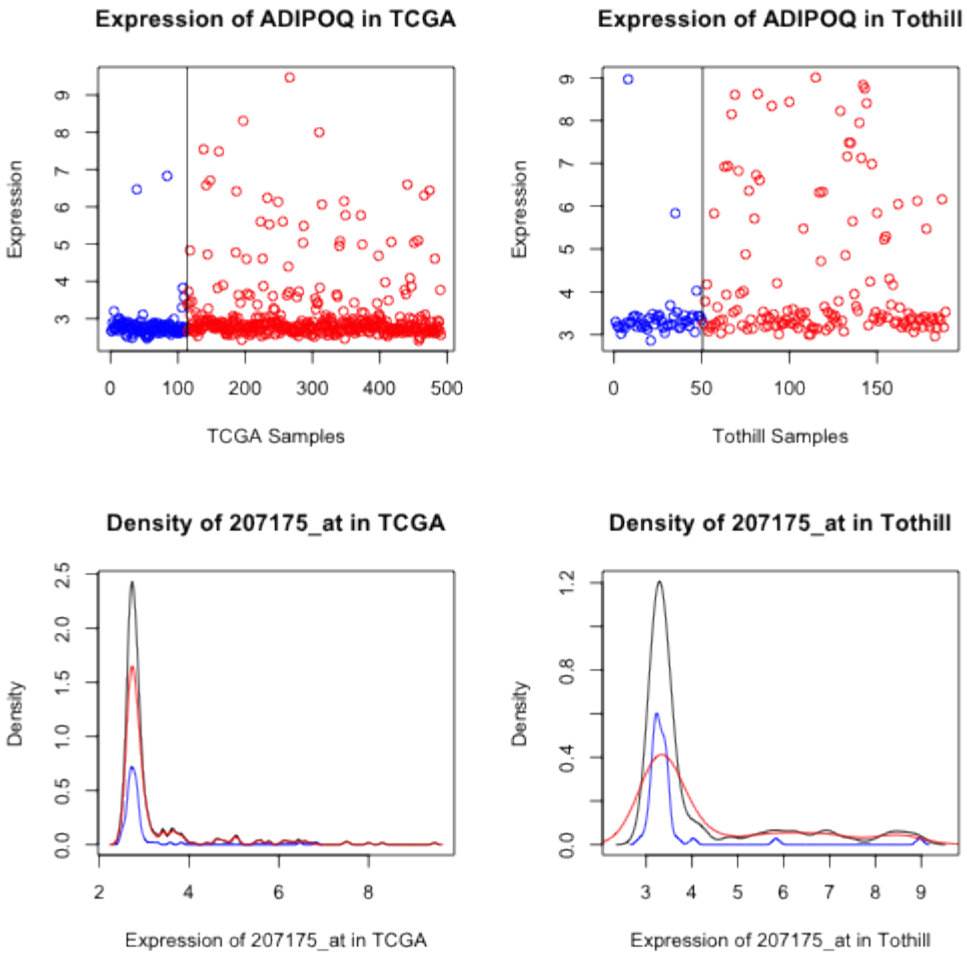


Figure 14: Dot and density plots for ADIPOQ in TCGA and Tothill. Cases with No RD are blue, RD are red. There is a qualitative shift in a subset of the patients.

7 Saving RData

Now we save the relevant information to an RData object.

```
save(tcgaCommonUsed, tothillCommonUsed, keyProbesets05pct, keyGenes05pct, keyProbesets10pct,
    keyGenes10pct, nTCGANoRD, nTCGARD, nTothillNoRD, nTothillRD, plotProbesetResults,
    rdTTestResults, file = file.path("RDataObjects", "rdFlaggedGenes.RData"))
```

8 Appendix

8.1 File Location

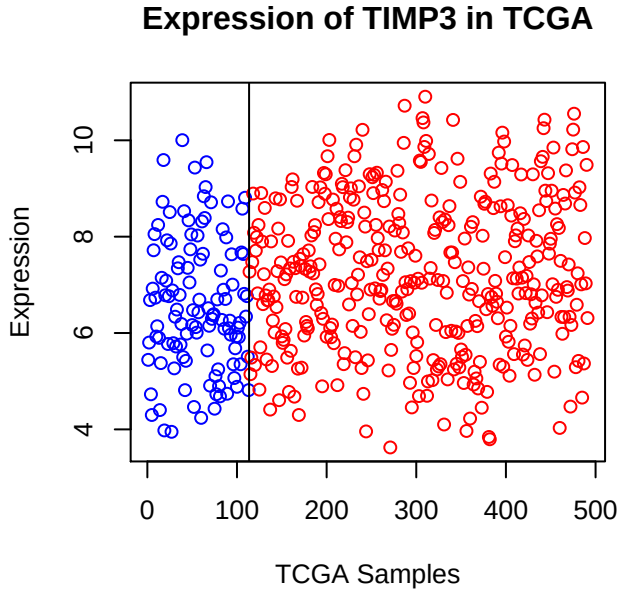
```
getwd()
```

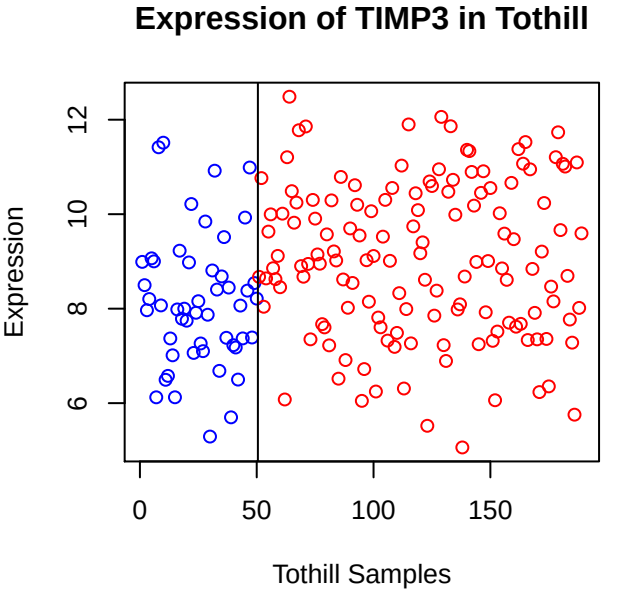
```
## [1] "/Users/slt/SLT WORKSPACE/EXEMPT/OVARIAN/Ovarian residual disease study 2012/RD
manuscript/Web page for paper/Webpage"
```

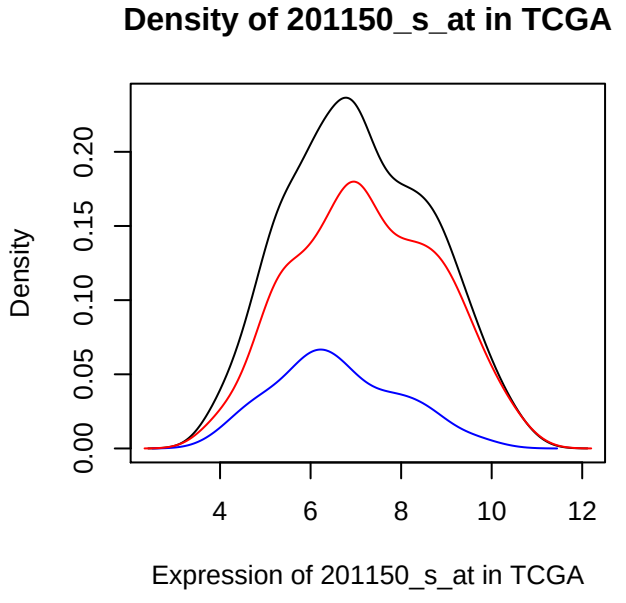
8.2 SessionInfo

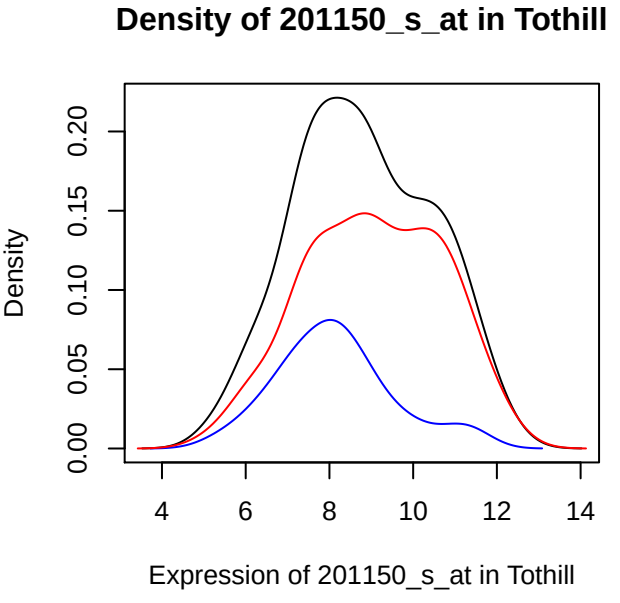
```
sessionInfo()
```

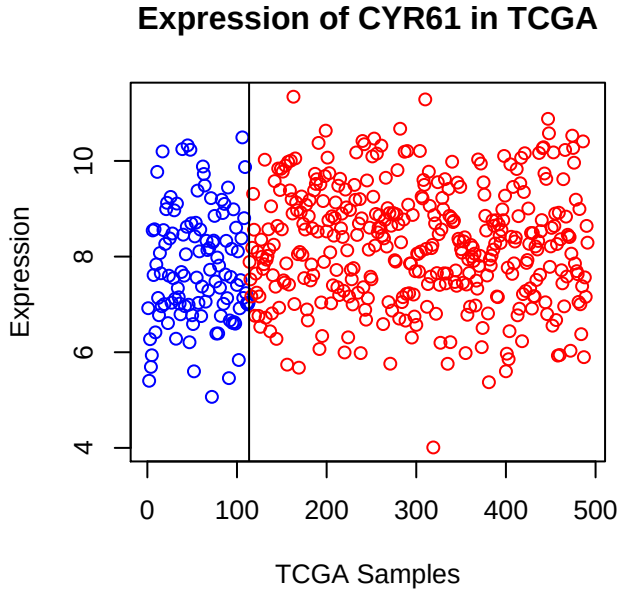
```
## R version 3.0.2 (2013-09-25)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] gplots_2.12.1       hthgu133a.db_2.9.0   org.Hs.eg.db_2.9.0
##  [4] RSQLite_0.11.4      DBI_0.2-7            AnnotationDbi_1.22.6
##  [7] affy_1.38.1         Biobase_2.20.1       BiocGenerics_0.6.0
## [10] knitr_1.5
##
## loaded via a namespace (and not attached):
##  [1] affyio_1.28.0         BiocInstaller_1.10.4  bitops_1.0-6
##  [4] caTools_1.14          evaluate_0.5.1        formatR_0.9
##  [7] gdata_2.13.2          gtools_3.1.0          IRanges_1.18.4
## [10] KernSmooth_2.23-10    preprocessCore_1.22.0 stats4_3.0.2
## [13] stringr_0.6.2         tools_3.0.2           zlibbioc_1.6.0
```

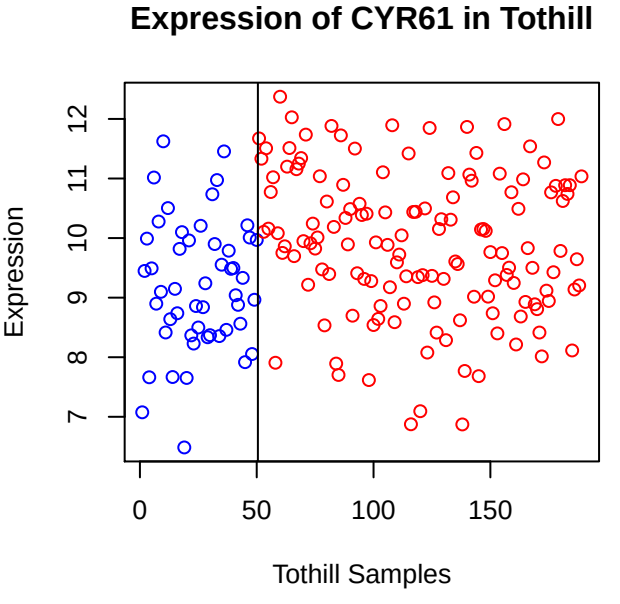

Expression of TIMP3 in TCGA
Expression
4
6
8
10
0
100
200
300
400
500
TCGA Samples

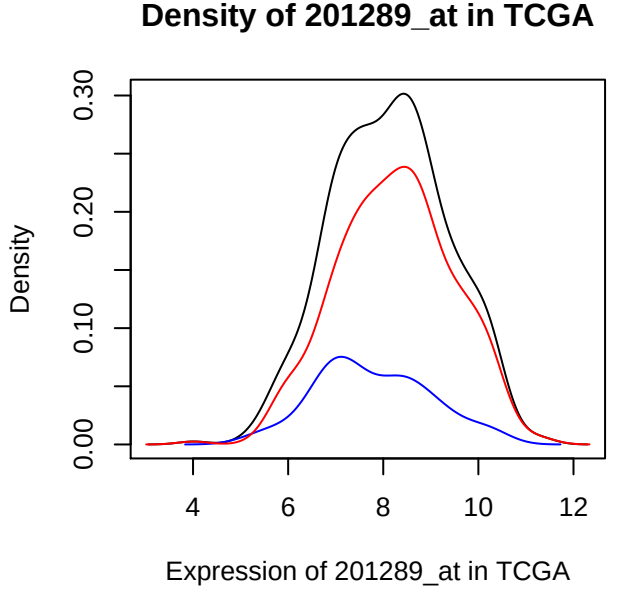

Expression of TIMP3 in Tothill
Expression
6
8
10
12
0
50
100
150
Tothill Samples

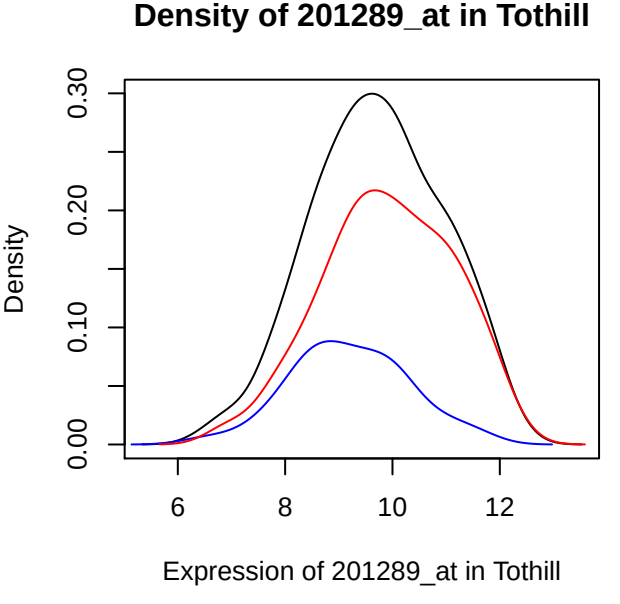

Density of 201150_s_at in TCGA
Density
0.00
0.05
0.10
0.15
0.20
4
6
8
10
12
Expression of 201150_s_at in TCGA

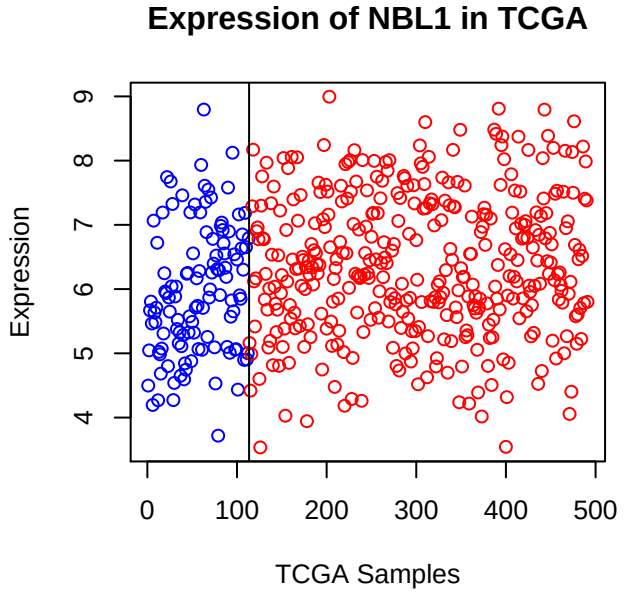

Density of 201150_s_at in Tothill
Density
0.00
0.05
0.10
0.15
0.20
4
6
8
10
12
14
Expression of 201150_s_at in Tothill

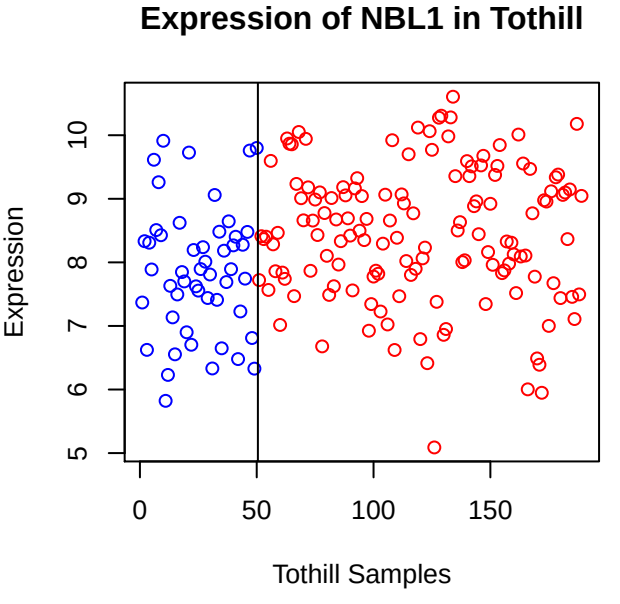

Expression of CYR61 in TCGA
Expression
4
6
8
10
0
100
200
300
400
500
TCGA Samples

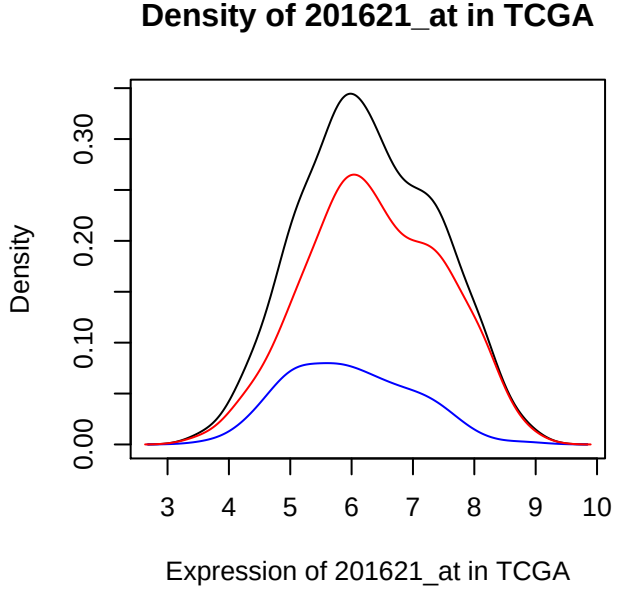

Expression of CYR61 in Tothill
Expression
7
8
9
10
11
12
0
50
100
150
Tothill Samples

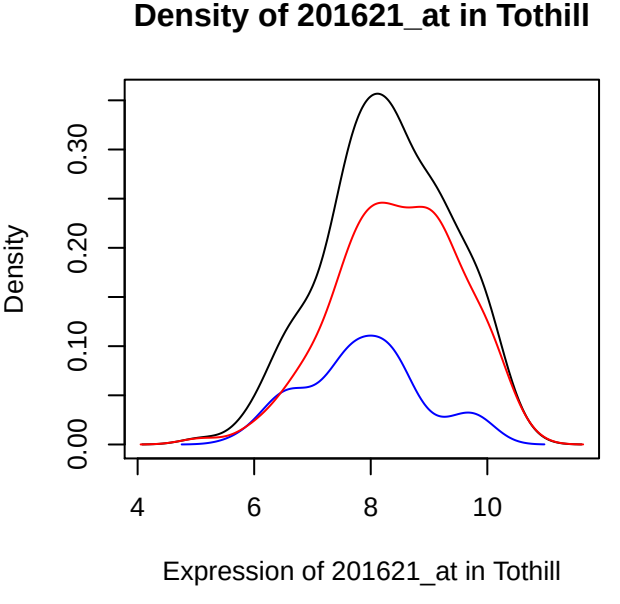

Density of 201289_at in TCGA
Density
0.00
0.10
0.20
0.30
4
6
8
10
12
Expression of 201289_at in TCGA

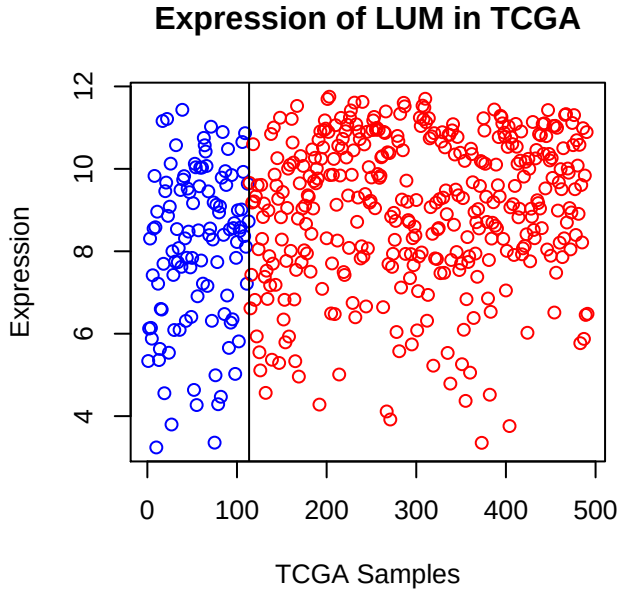

Density of 201289_at in Tothill
Density
0.00
0.10
0.20
0.30
6
8
10
12
Expression of 201289_at in Tothill

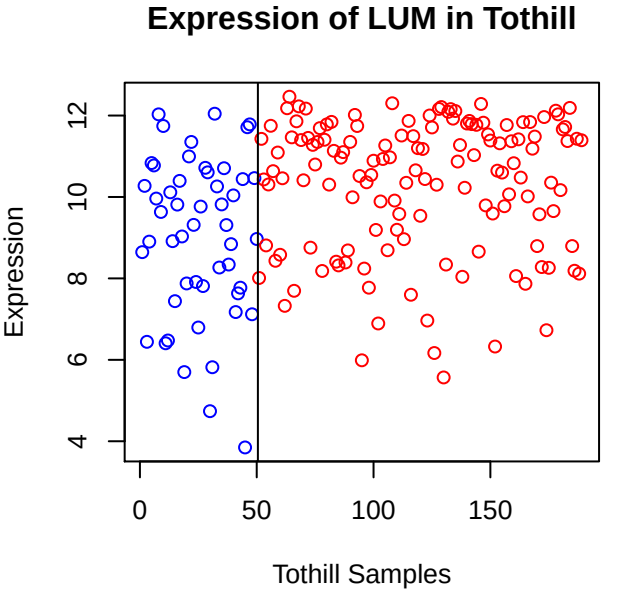

Expression of NBL1 in TCGA
Expression
TCGA Samples
0
100
200
300
400
500
4
5
6
7
8
9

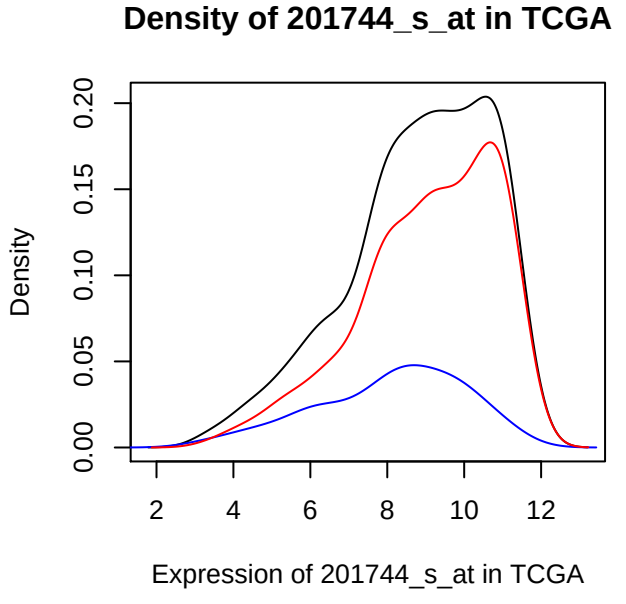

Expression of NBL1 in Tothill
Expression
Tothill Samples
0
50
100
150
5
6
7
8
9
10

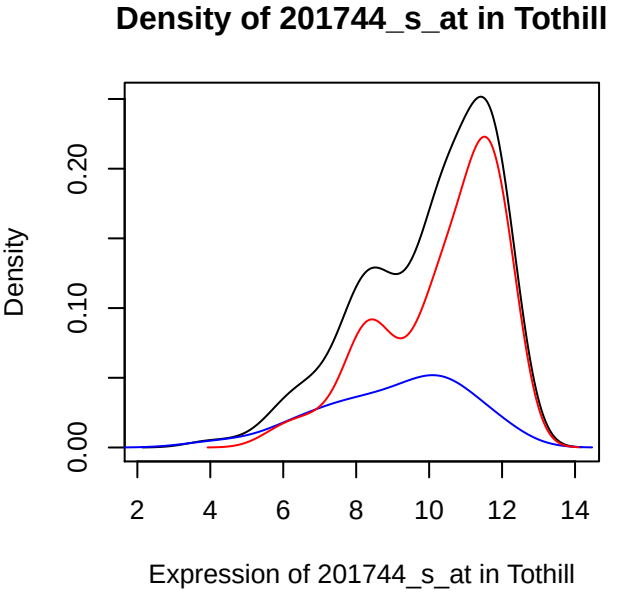

Density of 201621_at in TCGA
Density
Expression of 201621_at in TCGA
0.00
0.10
0.20
0.30
3
4
5
6
7
8
9
10

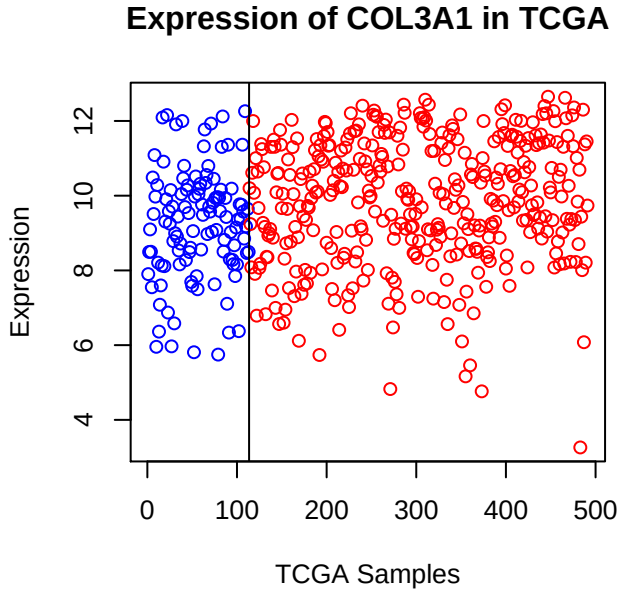

Density of 201621_at in Tothill
Density
Expression of 201621_at in Tothill
0.00
0.10
0.20
0.30
4
6
8
10

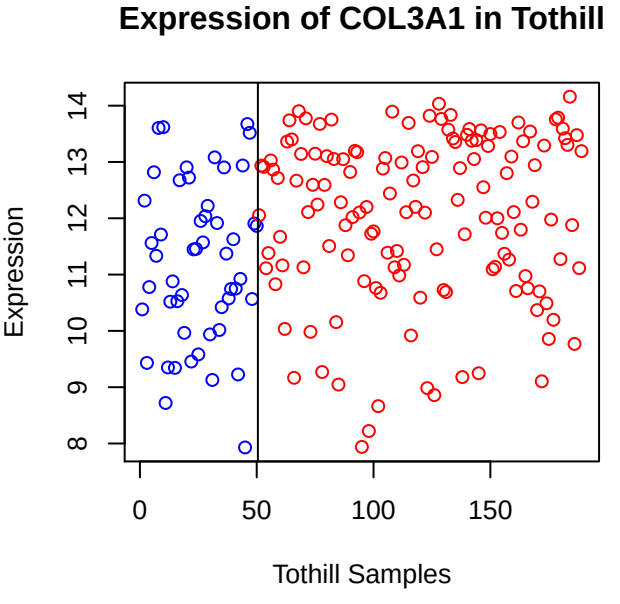

Expression of LUM in TCGA
Expression
4
6
8
10
12
0
100
200
300
400
500
TCGA Samples

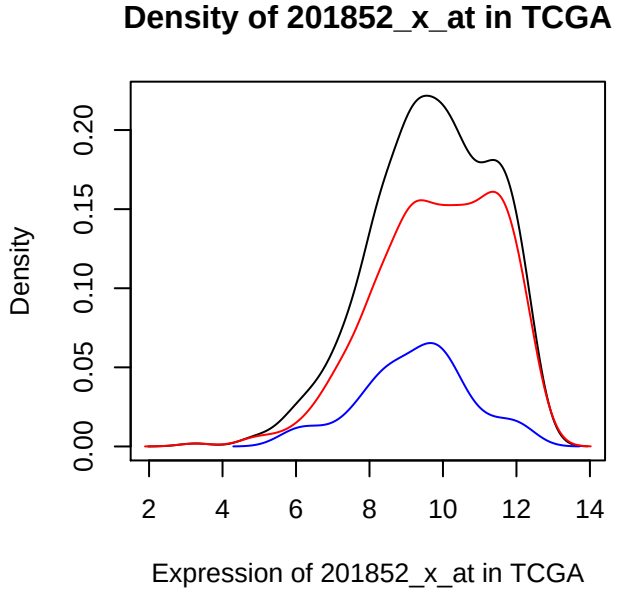

Expression of LUM in Tothill
Expression
4
6
8
10
12
0
50
100
150
Tothill Samples

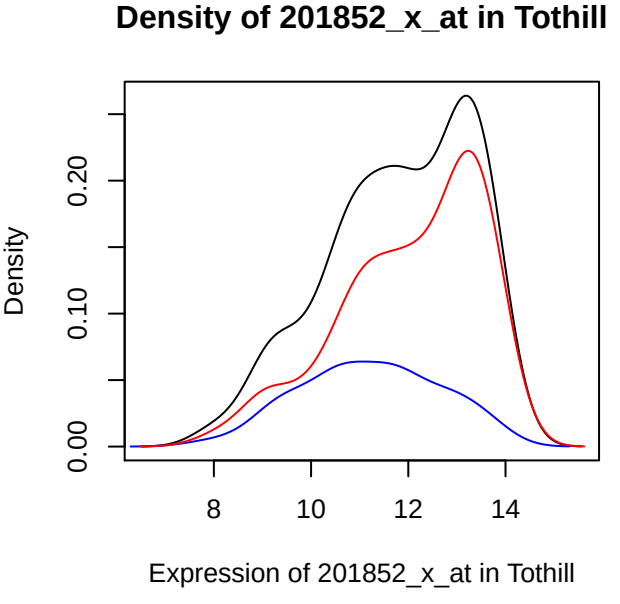

Density of 201744_s_at in TCGA
Density
0.00
0.05
0.10
0.15
0.20
2
4
6
8
10
12
Expression of 201744_s_at in TCGA

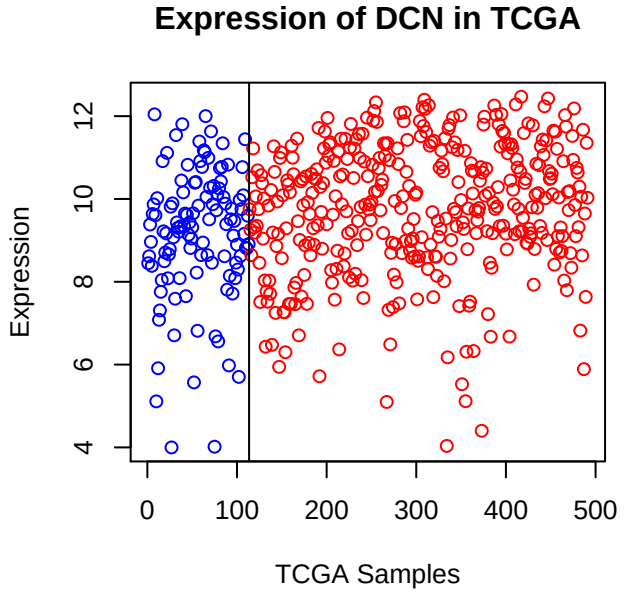

Density of 201744_s_at in Tothill
Density
0.00
0.10
0.20
2
4
6
8
10
12
14
Expression of 201744_s_at in Tothill

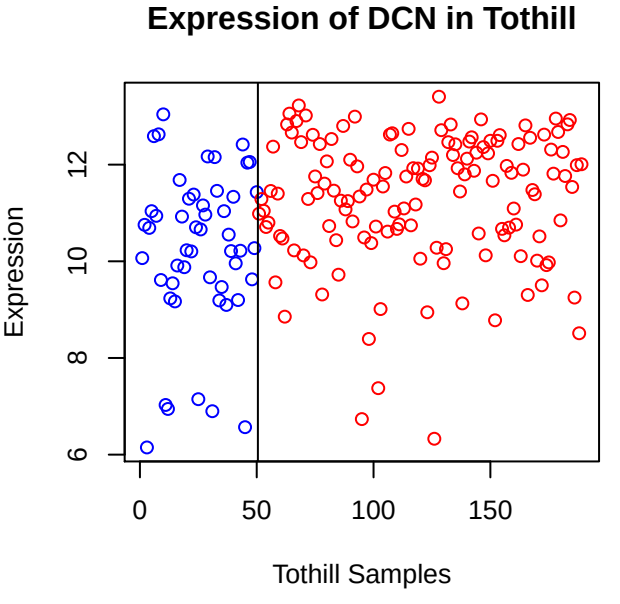

Expression of COL3A1 in TCGA
Expression
4
6
8
10
12
0
100
200
300
400
500
TCGA Samples

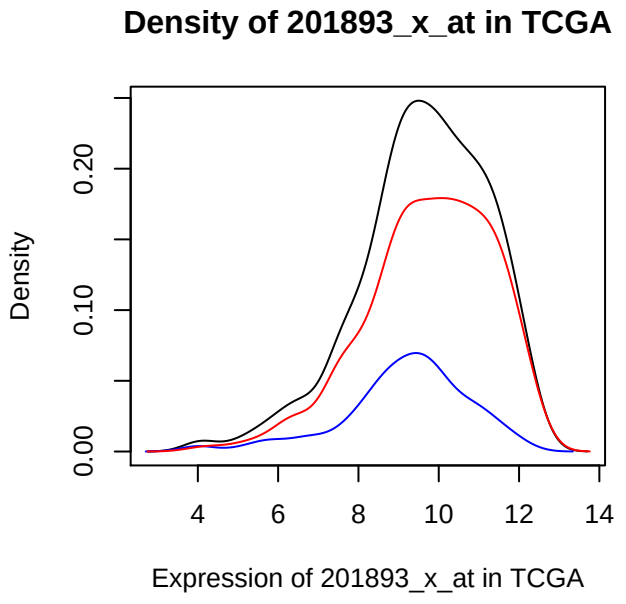

Expression of COL3A1 in Tothill
Expression
8
9
10
11
12
13
14
0
50
100
150
Tothill Samples

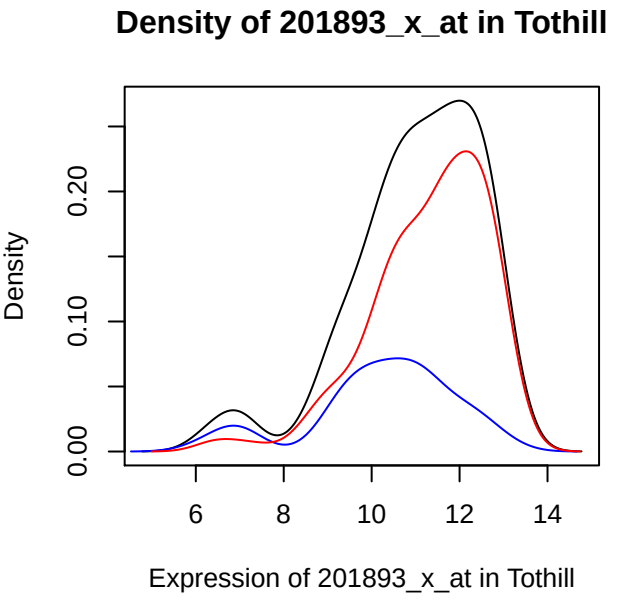

Density of 201852_x_at in TCGA
Density
0.00
0.05
0.10
0.15
0.20
2
4
6
8
10
12
14
Expression of 201852_x_at in TCGA

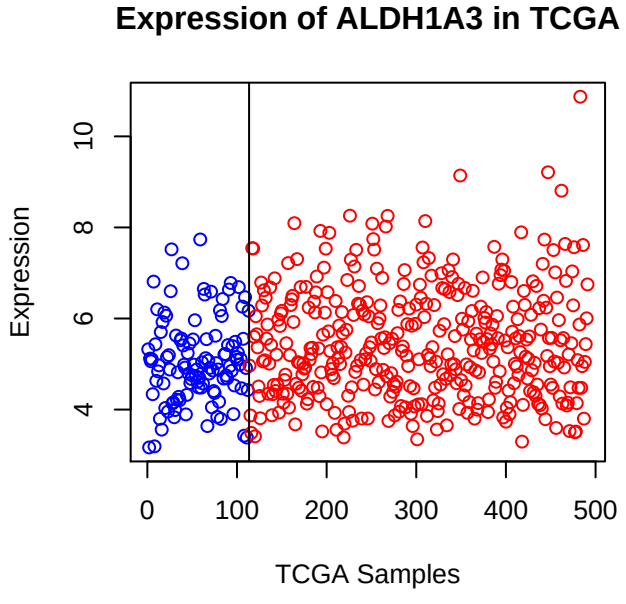

Density of 201852_x_at in Tothill
Density
0.00
0.10
0.20
8
10
12
14
Expression of 201852_x_at in Tothill

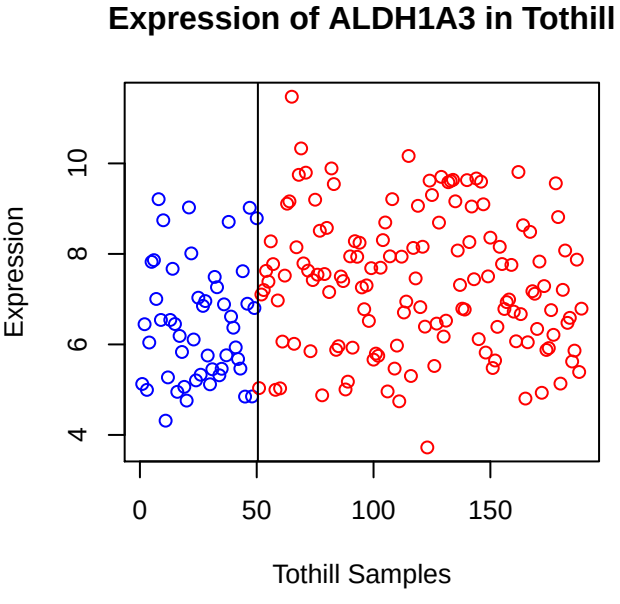

Expression of DCN in TCGA
Expression
4
6
8
10
12
0
100
200
300
400
500
TCGA Samples

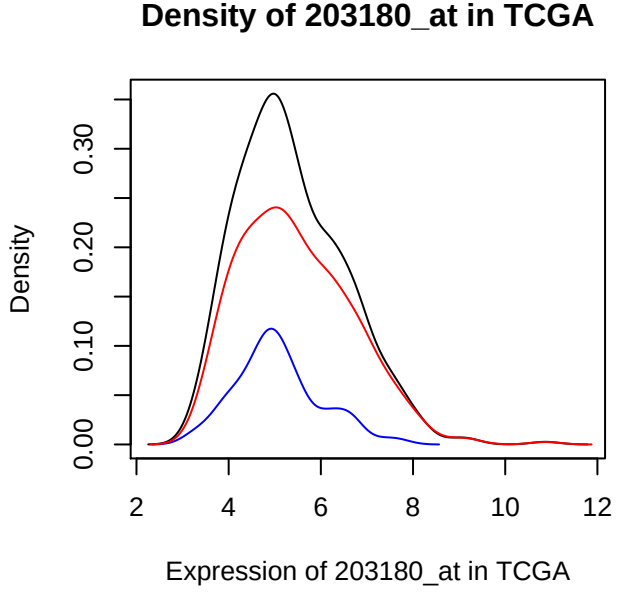

Expression of DCN in Tothill
Expression
6
8
10
12
0
50
100
150
Tothill Samples

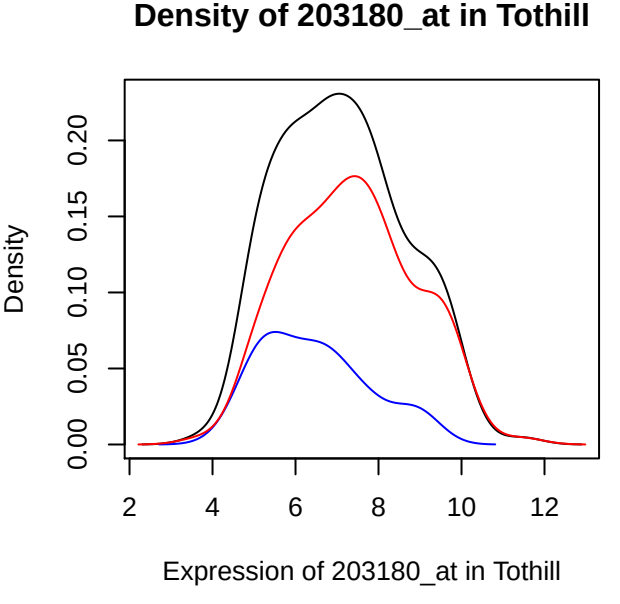

Density of 201893_x_at in TCGA
Density
0.00
0.10
0.20
4
6
8
10
12
14
Expression of 201893_x_at in TCGA

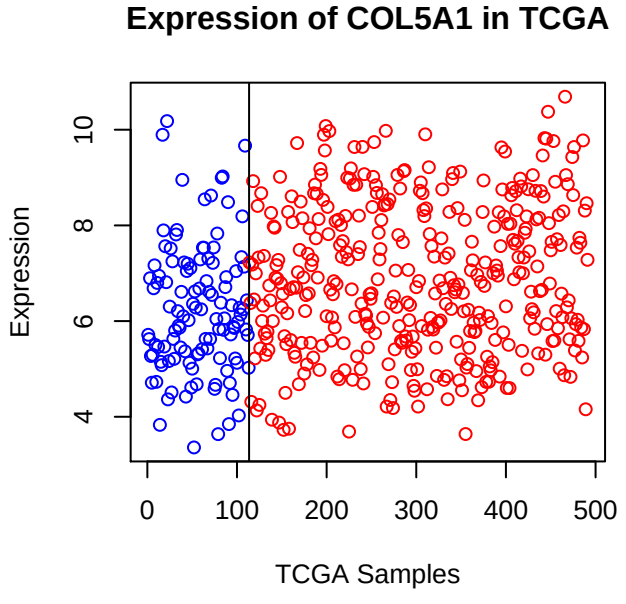

Density of 201893_x_at in Tothill
Density
0.00
0.10
0.20
6
8
10
12
14
Expression of 201893_x_at in Tothill

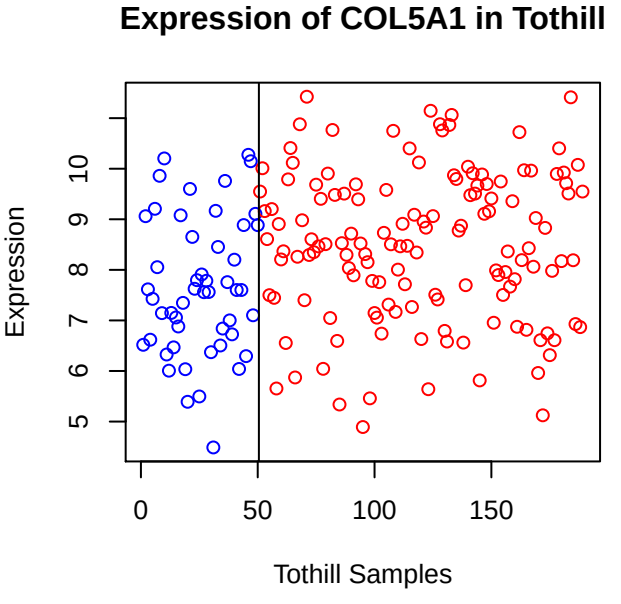

Expression of ALDH1A3 in TCGA
Expression
4
6
8
10
0
100
200
300
400
500
TCGA Samples

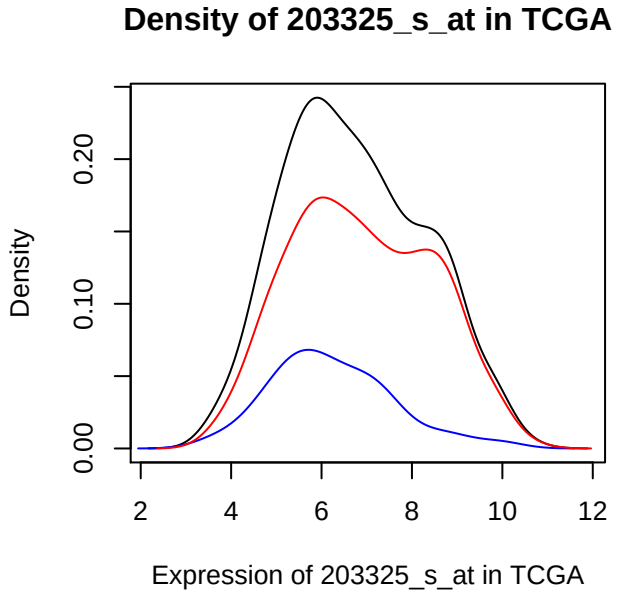

Expression of ALDH1A3 in Tothill
Expression
4
6
8
10
0
50
100
150
Tothill Samples

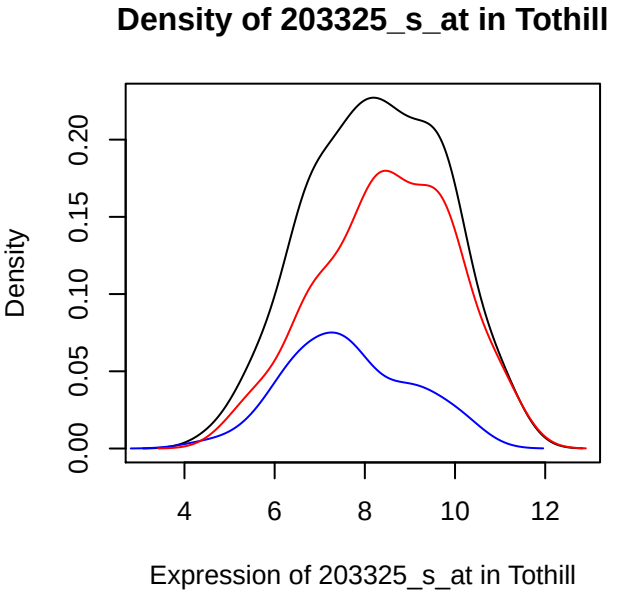

Density of 203180_at in TCGA
Density
0.00
0.10
0.20
0.30
2
4
6
8
10
12
Expression of 203180_at in TCGA

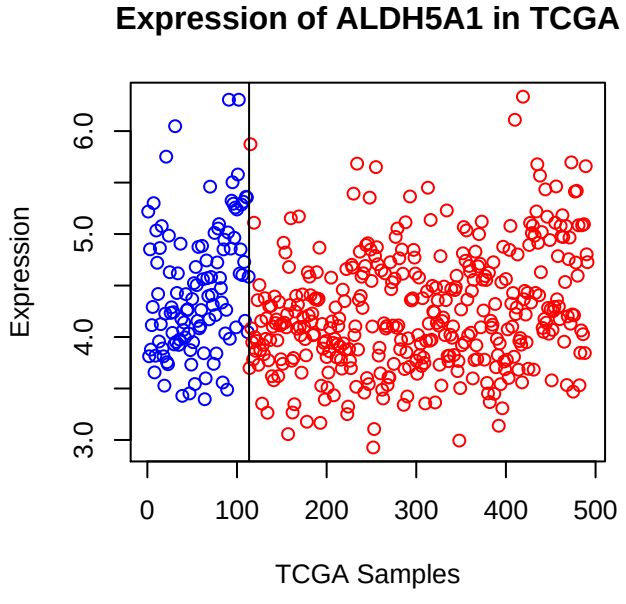

Density of 203180_at in Tothill
Density
0.00
0.05
0.10
0.15
0.20
2
4
6
8
10
12
Expression of 203180_at in Tothill

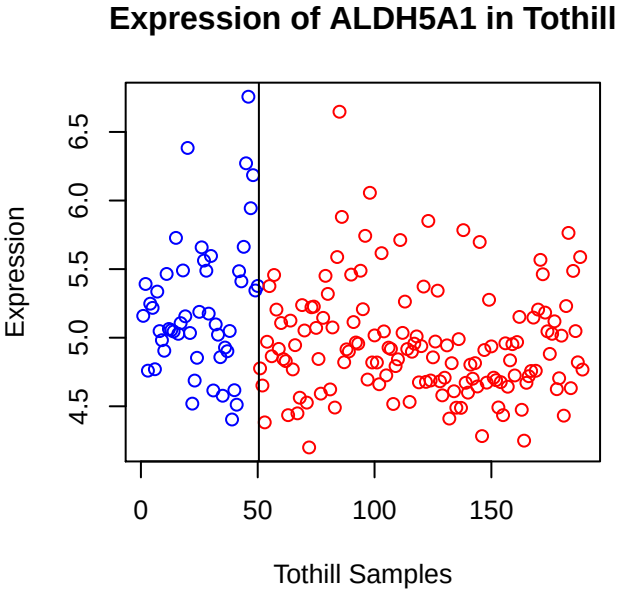

Expression of COL5A1 in TCGA
Expression
4
6
8
10
0
100
200
300
400
500
TCGA Samples

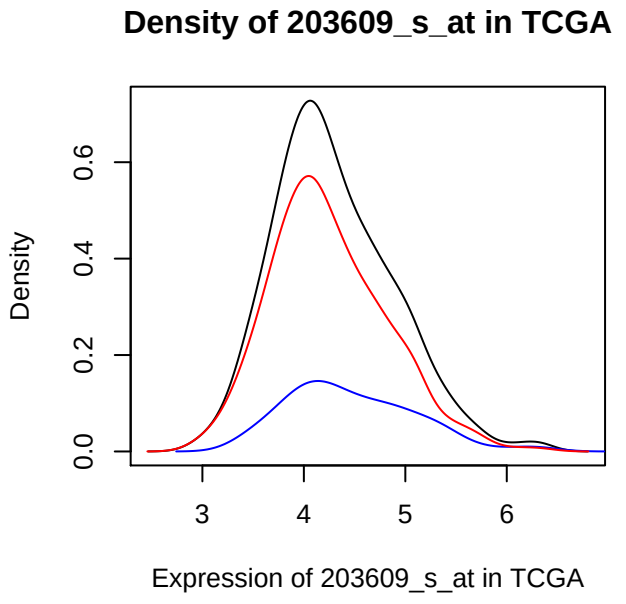

Expression of COL5A1 in Tothill
Expression
5
6
7
8
9
10
0
50
100
150
Tothill Samples

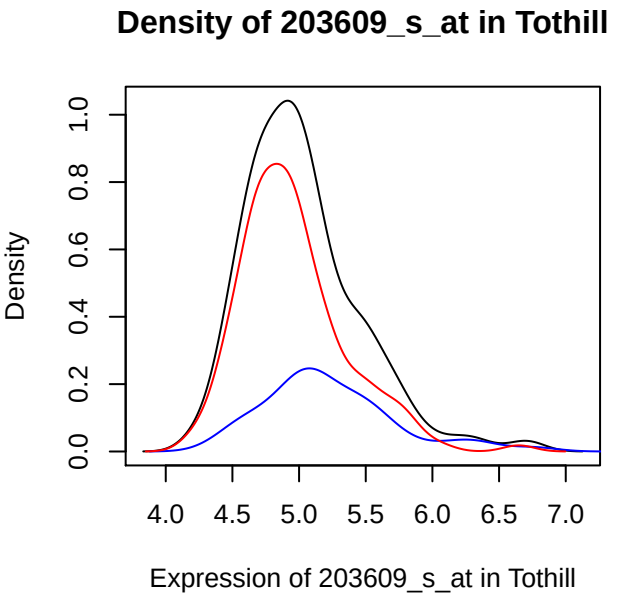

Density of 203325_s_at in TCGA
Density
0.00
0.10
0.20
2
4
6
8
10
12
Expression of 203325_s_at in TCGA

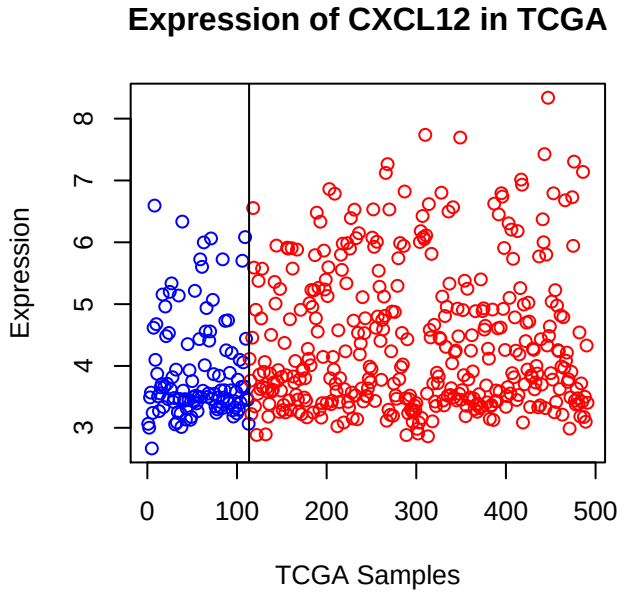

Density of 203325_s_at in Tothill
Density
0.00
0.05
0.10
0.15
0.20
4
6
8
10
12
Expression of 203325_s_at in Tothill

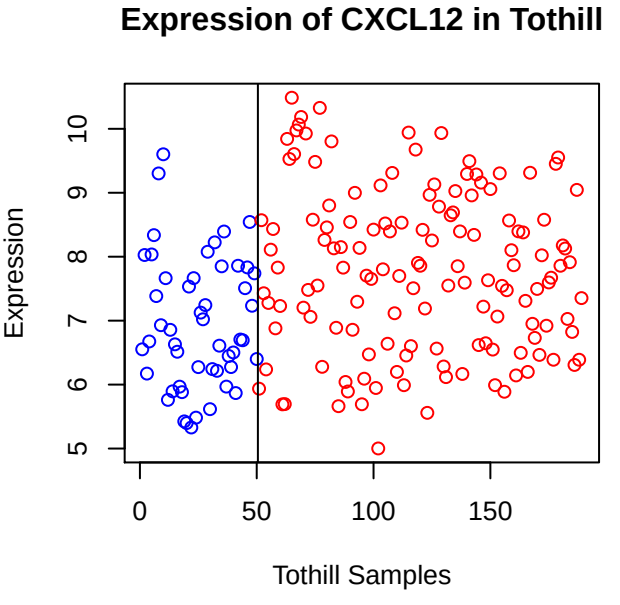

Expression of ALDH5A1 in TCGA
Expression
3.0
4.0
5.0
6.0
0
100
200
300
400
500
TCGA Samples

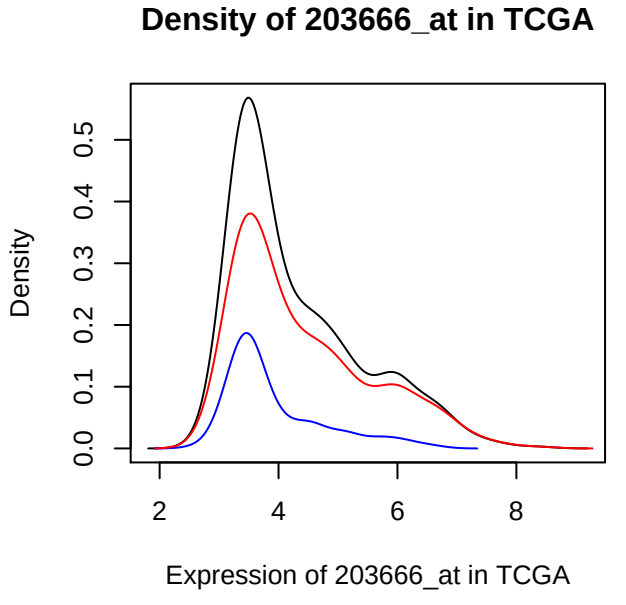

Expression of ALDH5A1 in Tothill
Expression
4.5
5.0
5.5
6.0
6.5
0
50
100
150
Tothill Samples

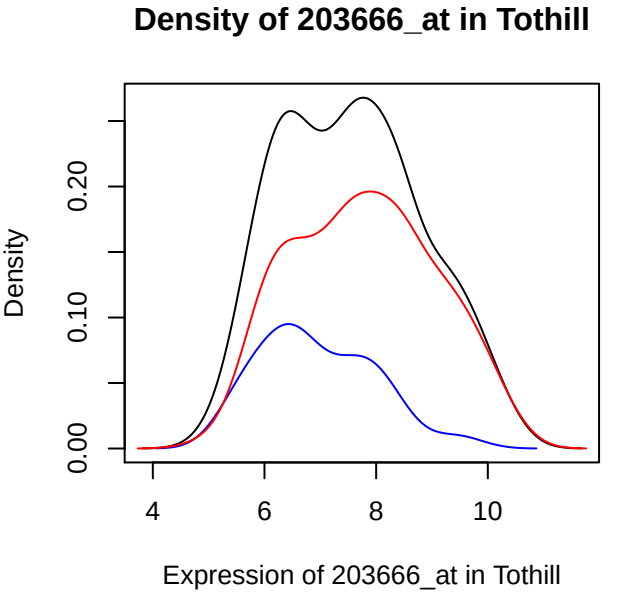

Density of 203609_s_at in TCGA
Density
0.0
0.2
0.4
0.6
3
4
5
6
Expression of 203609_s_at in TCGA

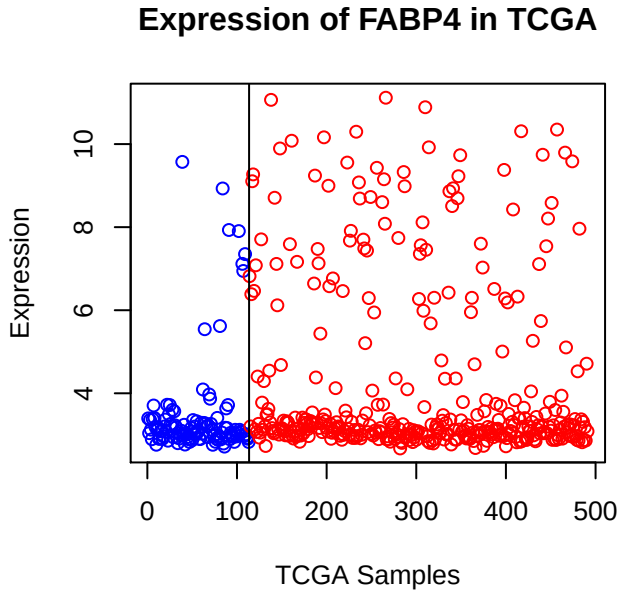

Density of 203609_s_at in Tothill
Density
0.0
0.2
0.4
0.6
0.8
1.0
4.0
4.5
5.0
5.5
6.0
6.5
7.0
Expression of 203609_s_at in Tothill

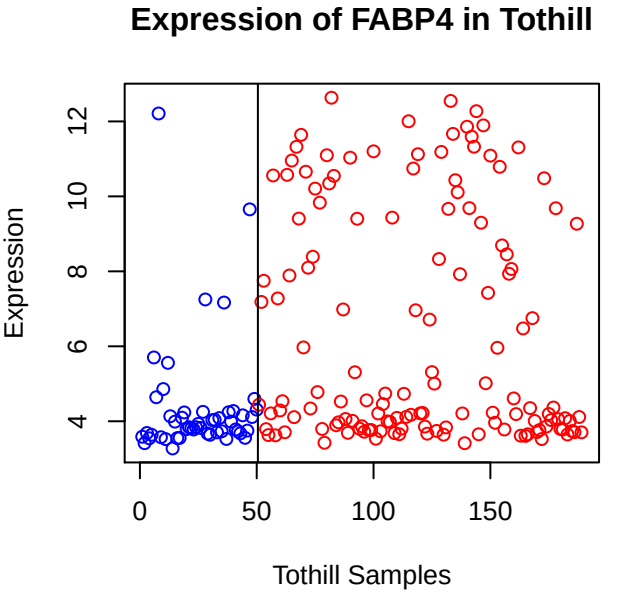

Expression of CXCL12 in TCGA
Expression
3
4
5
6
7
8
0
100
200
300
400
500
TCGA Samples

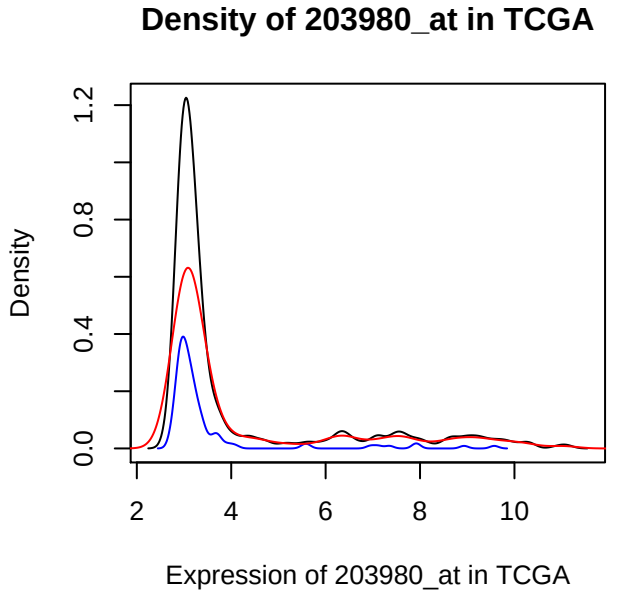

Expression of CXCL12 in Tothill
Expression
5
6
7
8
9
10
0
50
100
150
Tothill Samples

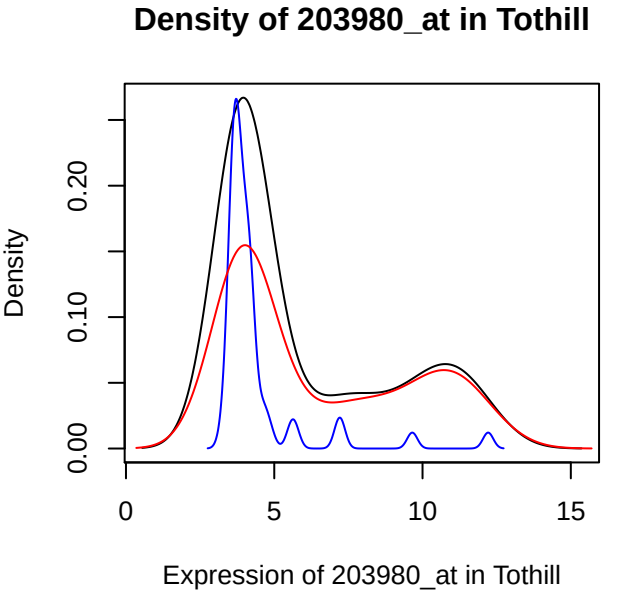

Density of 203666_at in TCGA
Density
0.0
0.1
0.2
0.3
0.4
0.5
2
4
6
8
Expression of 203666_at in TCGA

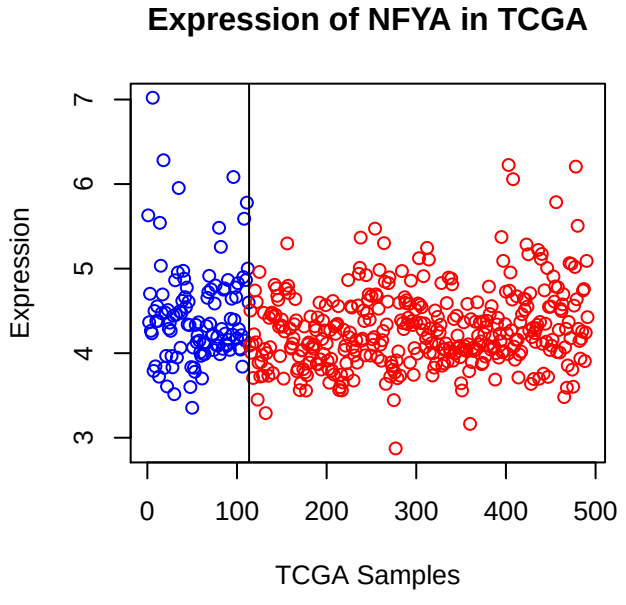

Density of 203666_at in Tothill
Density
0.00
0.10
0.20
4
6
8
10
Expression of 203666_at in Tothill

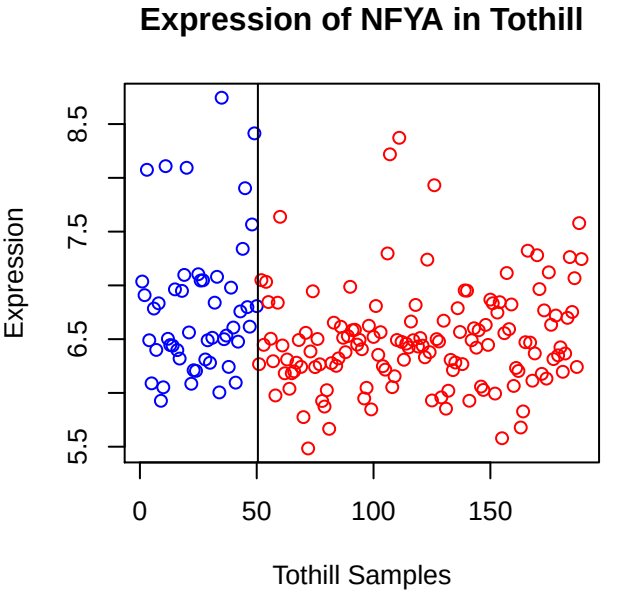

Expression of FABP4 in TCGA
Expression
4
6
8
10
0
100
200
300
400
500
TCGA Samples

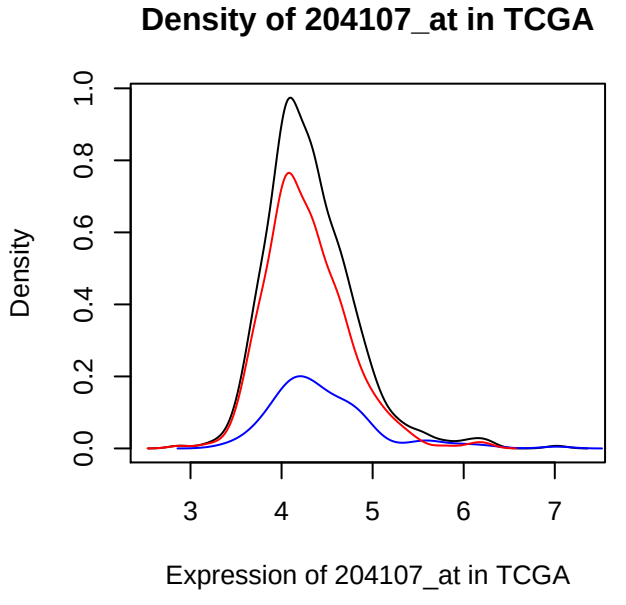

Expression of FABP4 in Tothill
Expression
4
6
8
10
12
0
50
100
150
Tothill Samples

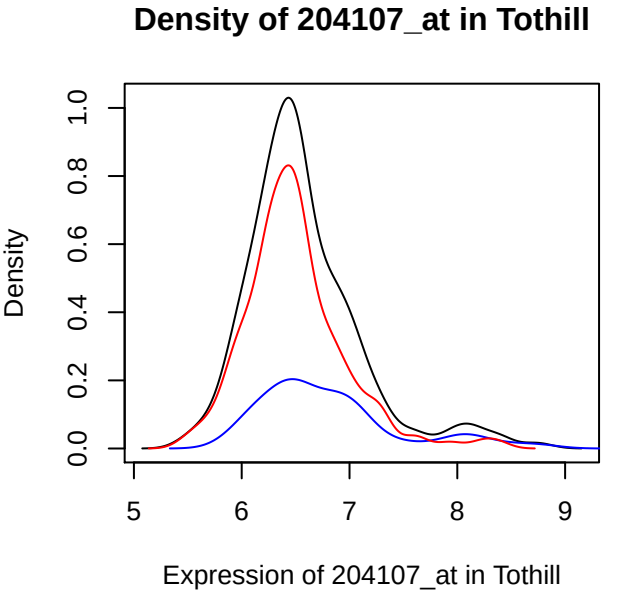

Density of 203980_at in TCGA
Density
0.0
0.4
0.8
1.2
2
4
6
8
10
Expression of 203980_at in TCGA

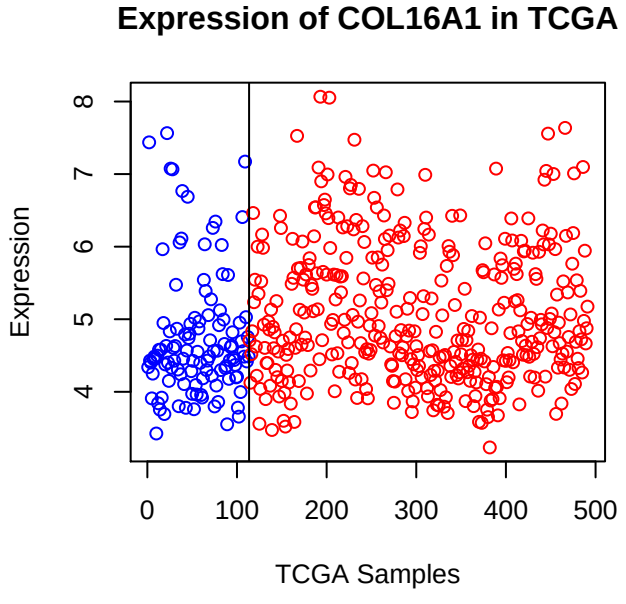

Density of 203980_at in Tothill
Density
0.00
0.10
0.20
0
5
10
15
Expression of 203980_at in Tothill

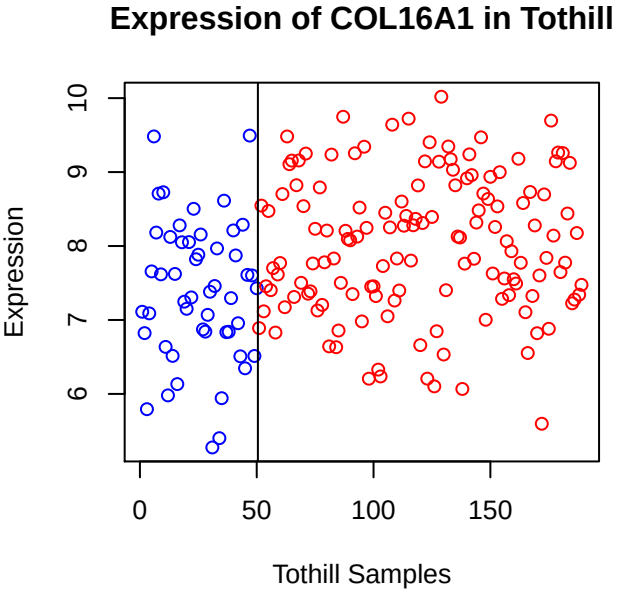

Expression of NFYA in TCGA
Expression
3
4
5
6
7
0
100
200
300
400
500
TCGA Samples

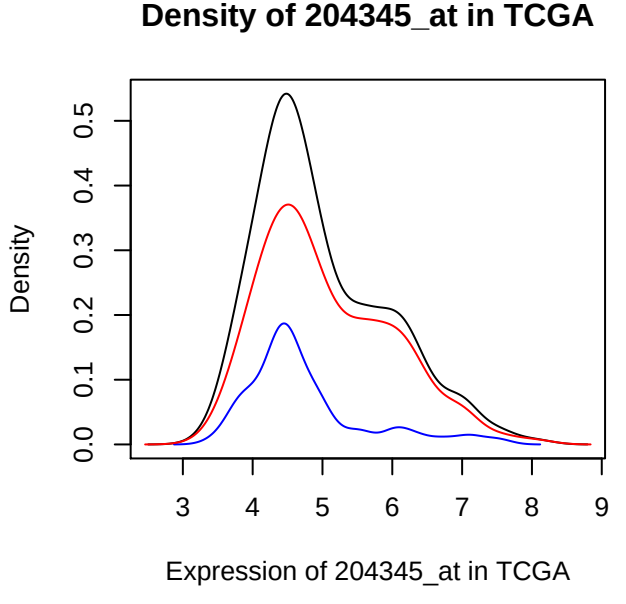

Expression of NFYA in Tothill
Expression
5.5
6.5
7.5
8.5
0
50
100
150
Tothill Samples

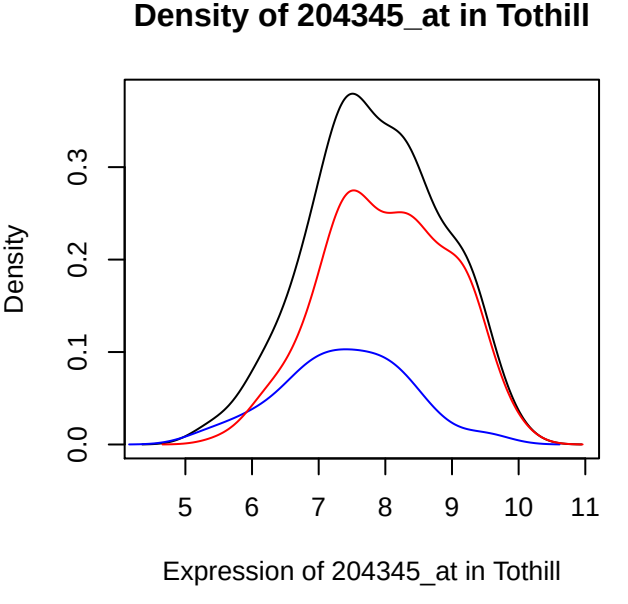

Density of 204107_at in TCGA
Density
0.0
0.2
0.4
0.6
0.8
1.0
3
4
5
6
7
Expression of 204107_at in TCGA

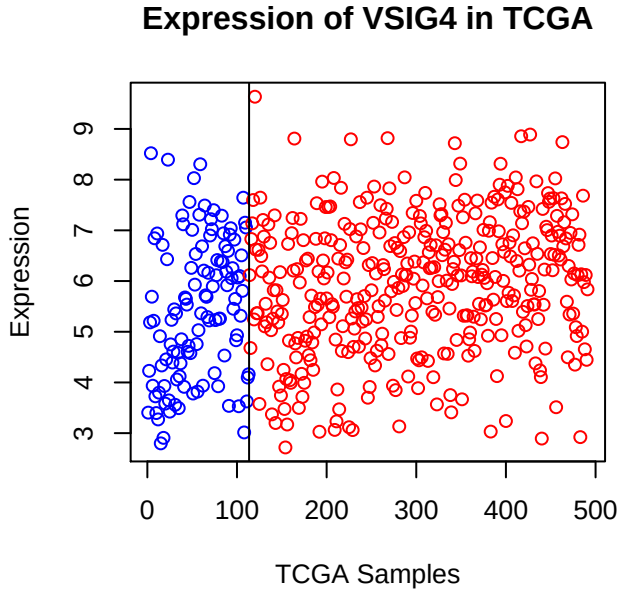

Density of 204107_at in Tothill
Density
0.0
0.2
0.4
0.6
0.8
1.0
5
6
7
8
9
Expression of 204107_at in Tothill

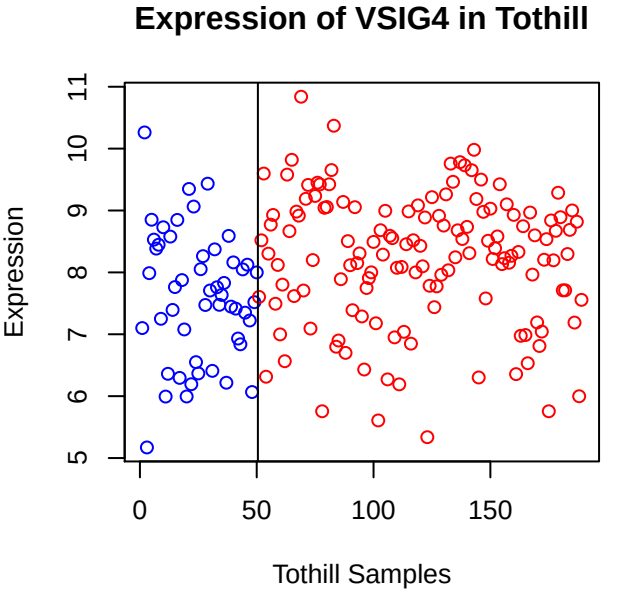

Expression of COL16A1 in TCGA
Expression
4
5
6
7
8
0
100
200
300
400
500
TCGA Samples

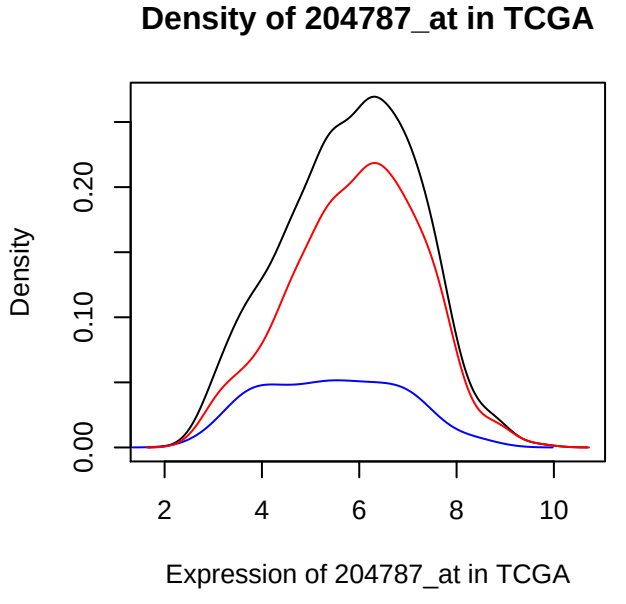

Expression of COL16A1 in Tothill
Expression
6
7
8
9
10
0
50
100
150
Tothill Samples

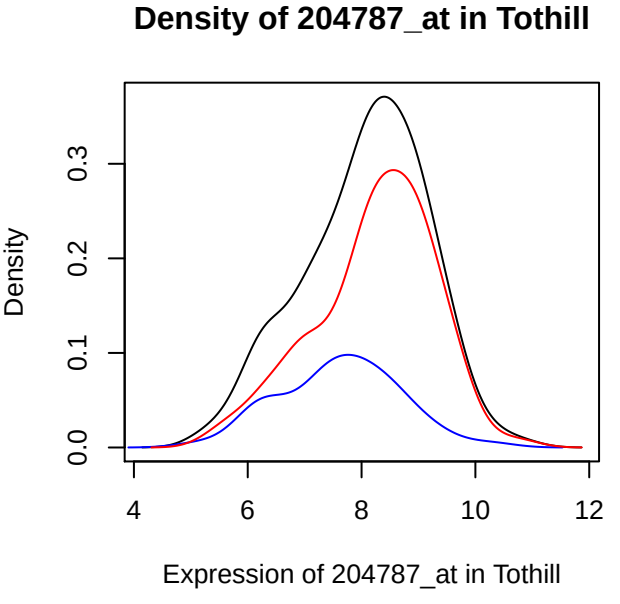

Density of 204345_at in TCGA
Density
0.0
0.1
0.2
0.3
0.4
0.5
3
4
5
6
7
8
9
Expression of 204345_at in TCGA

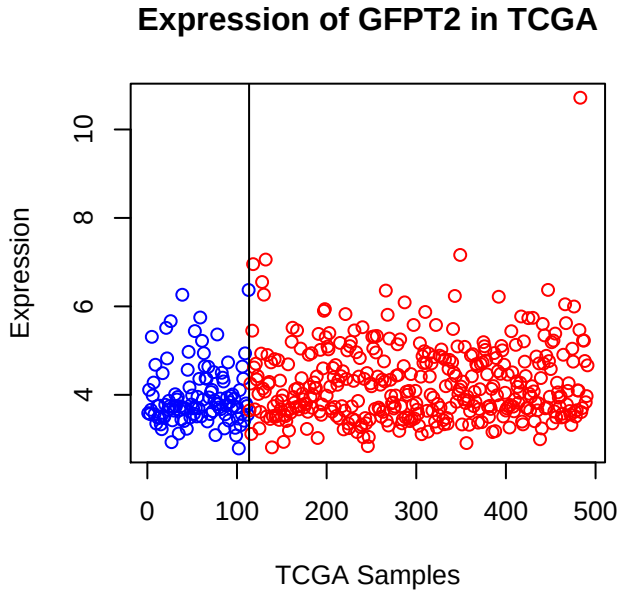

Density of 204345_at in Tothill
Density
0.0
0.1
0.2
0.3
5
6
7
8
9
10
11
Expression of 204345_at in Tothill

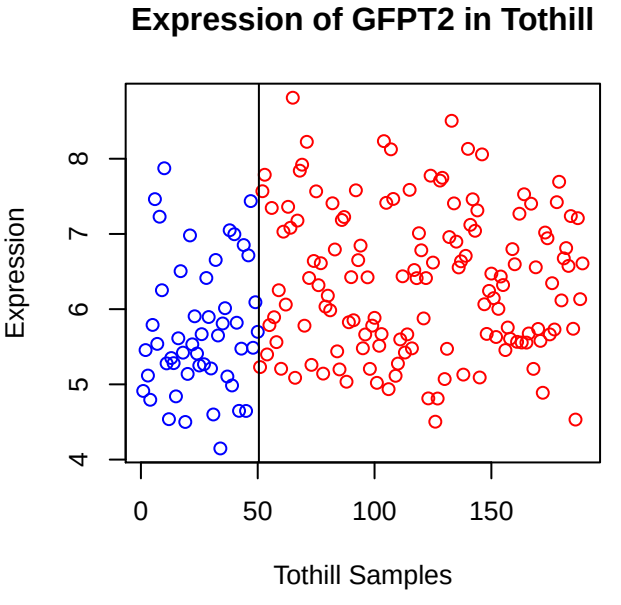

Expression of VSIG4 in TCGA
Expression
3
4
5
6
7
8
9
0
100
200
300
400
500
TCGA Samples

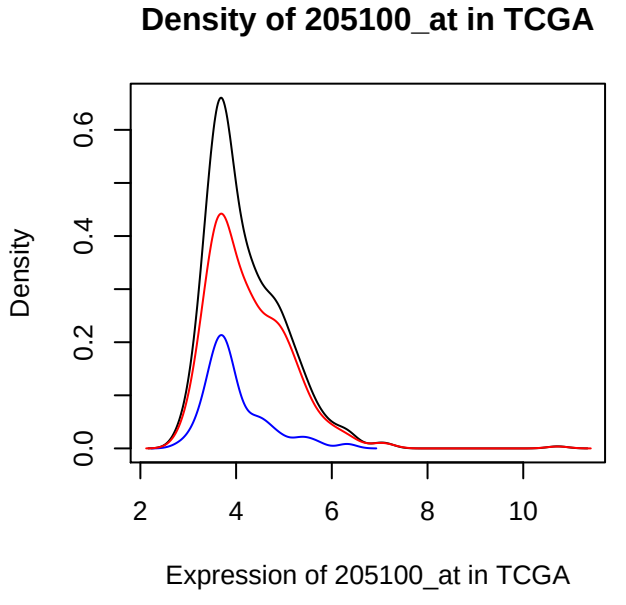

Expression of VSIG4 in Tothill
Expression
5
6
7
8
9
10
11
0
50
100
150
Tothill Samples

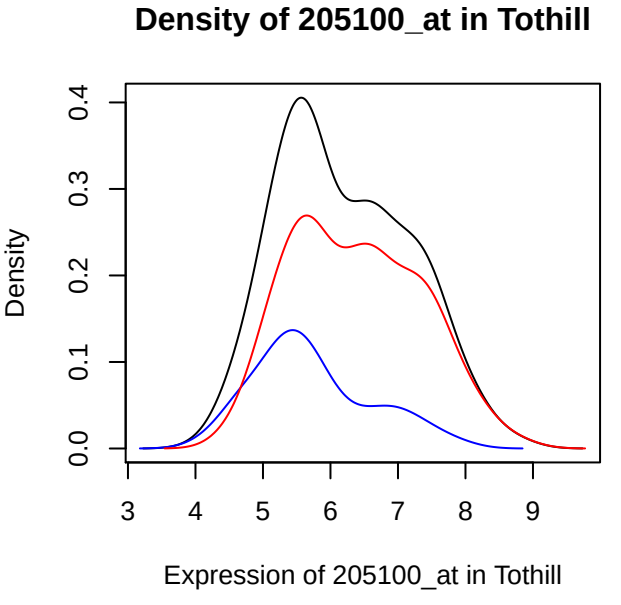

Density of 204787_at in TCGA
Density
0.00
0.10
0.20
2
4
6
8
10
Expression of 204787_at in TCGA

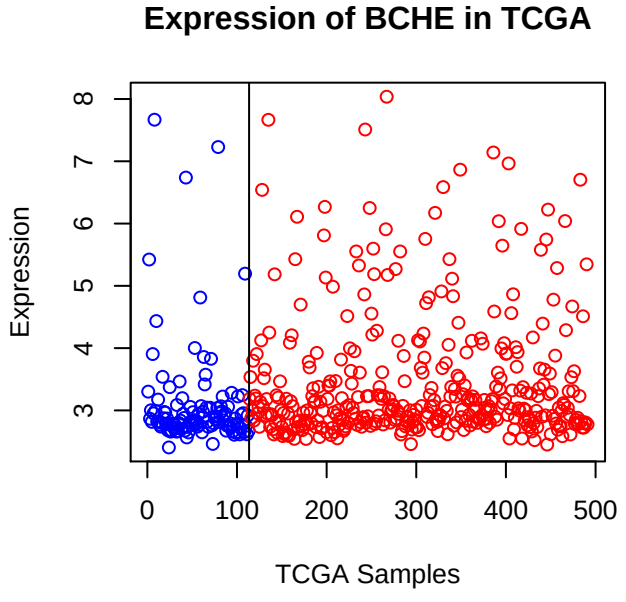

Density of 204787_at in Tothill
Density
0.0
0.1
0.2
0.3
4
6
8
10
12
Expression of 204787_at in Tothill

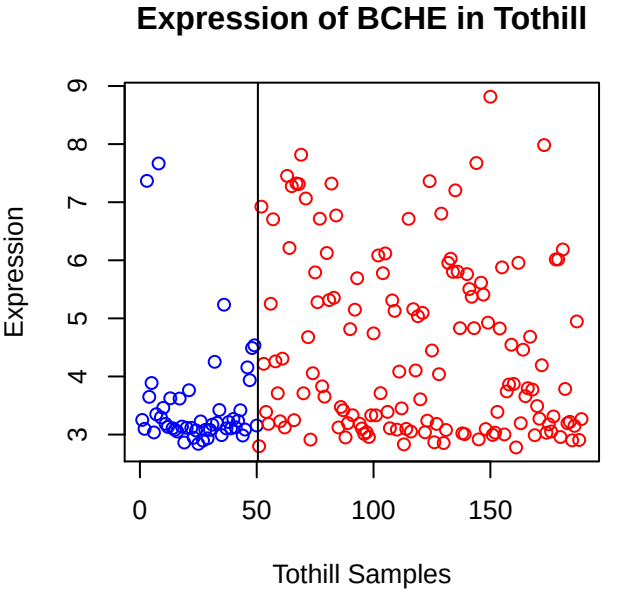

Expression of GFPT2 in TCGA
Expression
4
6
8
10
0
100
200
300
400
500
TCGA Samples

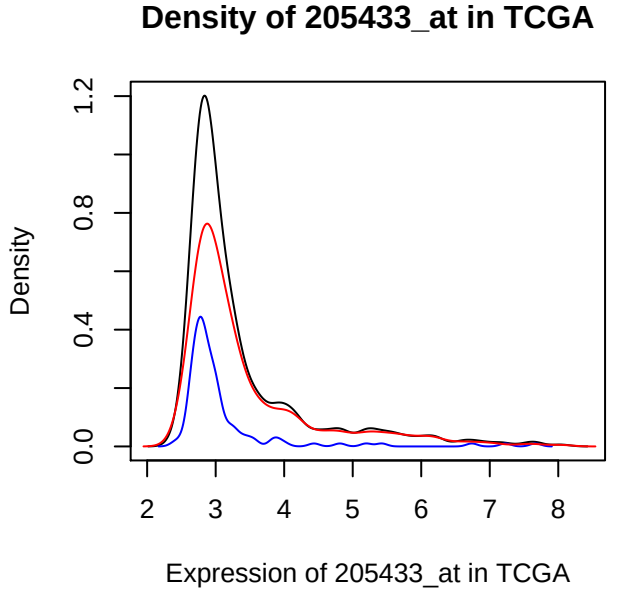

Expression of GFPT2 in Tothill
Expression
4
5
6
7
8
0
50
100
150
Tothill Samples

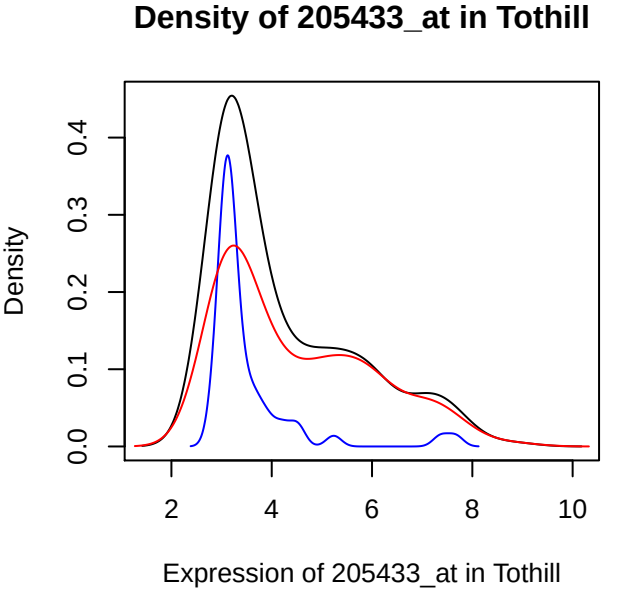

Density of 205100_at in TCGA
Density
0.0
0.2
0.4
0.6
2
4
6
8
10
Expression of 205100_at in TCGA

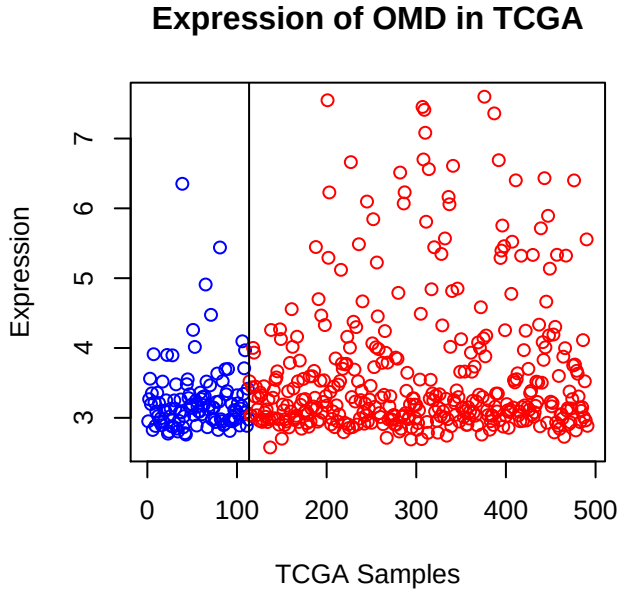

Density of 205100_at in Tothill
Density
0.0
0.1
0.2
0.3
0.4
3
4
5
6
7
8
9
Expression of 205100_at in Tothill

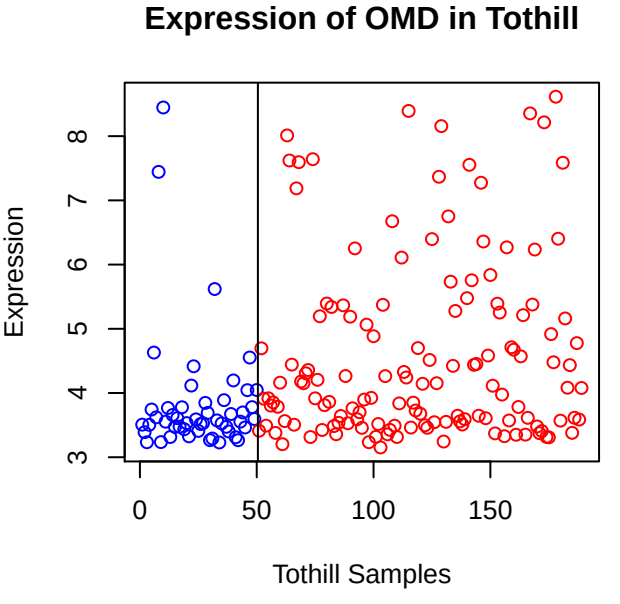

Expression of BCHE in TCGA
Expression
3
4
5
6
7
8
0
100
200
300
400
500
TCGA Samples

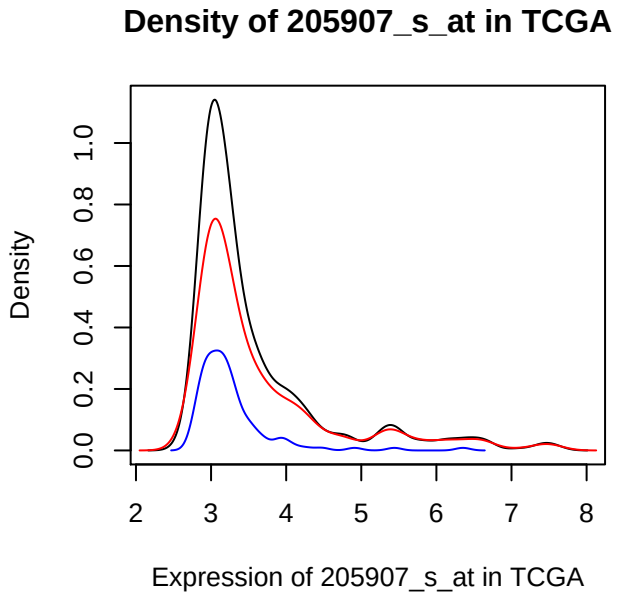

Expression of BCHE in Tothill
Expression
3
4
5
6
7
8
9
0
50
100
150
Tothill Samples

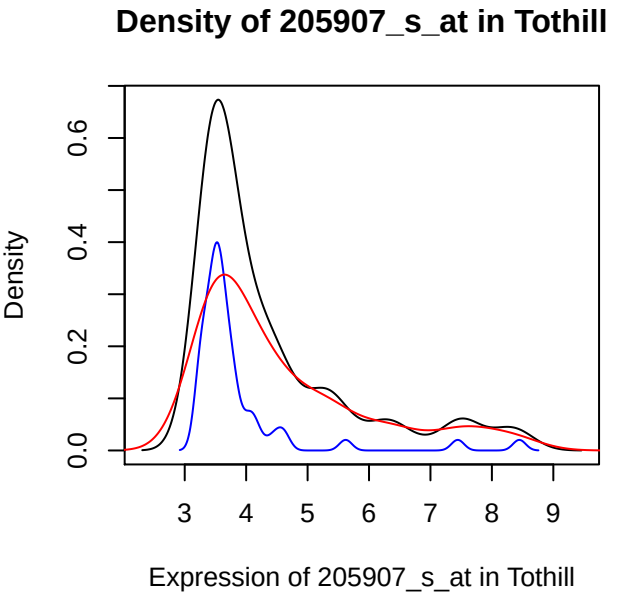

Density of 205433_at in TCGA
Density
0.0
0.4
0.8
1.2
2
3
4
5
6
7
8
Expression of 205433_at in TCGA

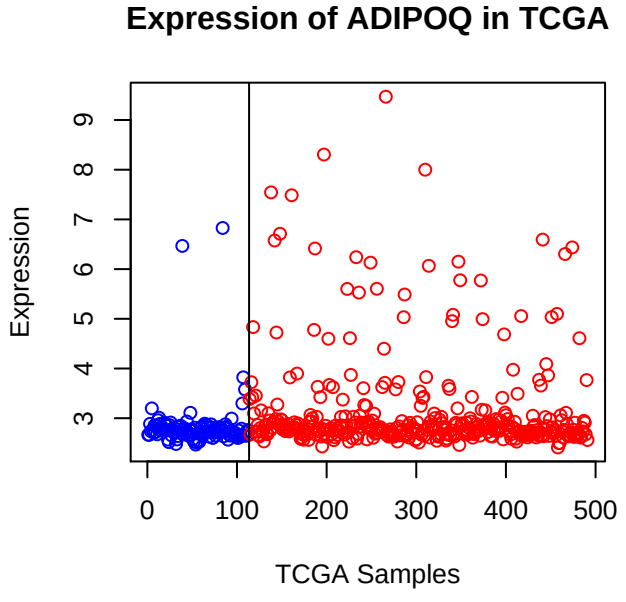

Density of 205433_at in Tothill
Density
0.0
0.1
0.2
0.3
0.4
2
4
6
8
10
Expression of 205433_at in Tothill

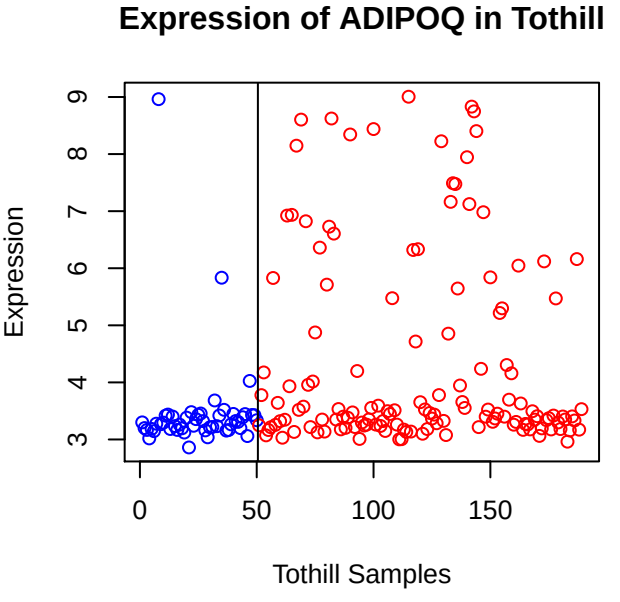

Expression of OMD in TCGA
Expression
3
4
5
6
7
0
100
200
300
400
500
TCGA Samples

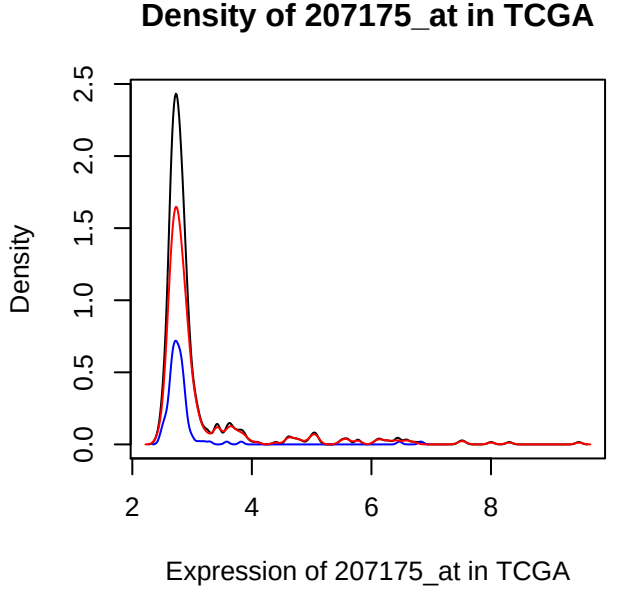

Expression of OMD in Tothill
Expression
3
4
5
6
7
8
0
50
100
150
Tothill Samples

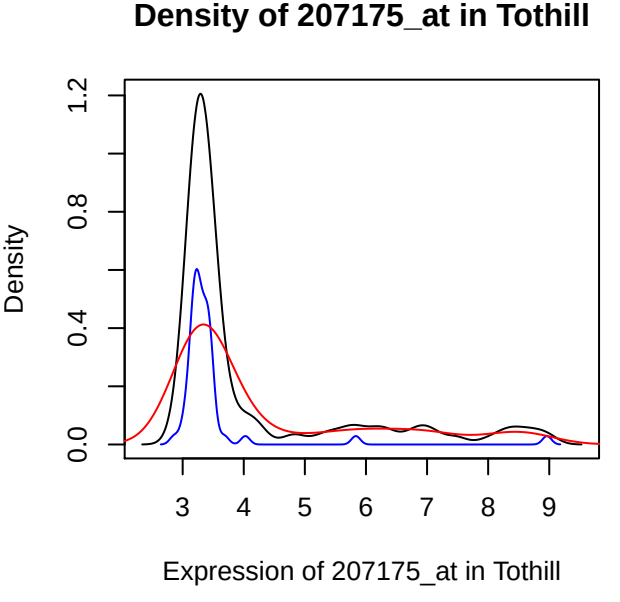

Density of 205907_s_at in TCGA
Density
0.0
0.2
0.4
0.6
0.8
1.0
2
3
4
5
6
7
8
Expression of 205907_s_at in TCGA

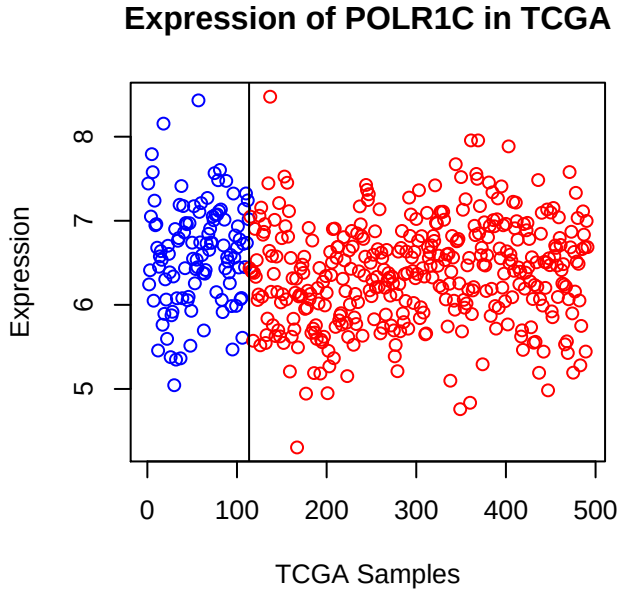

Density of 205907_s_at in Tothill
Density
0.0
0.2
0.4
0.6
3
4
5
6
7
8
9
Expression of 205907_s_at in Tothill

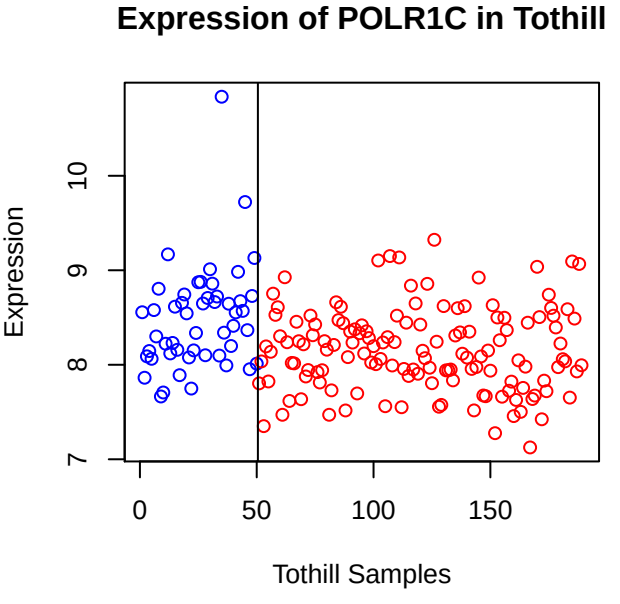

Expression of ADIPOQ in TCGA
Expression
3
4
5
6
7
8
9
0
100
200
300
400
500
TCGA Samples

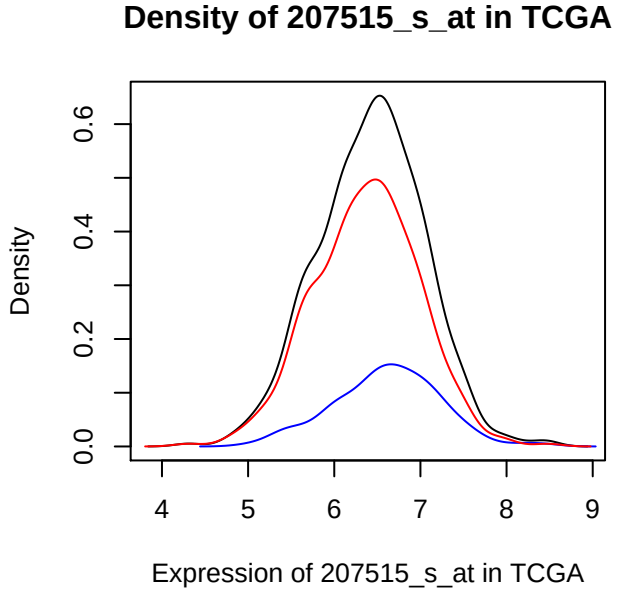

Expression of ADIPOQ in Tothill
Expression
3
4
5
6
7
8
9
0
50
100
150
Tothill Samples

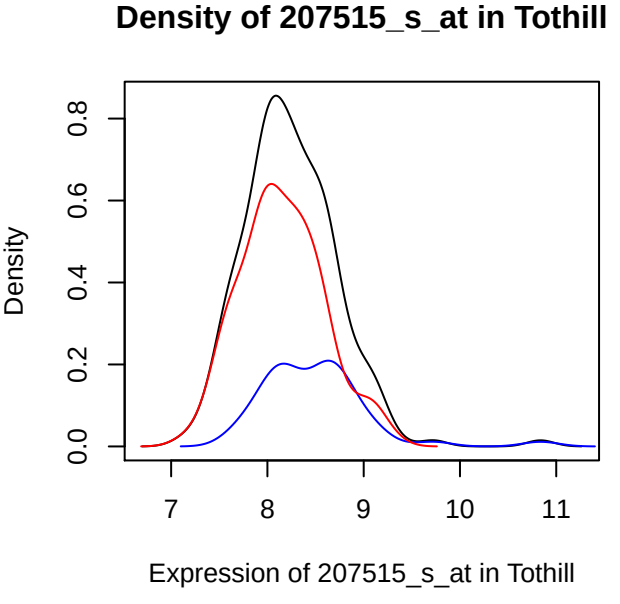

Density of 207175_at in TCGA
Density
0.0
0.5
1.0
1.5
2.0
2.5
2
4
6
8
Expression of 207175_at in TCGA

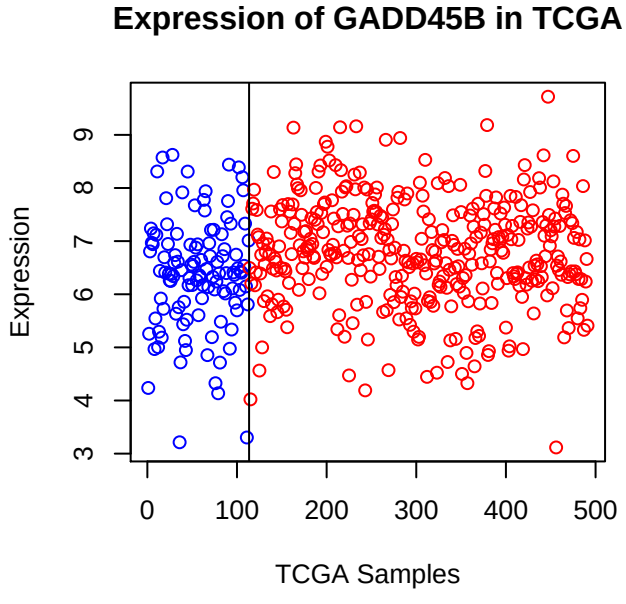

Density of 207175_at in Tothill
Density
0.0
0.4
0.8
1.2
3
4
5
6
7
8
9
Expression of 207175_at in Tothill

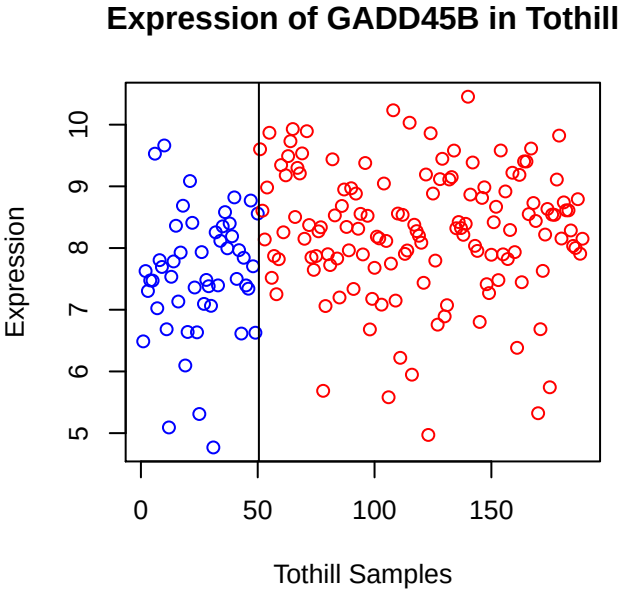

Expression of POLR1C in TCGA
Expression
5
6
7
8
0
100
200
300
400
500
TCGA Samples

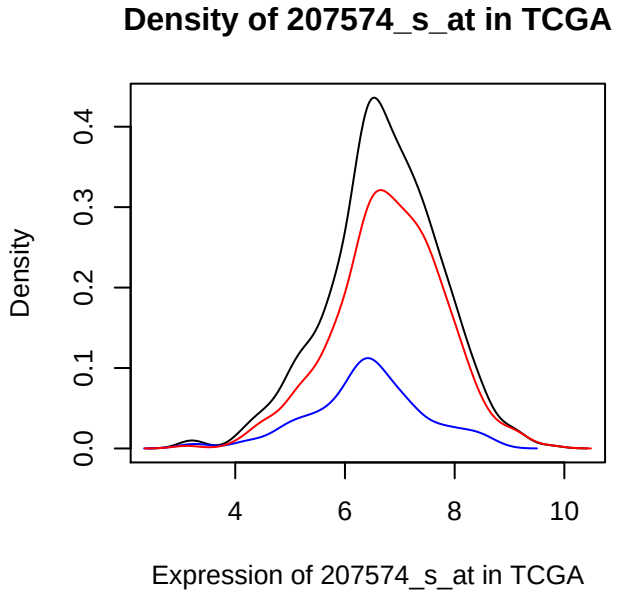

Expression of POLR1C in Tothill
Expression
7
8
9
10
0
50
100
150
Tothill Samples

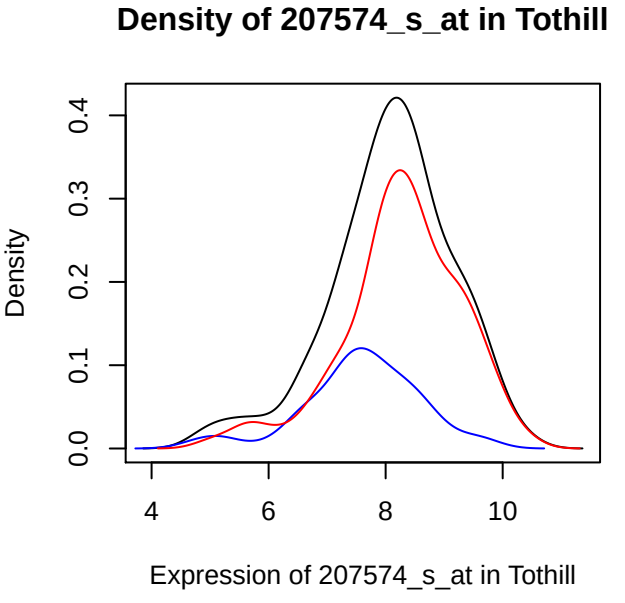

Density of 207515_s_at in TCGA
Density
0.0
0.2
0.4
0.6
4
5
6
7
8
9
Expression of 207515_s_at in TCGA

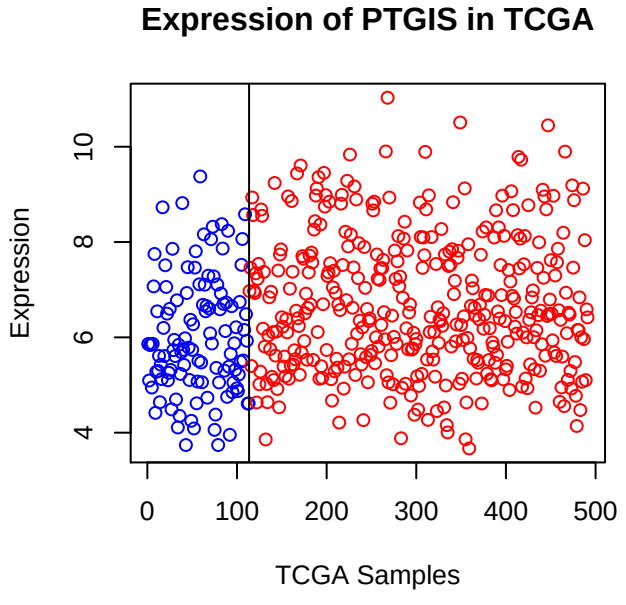

Density of 207515_s_at in Tothill
Density
0.0
0.2
0.4
0.6
0.8
7
8
9
10
11
Expression of 207515_s_at in Tothill

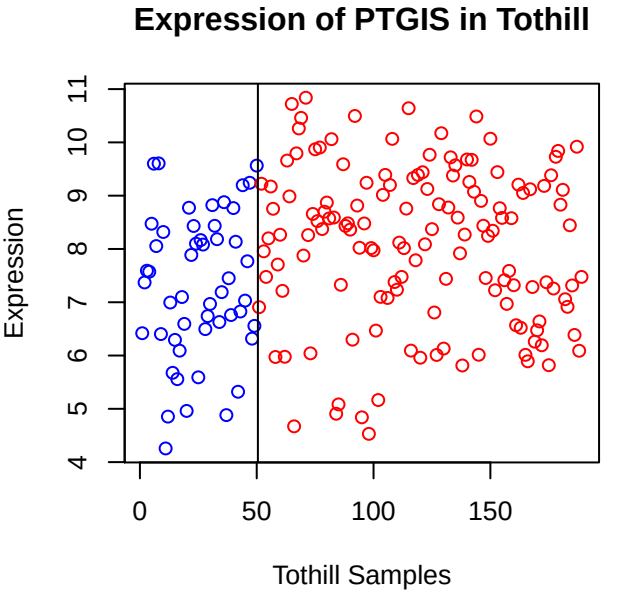

Expression of GADD45B in TCGA
Expression
3
4
5
6
7
8
9
0
100
200
300
400
500
TCGA Samples

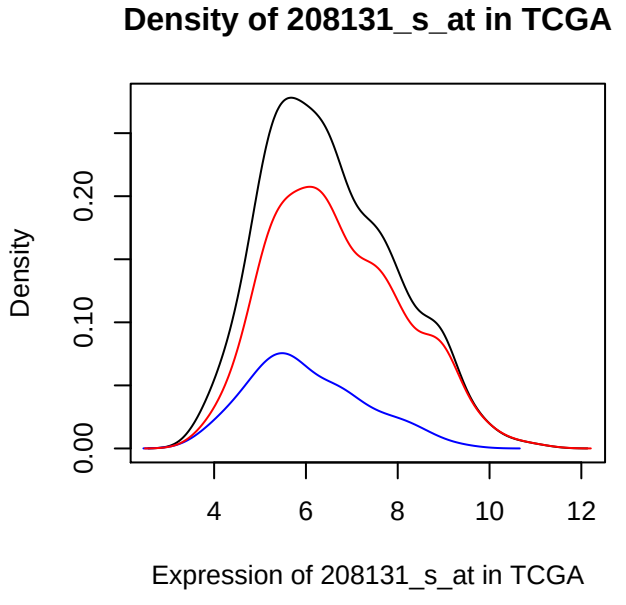

Expression of GADD45B in Tothill
Expression
5
6
7
8
9
10
0
50
100
150
Tothill Samples

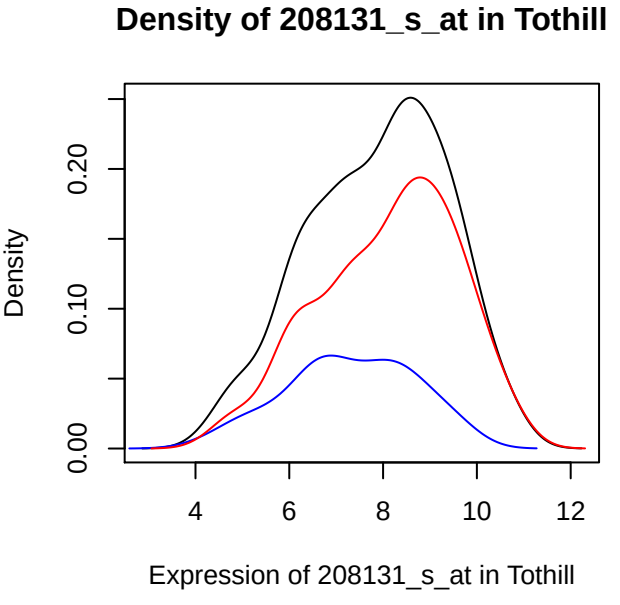

Density of 207574_s_at in TCGA
Density
0.0
0.1
0.2
0.3
0.4
4
6
8
10
Expression of 207574_s_at in TCGA

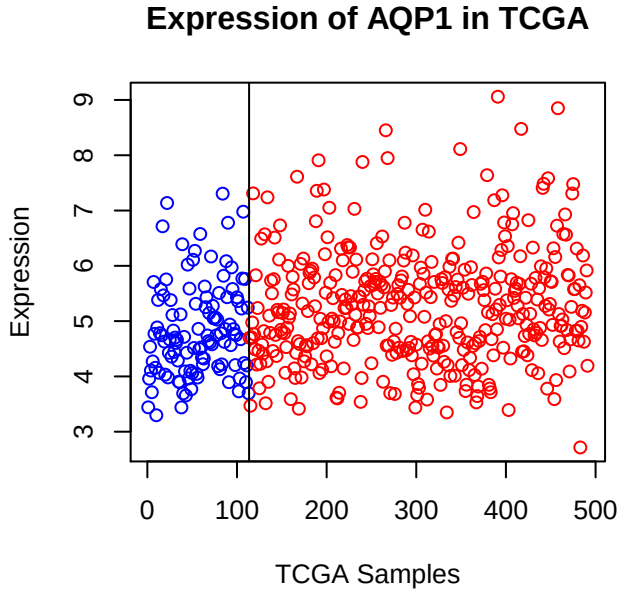

Density of 207574_s_at in Tothill
Density
0.0
0.1
0.2
0.3
0.4
4
6
8
10
Expression of 207574_s_at in Tothill

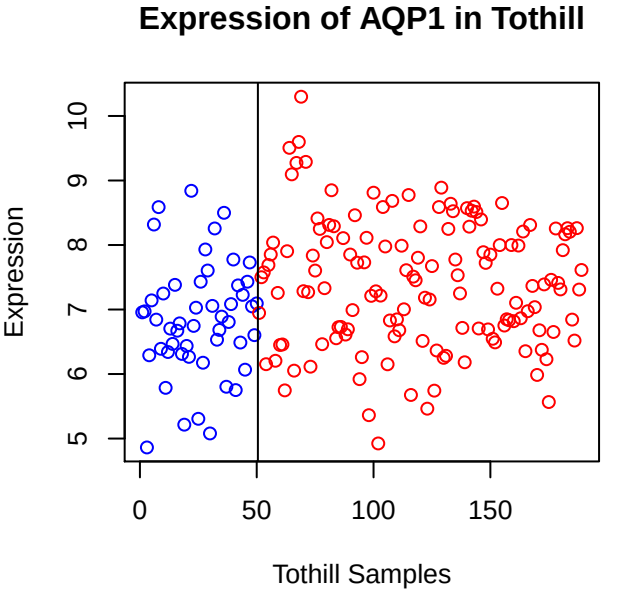

Expression of PTGIS in TCGA
Expression
4
6
8
10
0
100
200
300
400
500
TCGA Samples

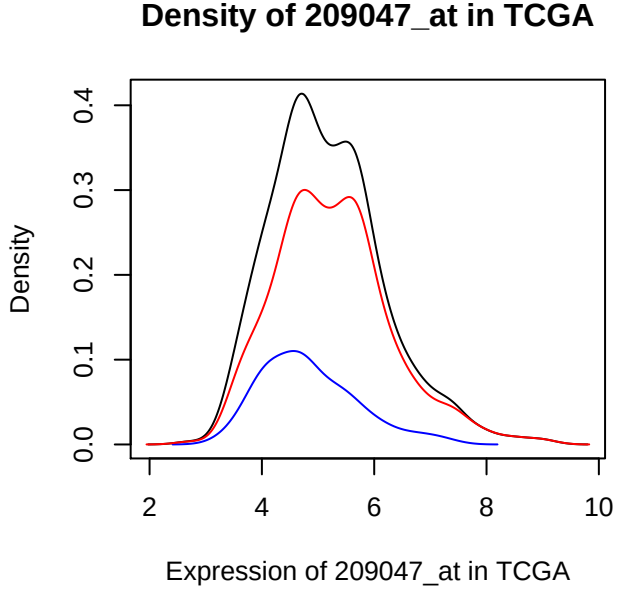

Expression of PTGIS in Tothill
Expression
4
5
6
7
8
9
10
11
0
50
100
150
Tothill Samples

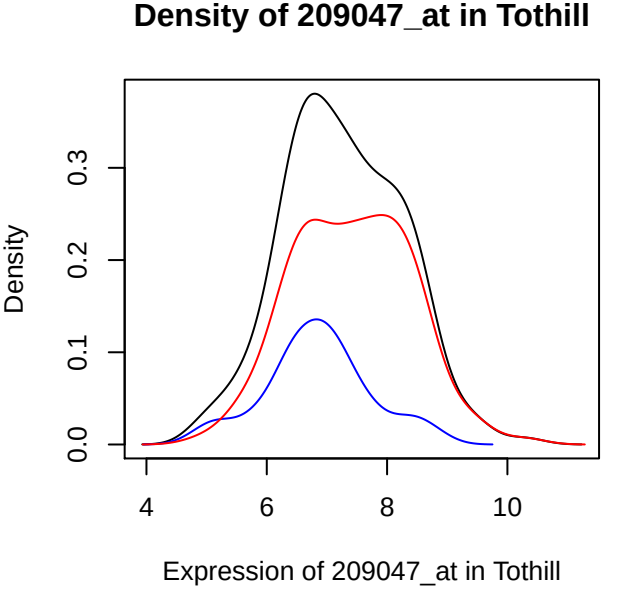

Density of 208131_s_at in TCGA
Density
0.00
0.10
0.20
4
6
8
10
12
Expression of 208131_s_at in TCGA

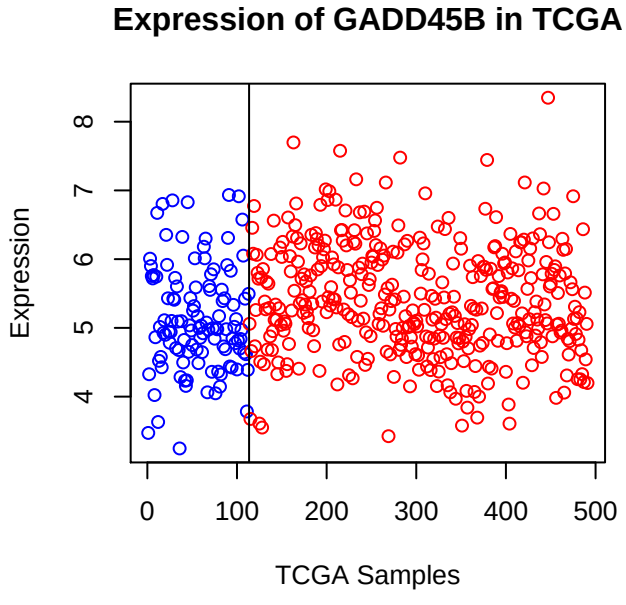

Density of 208131_s_at in Tothill
Density
0.00
0.10
0.20
4
6
8
10
12
Expression of 208131_s_at in Tothill

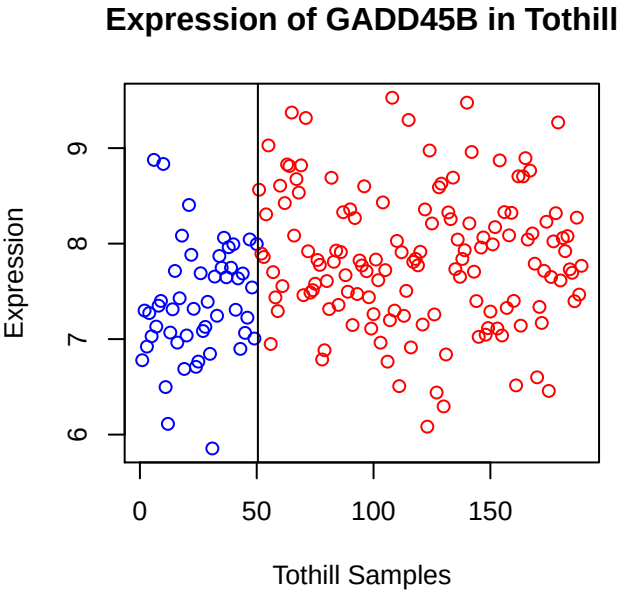

Expression of AQP1 in TCGA
Expression
3
4
5
6
7
8
9
0
100
200
300
400
500
TCGA Samples

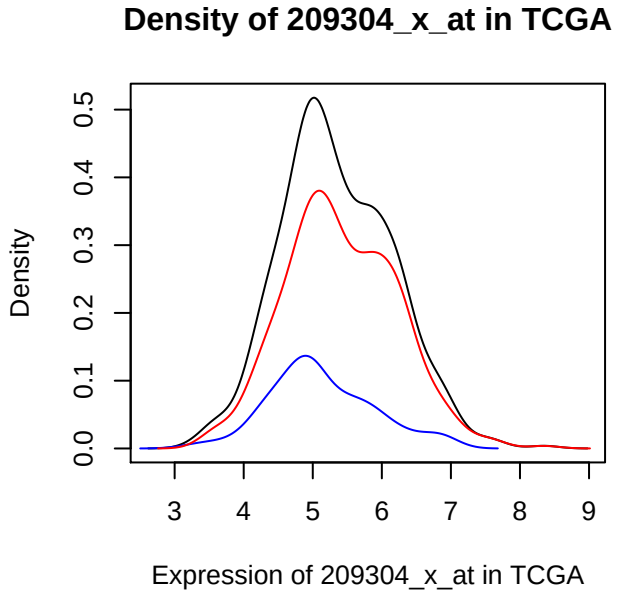

Expression of AQP1 in Tothill
Expression
5
6
7
8
9
10
0
50
100
150
Tothill Samples

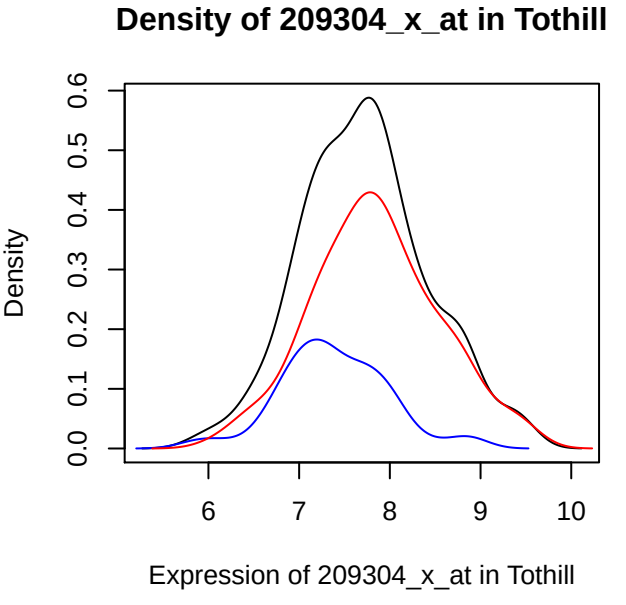

Density of 209047_at in TCGA
Density
0.0
0.1
0.2
0.3
0.4
2
4
6
8
10
Expression of 209047_at in TCGA

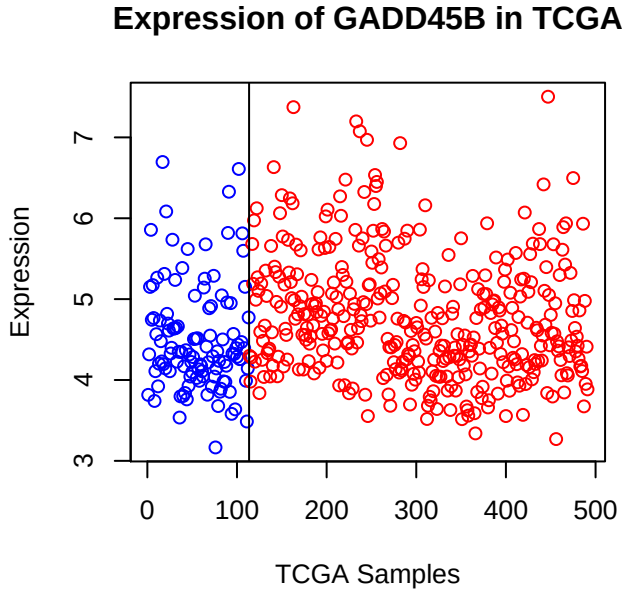

Density of 209047_at in Tothill
Density
0.0
0.1
0.2
0.3
4
6
8
10
Expression of 209047_at in Tothill

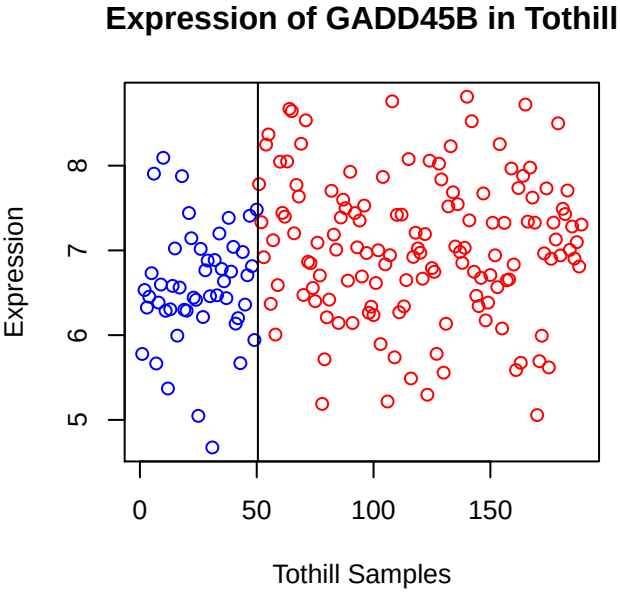

Expression of GADD45B in TCGA
Expression
TCGA Samples
0
100
200
300
400
500
4
5
6
7
8

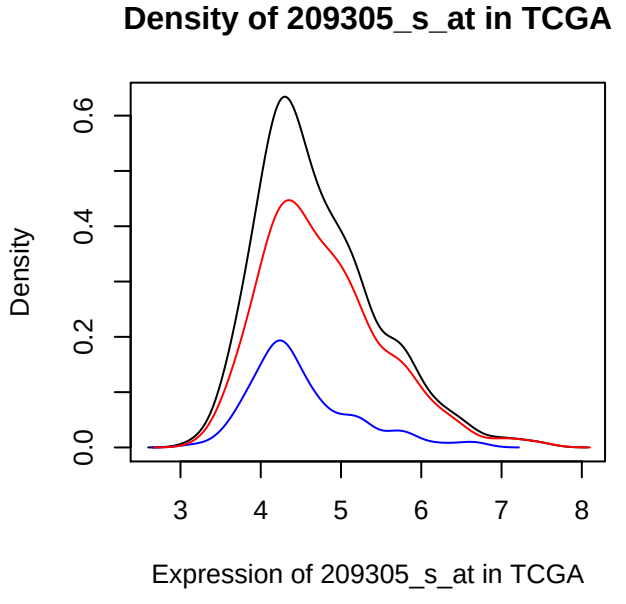

Expression of GADD45B in Tothill
Expression
Tothill Samples
0
50
100
150
6
7
8
9

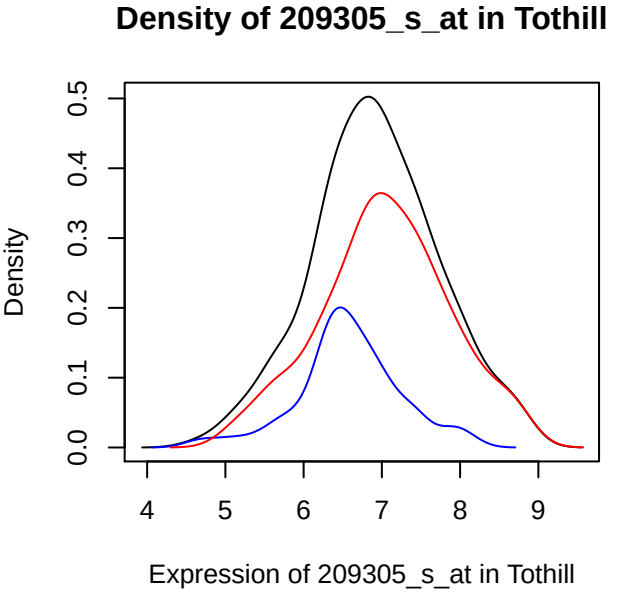

Density of 209304_x_at in TCGA
Density
Expression of 209304_x_at in TCGA
3
4
5
6
7
8
9
0.0
0.1
0.2
0.3
0.4
0.5

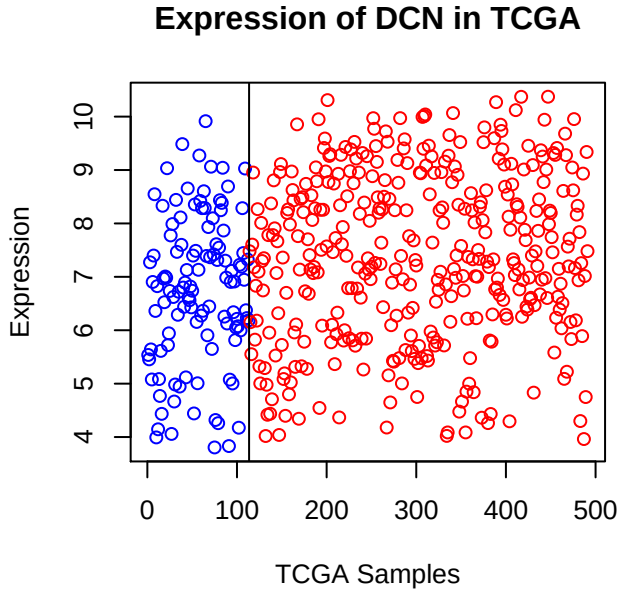

Density of 209304_x_at in Tothill
Density
Expression of 209304_x_at in Tothill
6
7
8
9
10
0.0
0.1
0.2
0.3
0.4
0.5
0.6

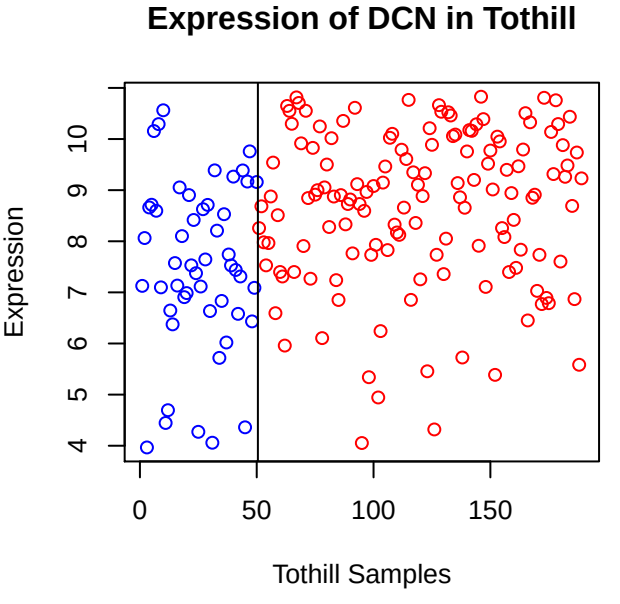

Expression of GADD45B in TCGA
Expression
3
4
5
6
7
0
100
200
300
400
500
TCGA Samples

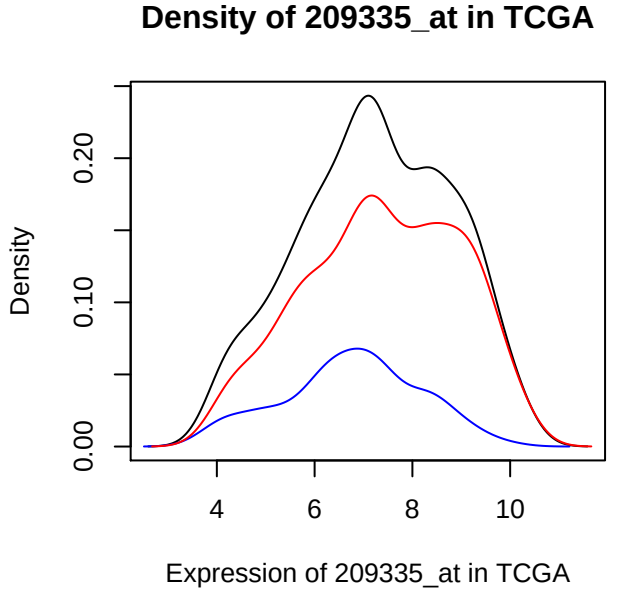

Expression of GADD45B in Tothill
Expression
5
6
7
8
0
50
100
150
Tothill Samples

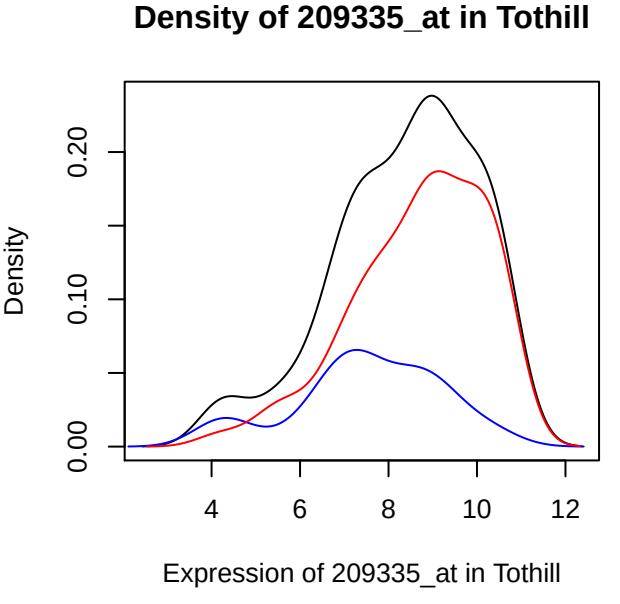

Density of 209305_s_at in TCGA
Density
0.0
0.2
0.4
0.6
3
4
5
6
7
8
Expression of 209305_s_at in TCGA

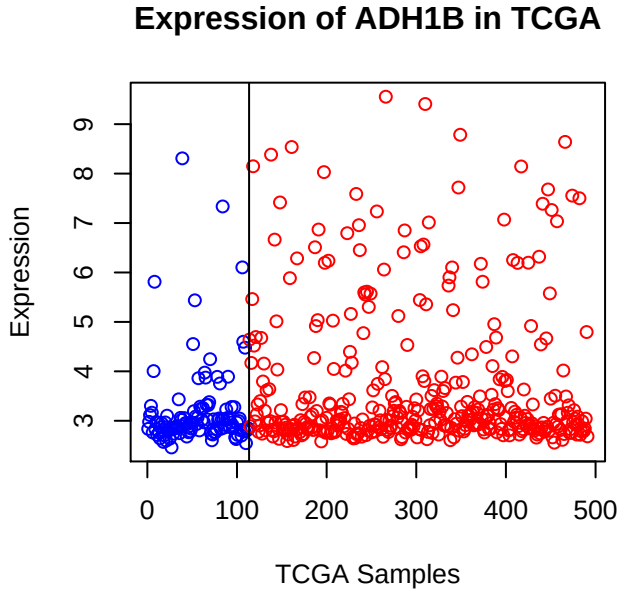

Density of 209305_s_at in Tothill
Density
0.0
0.1
0.2
0.3
0.4
0.5
4
5
6
7
8
9
Expression of 209305_s_at in Tothill

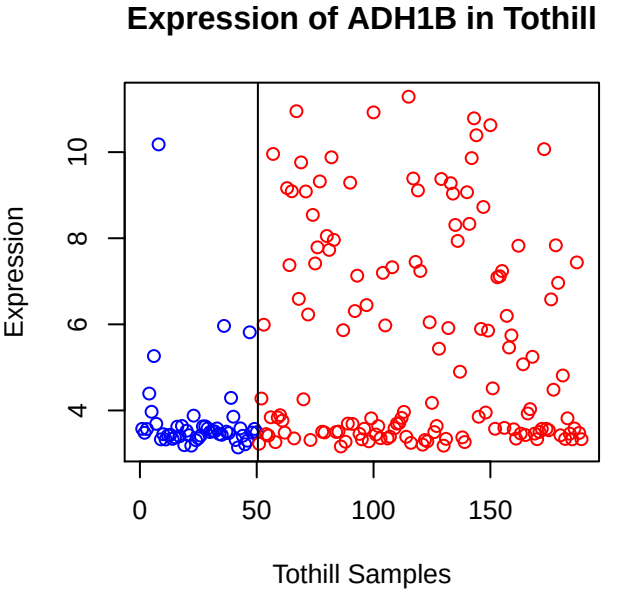

Expression of DCN in TCGA
Expression
4
5
6
7
8
9
10
0
100
200
300
400
500
TCGA Samples

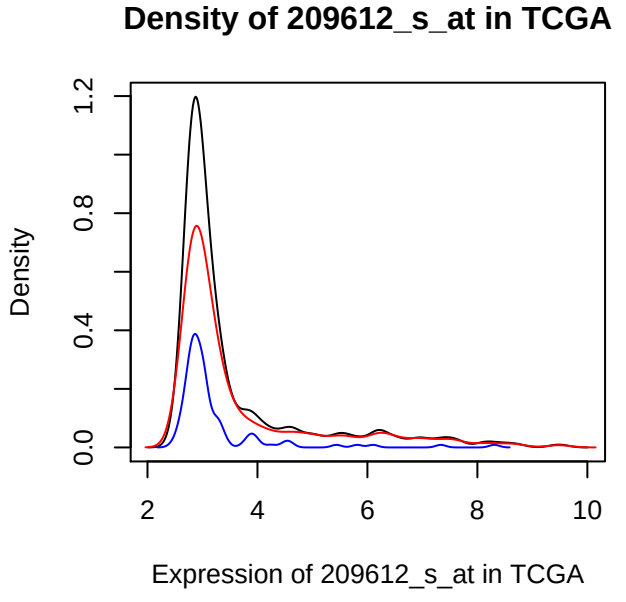

Expression of DCN in Tothill
Expression
4
5
6
7
8
9
10
0
50
100
150
Tothill Samples

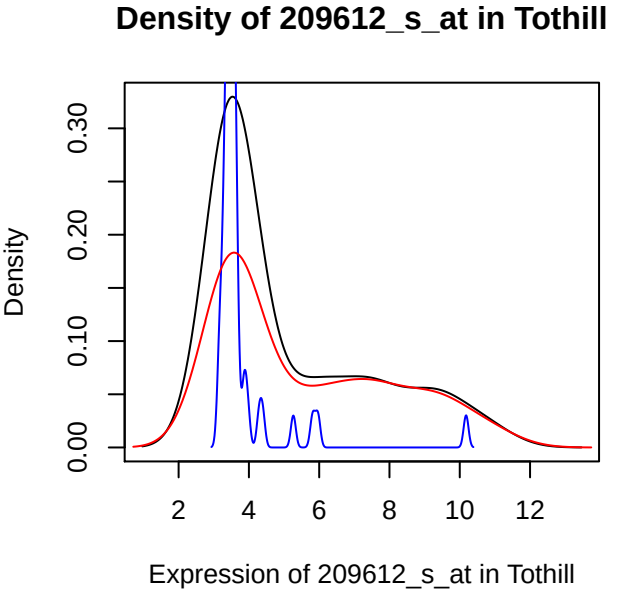

Density of 209335_at in TCGA
Density
0.00
0.10
0.20
4
6
8
10
Expression of 209335_at in TCGA

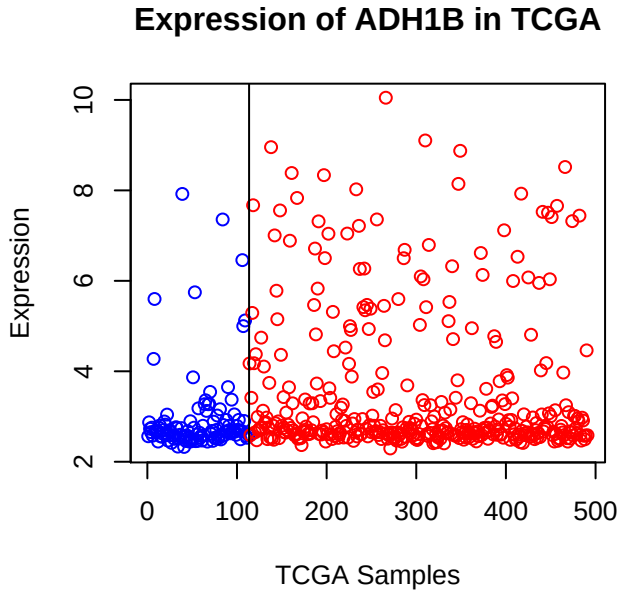

Density of 209335_at in Tothill
Density
0.00
0.10
0.20
4
6
8
10
12
Expression of 209335_at in Tothill

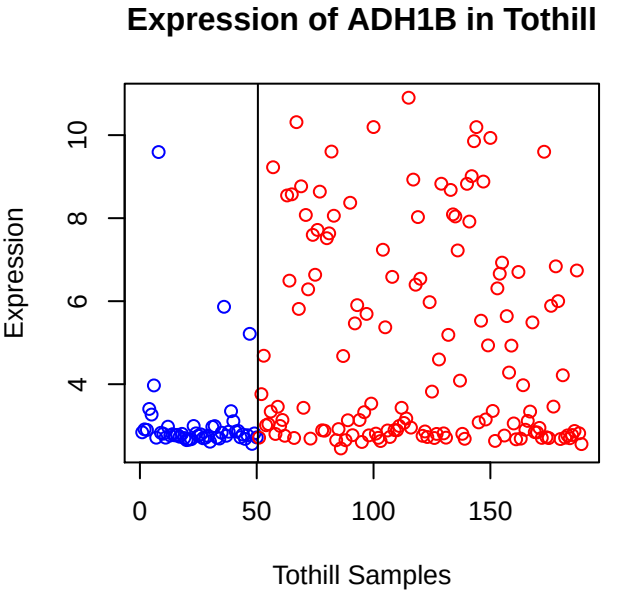

Expression of ADH1B in TCGA
Expression
3
4
5
6
7
8
9
0
100
200
300
400
500
TCGA Samples

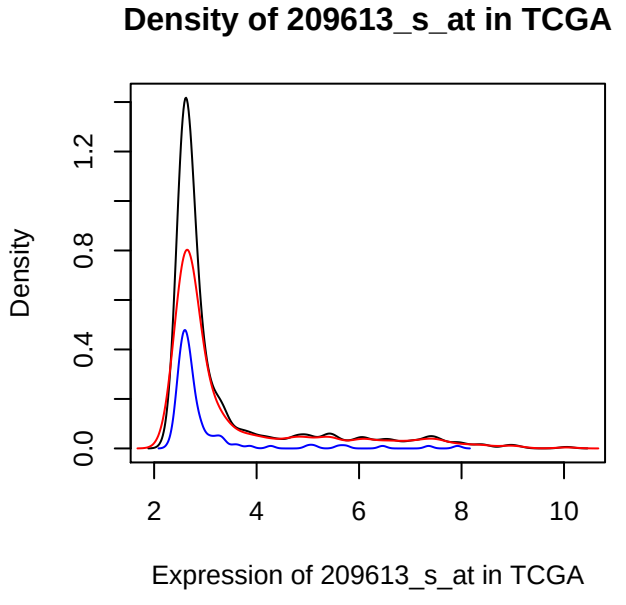

Expression of ADH1B in Tothill
Expression
4
6
8
10
0
50
100
150
Tothill Samples

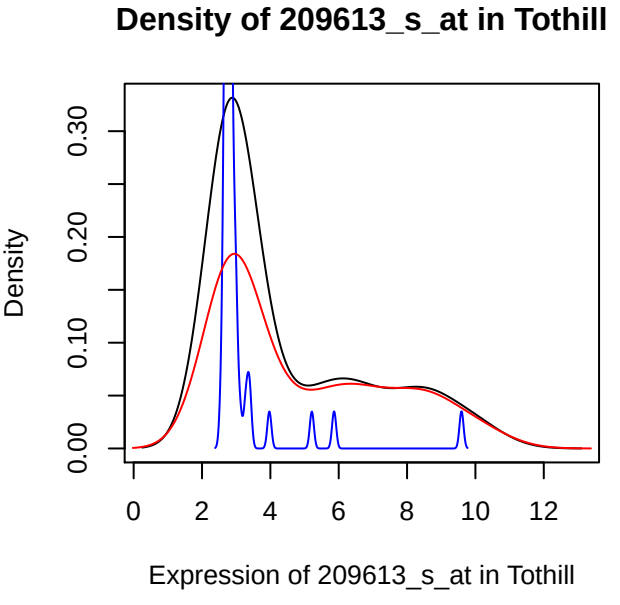

Density of 209612_s_at in TCGA
Density
0.0
0.4
0.8
1.2
2
4
6
8
10
Expression of 209612_s_at in TCGA

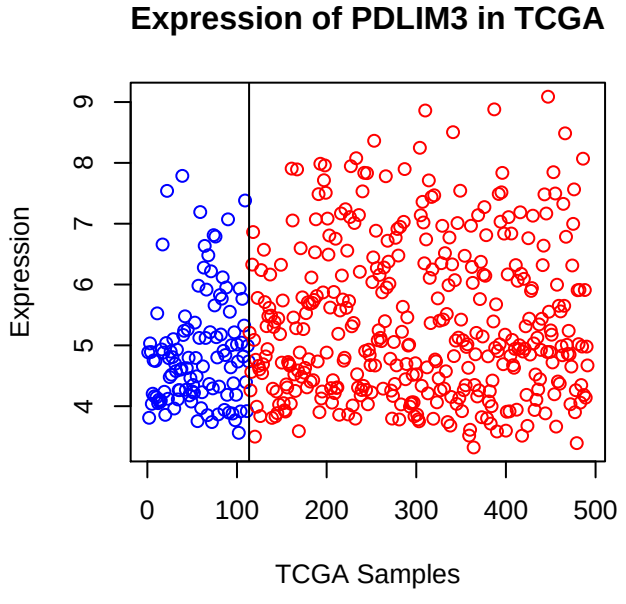

Density of 209612_s_at in Tothill
Density
0.00
0.10
0.20
0.30
2
4
6
8
10
12
Expression of 209612_s_at in Tothill

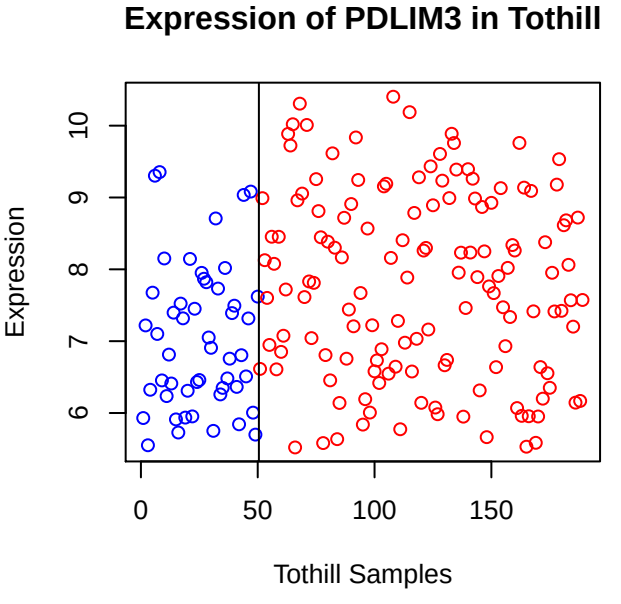

Expression of ADH1B in TCGA
Expression
2
4
6
8
10
0
100
200
300
400
500
TCGA Samples

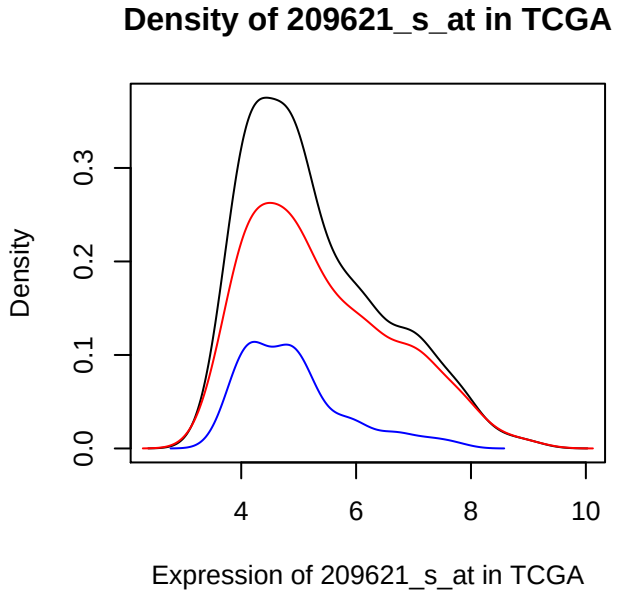

Expression of ADH1B in Tothill
Expression
4
6
8
10
0
50
100
150
Tothill Samples

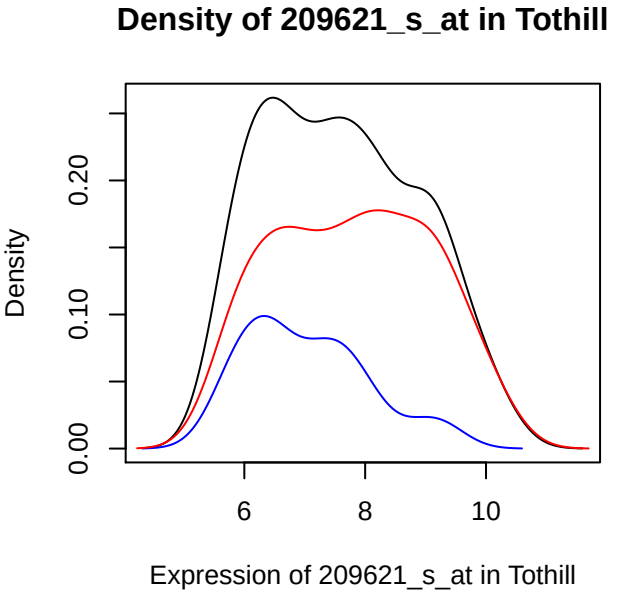

Density of 209613_s_at in TCGA
Density
0.0
0.4
0.8
1.2
2
4
6
8
10
Expression of 209613_s_at in TCGA

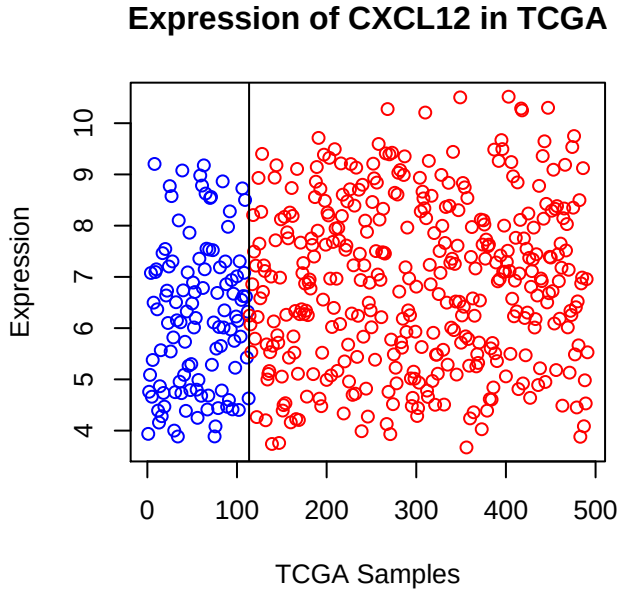

Density of 209613_s_at in Tothill
Density
0.00
0.10
0.20
0.30
0
2
4
6
8
10
12
Expression of 209613_s_at in Tothill

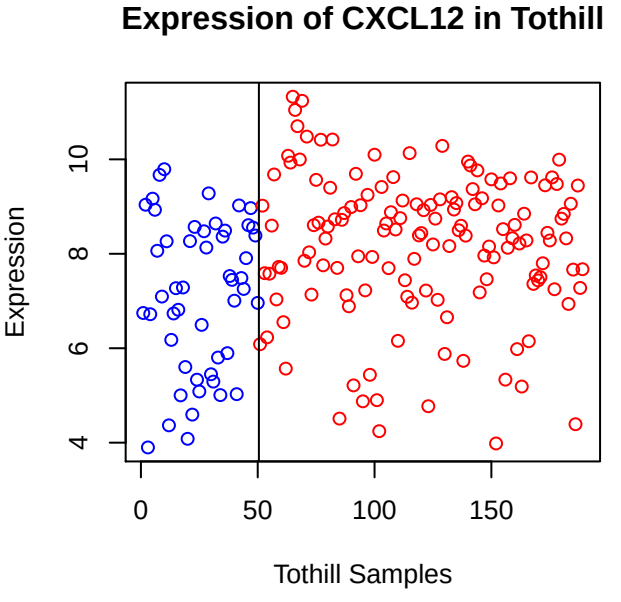

Expression of PDLIM3 in TCGA
Expression
TCGA Samples
0
100
200
300
400
500
4
5
6
7
8
9

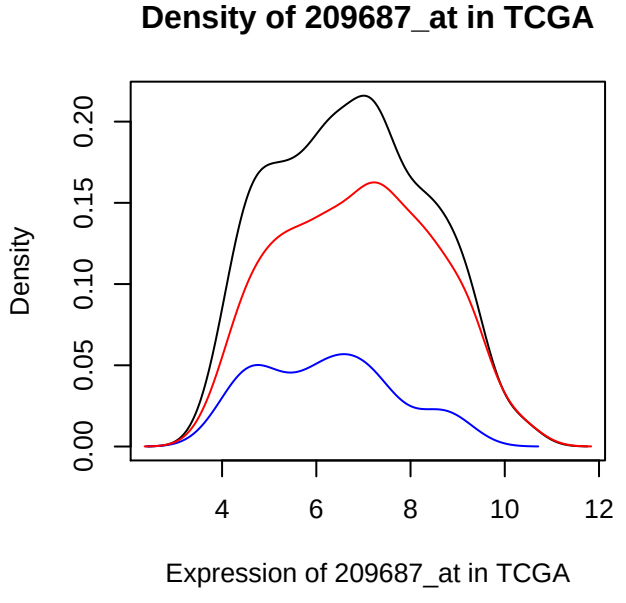

Expression of PDLIM3 in Tothill
Expression
Tothill Samples
0
50
100
150
6
7
8
9
10

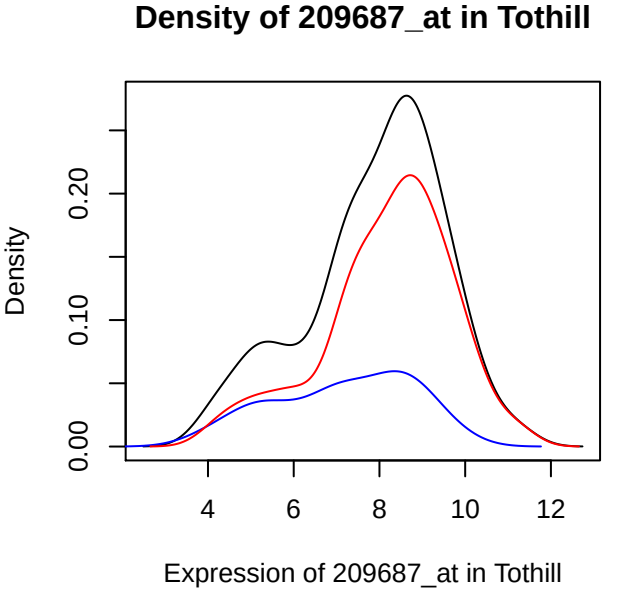

Density of 209621_s_at in TCGA
Density
Expression of 209621_s_at in TCGA
4
6
8
10
0.0
0.1
0.2
0.3

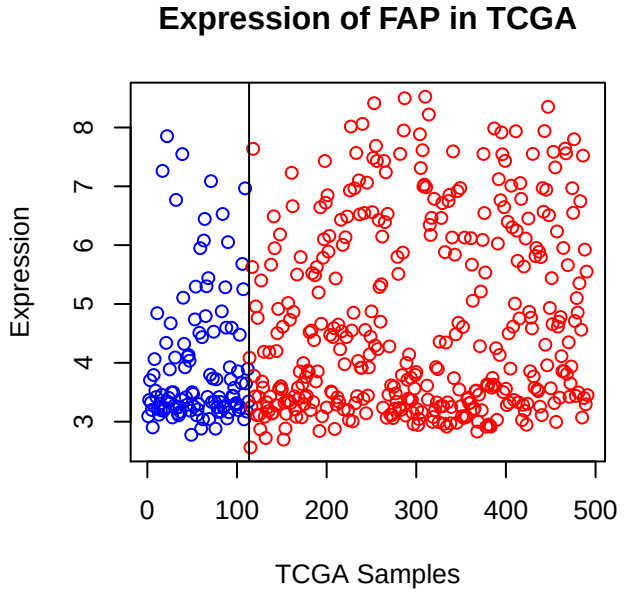

Density of 209621_s_at in Tothill
Density
Expression of 209621_s_at in Tothill
6
8
10
0.00
0.10
0.20

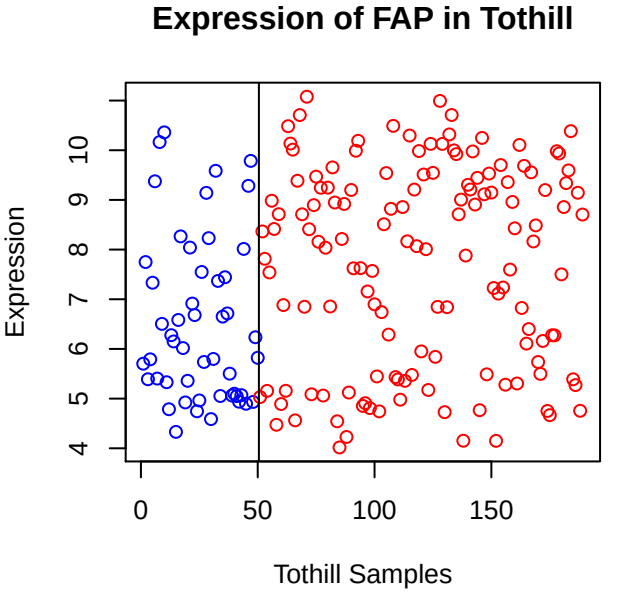

Expression of CXCL12 in TCGA
Expression
4
5
6
7
8
9
10
0
100
200
300
400
500
TCGA Samples

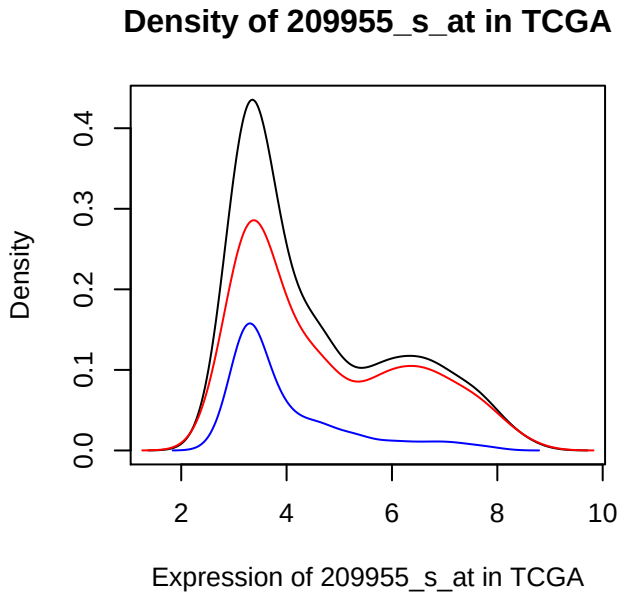

Expression of CXCL12 in Tothill
Expression
4
6
8
10
0
50
100
150
Tothill Samples

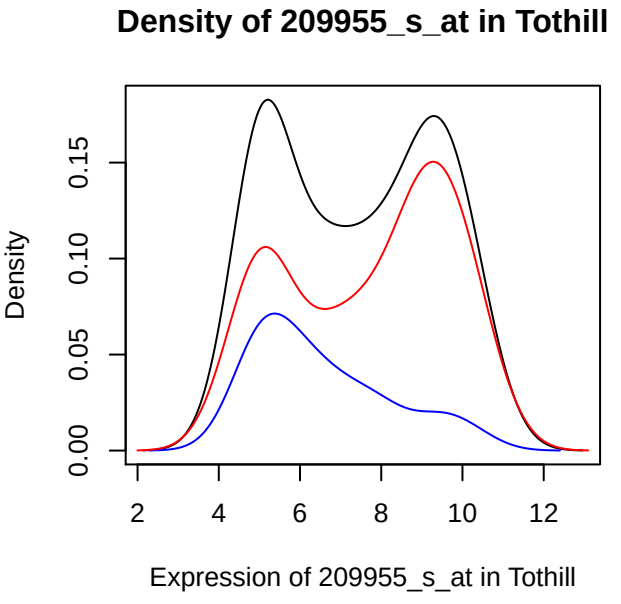

Density of 209687_at in TCGA
Density
0.00
0.05
0.10
0.15
0.20
4
6
8
10
12
Expression of 209687_at in TCGA

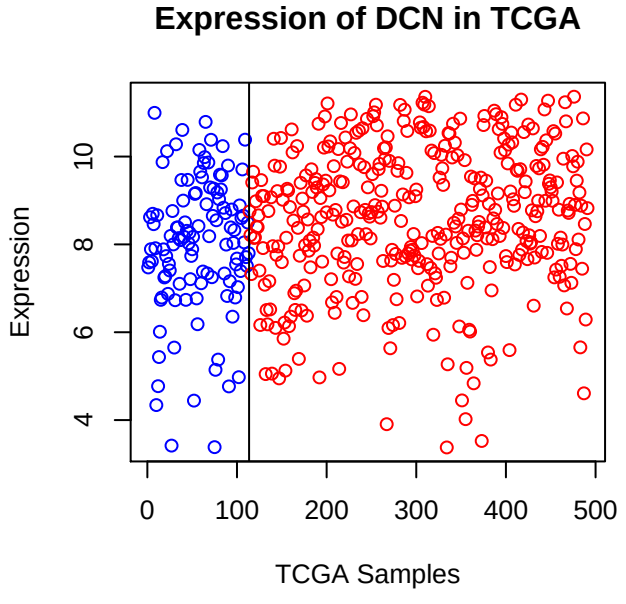

Density of 209687_at in Tothill
Density
0.00
0.10
0.20
4
6
8
10
12
Expression of 209687_at in Tothill

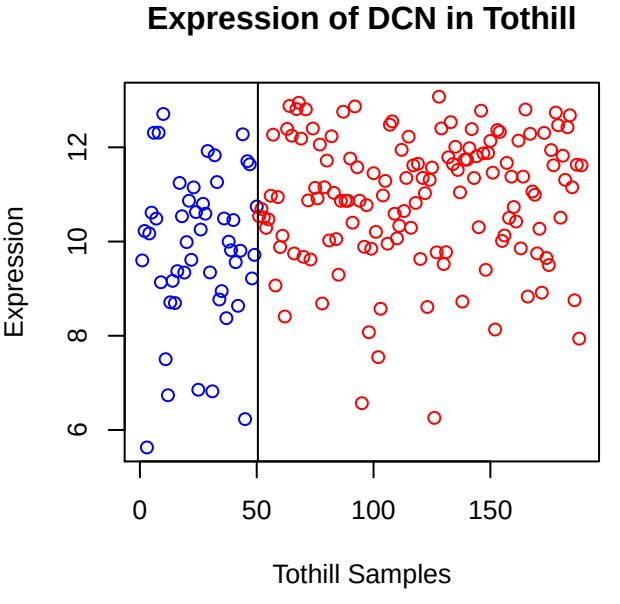

Expression of FAP in TCGA
Expression
3
4
5
6
7
8
0
100
200
300
400
500
TCGA Samples

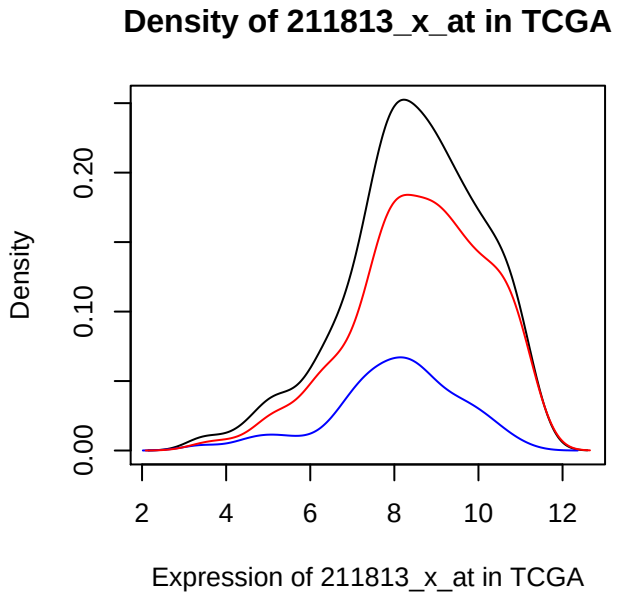

Expression of FAP in Tothill
Expression
4
5
6
7
8
9
10
0
50
100
150
Tothill Samples

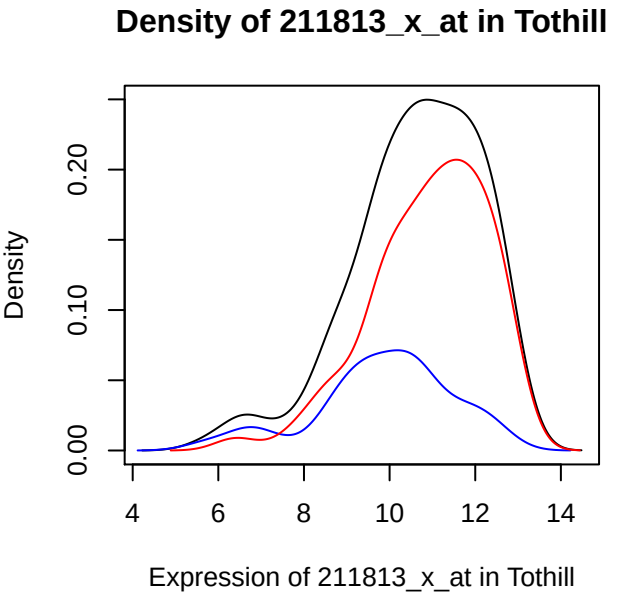

Density of 209955_s_at in TCGA
Density
0.0
0.1
0.2
0.3
0.4
2
4
6
8
10
Expression of 209955_s_at in TCGA

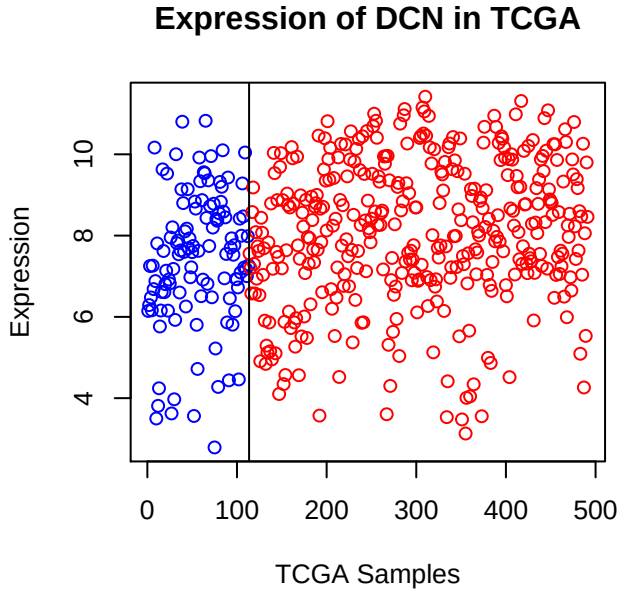

Density of 209955_s_at in Tothill
Density
0.00
0.05
0.10
0.15
2
4
6
8
10
12
Expression of 209955_s_at in Tothill

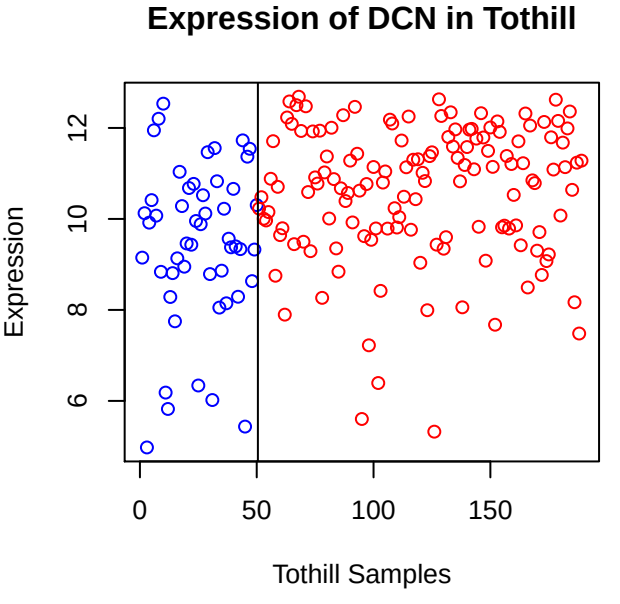

Expression of DCN in TCGA
Expression
4
6
8
10
0
100
200
300
400
500
TCGA Samples

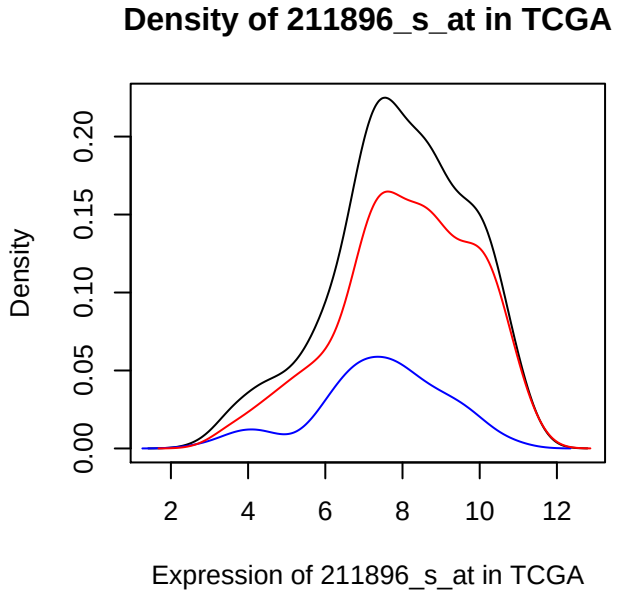

Expression of DCN in Tothill
Expression
6
8
10
12
0
50
100
150
Tothill Samples

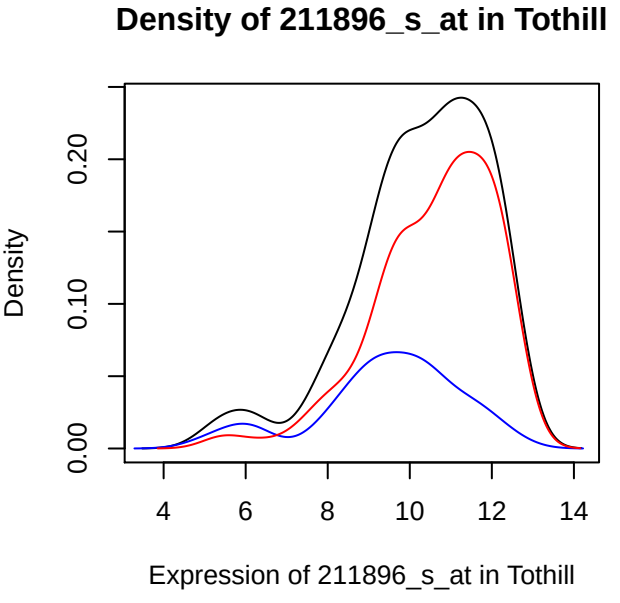

Density of 211813_x_at in TCGA
Density
0.00
0.10
0.20
2
4
6
8
10
12
Expression of 211813_x_at in TCGA

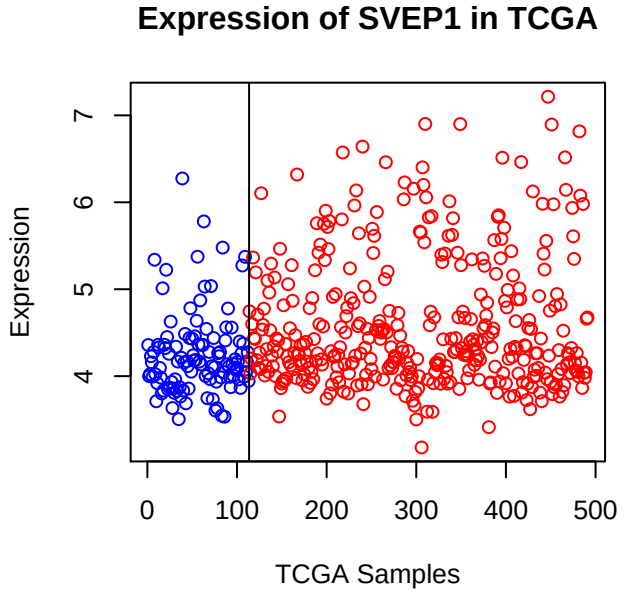

Density of 211813_x_at in Tothill
Density
0.00
0.10
0.20
4
6
8
10
12
14
Expression of 211813_x_at in Tothill

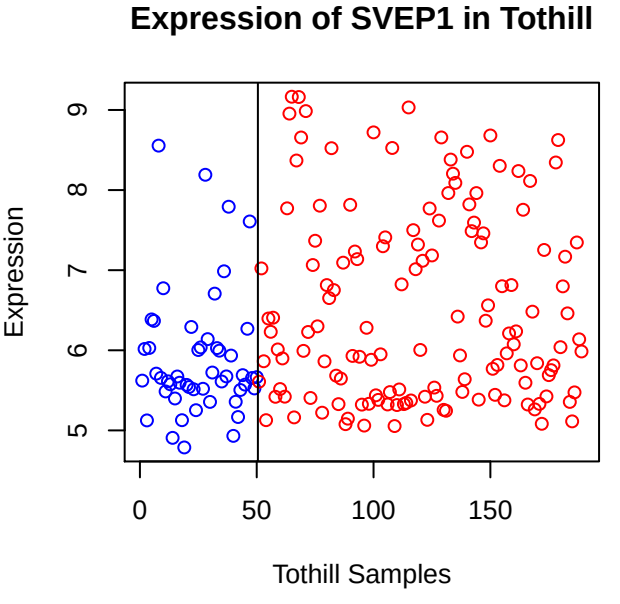

Expression of DCN in TCGA
Expression
4
6
8
10
0
100
200
300
400
500
TCGA Samples

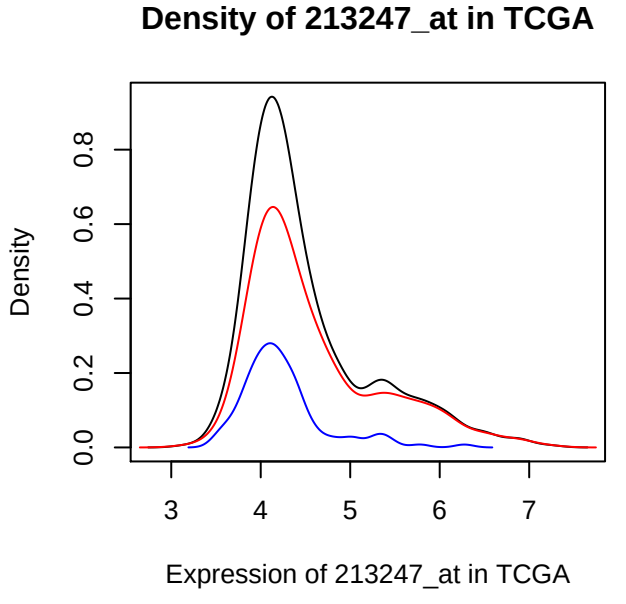

Expression of DCN in Tothill
Expression
6
8
10
12
0
50
100
150
Tothill Samples

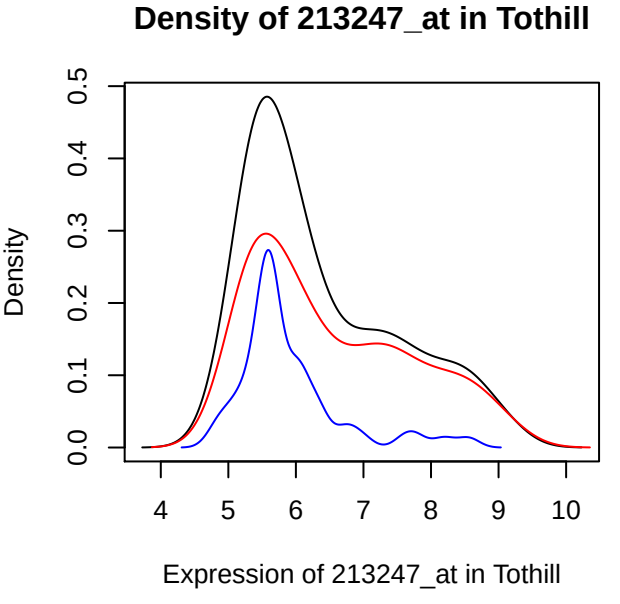

Density of 211896_s_at in TCGA
Density
0.00
0.05
0.10
0.15
0.20
2
4
6
8
10
12
Expression of 211896_s_at in TCGA

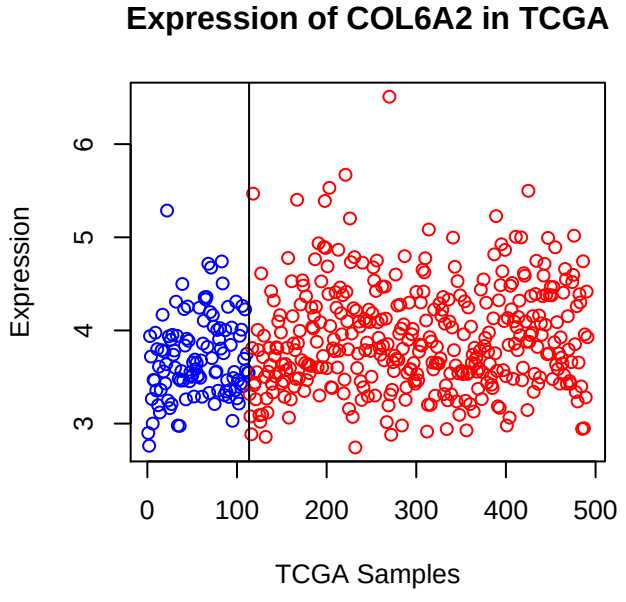

Density of 211896_s_at in Tothill
Density
0.00
0.10
0.20
4
6
8
10
12
14
Expression of 211896_s_at in Tothill

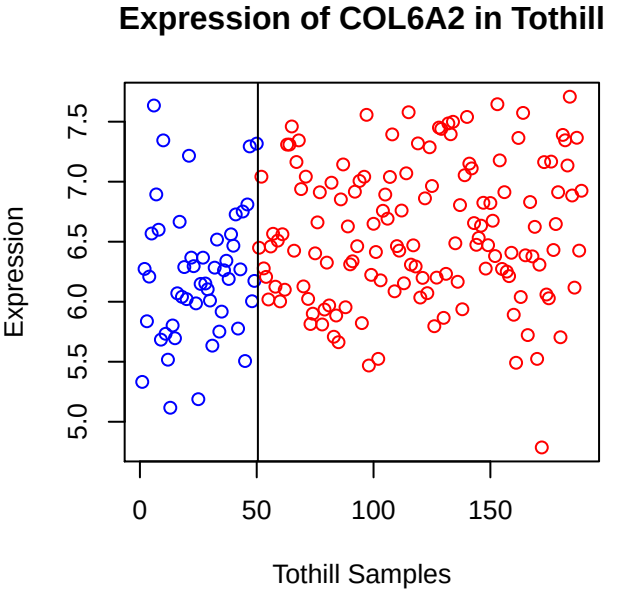

Expression of SVEP1 in TCGA
Expression
4
5
6
7
0
100
200
300
400
500
TCGA Samples

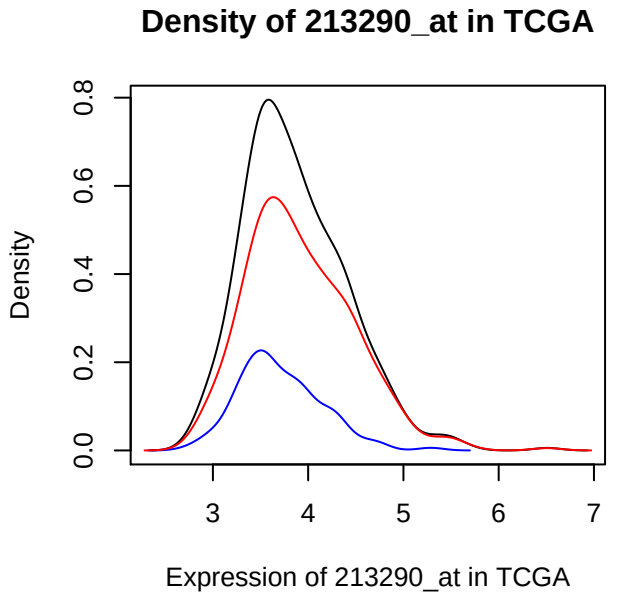

Expression of SVEP1 in Tothill
Expression
5
6
7
8
9
0
50
100
150
Tothill Samples

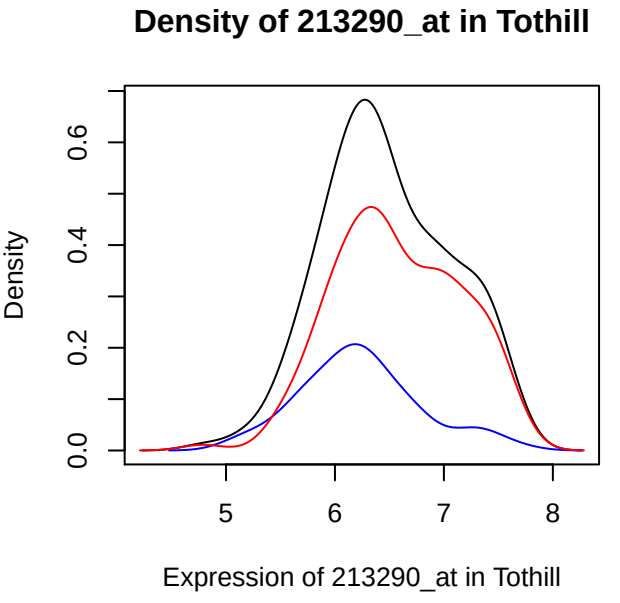

Density of 213247_at in TCGA
Density
0.0
0.2
0.4
0.6
0.8
3
4
5
6
7
Expression of 213247_at in TCGA

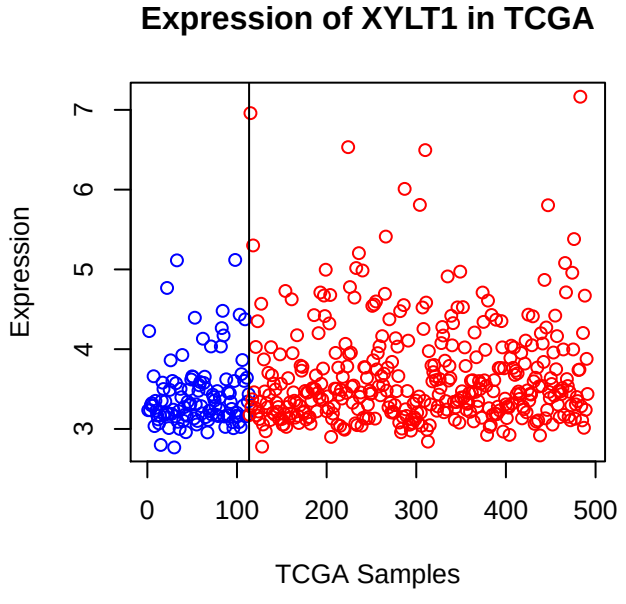

Density of 213247_at in Tothill
Density
0.0
0.1
0.2
0.3
0.4
0.5
4
5
6
7
8
9
10
Expression of 213247_at in Tothill

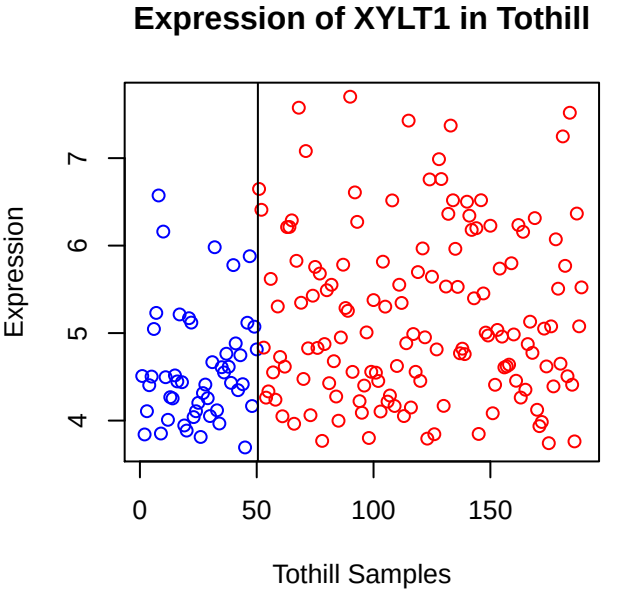

Expression of COL6A2 in TCGA
Expression
3
4
5
6
0
100
200
300
400
500
TCGA Samples

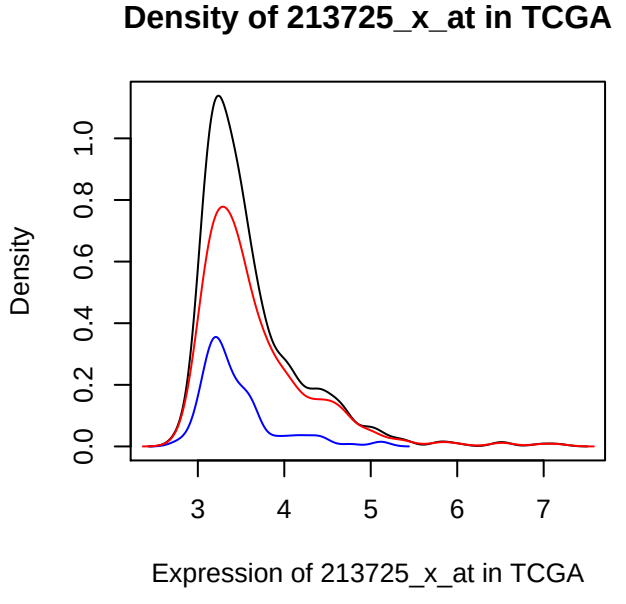

Expression of COL6A2 in Tothill
Expression
5.0
5.5
6.0
6.5
7.0
7.5
0
50
100
150
Tothill Samples

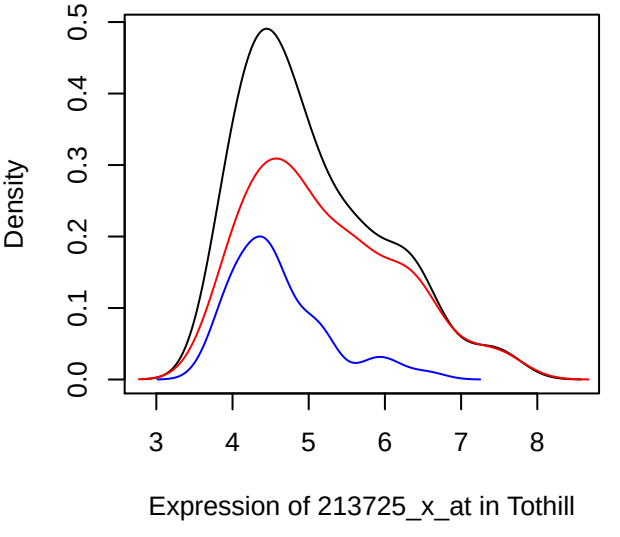

Density of 213290_at in TCGA
Density
0.0
0.2
0.4
0.6
0.8
3
4
5
6
7
Expression of 213290_at in TCGA

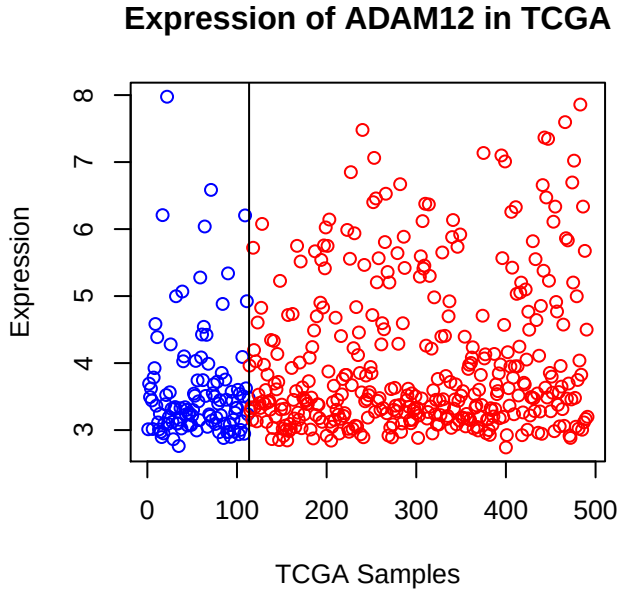

Density of 213290_at in Tothill
Density
0.0
0.2
0.4
0.6
5
6
7
8
Expression of 213290_at in Tothill

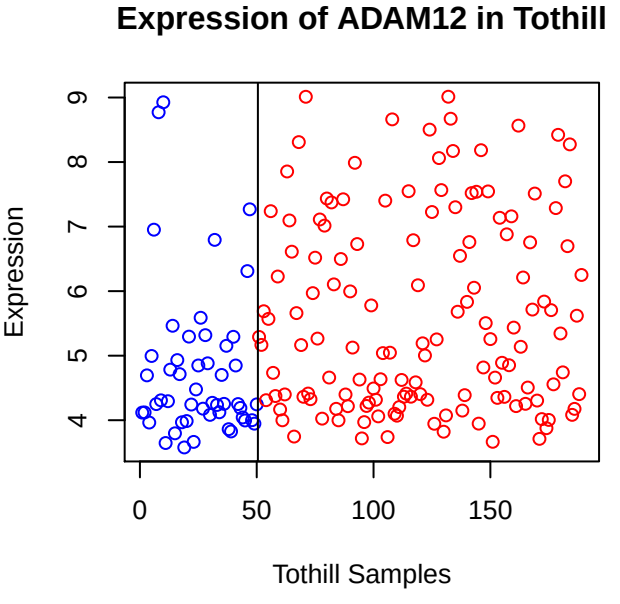

Expression of XYLT1 in TCGA
Expression
3
4
5
6
7
0
100
200
300
400
500
TCGA Samples

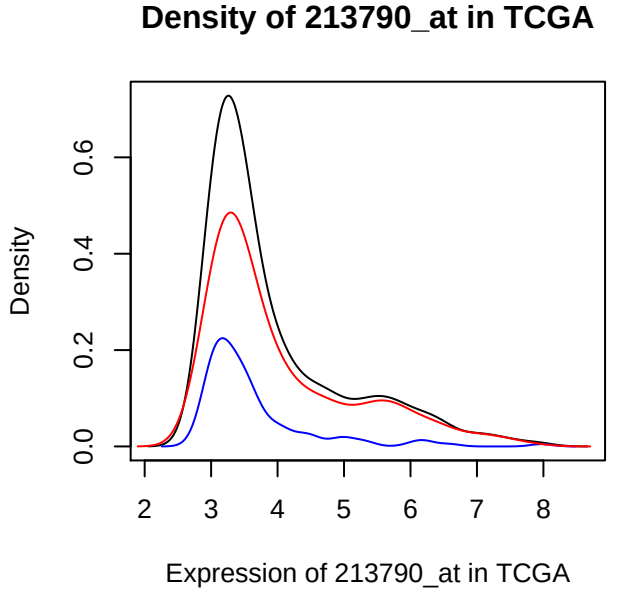

Expression of XYLT1 in Tothill
Expression
4
5
6
7
0
50
100
150
Tothill Samples

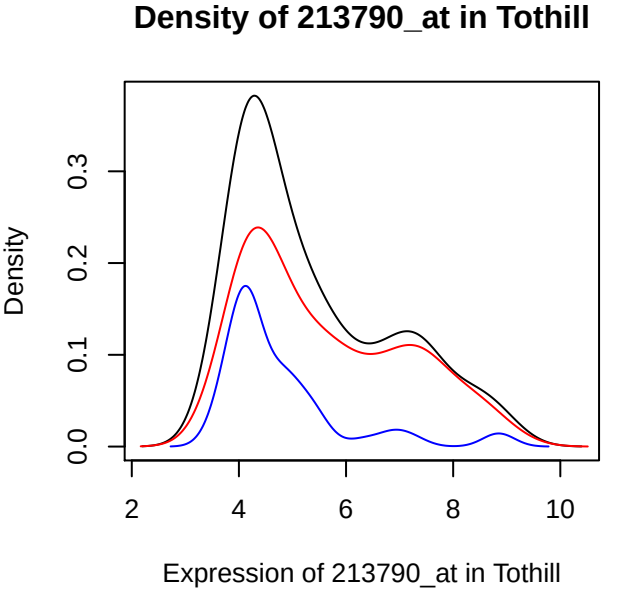

Density of 213725_x_at in TCGA
Density
0.0
0.2
0.4
0.6
0.8
1.0
3
4
5
6
7
Expression of 213725_x_at in TCGA

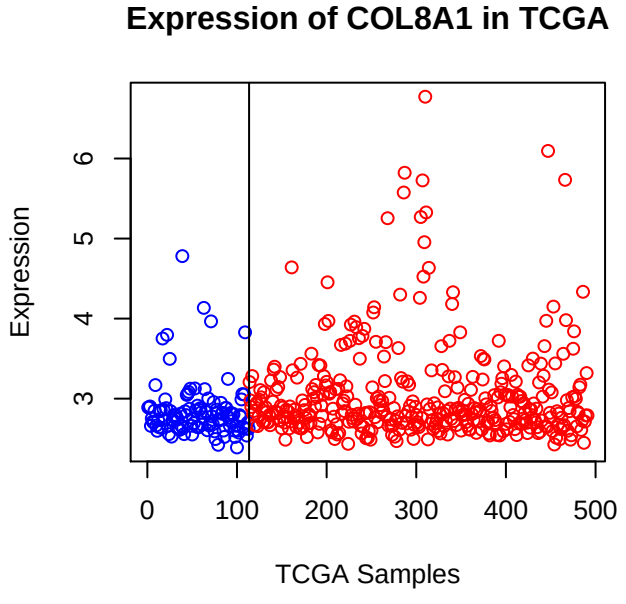

Density of 213725_x_at in Tothill
Density
0.0
0.1
0.2
0.3
0.4
0.5
3
4
5
6
7
8
Expression of 213725_x_at in Tothill

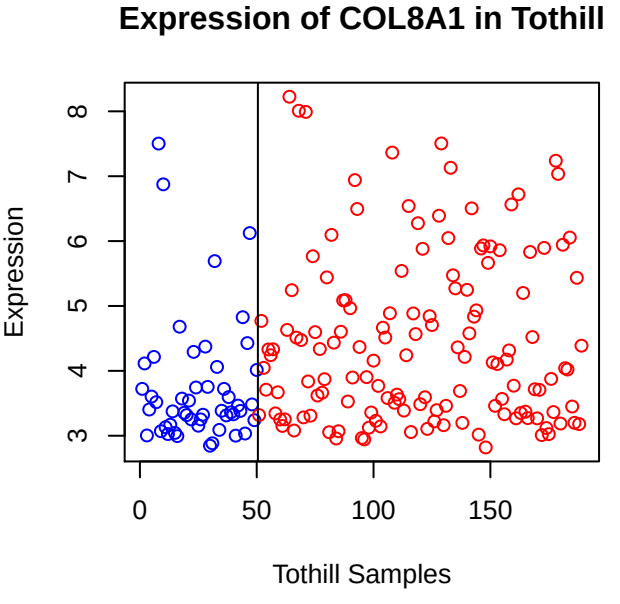

Expression of ADAM12 in TCGA
Expression
3
4
5
6
7
8
0
100
200
300
400
500
TCGA Samples

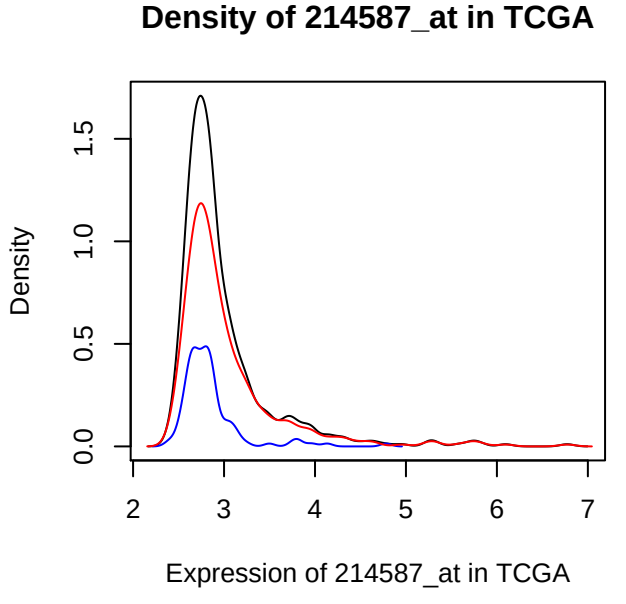

Expression of ADAM12 in Tothill
Expression
4
5
6
7
8
9
0
50
100
150
Tothill Samples

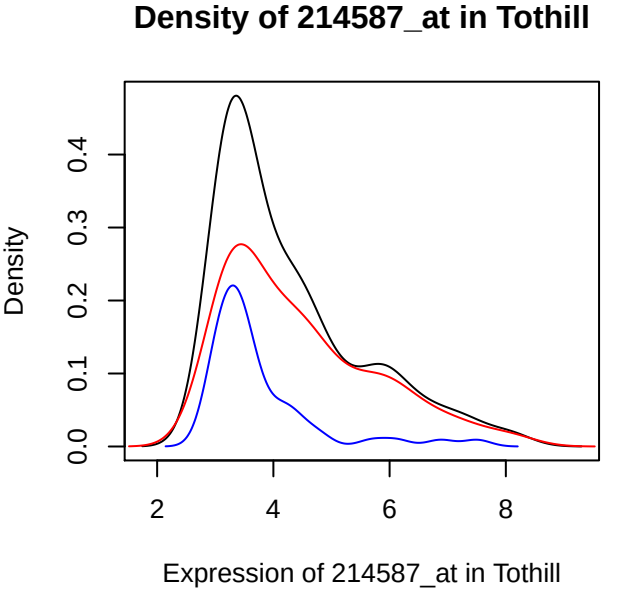

Density of 213790_at in TCGA
Density
0.0
0.2
0.4
0.6
2
3
4
5
6
7
8
Expression of 213790_at in TCGA

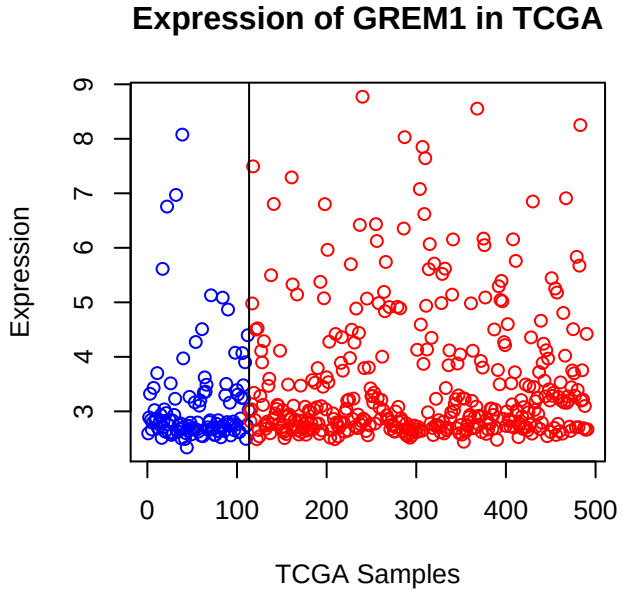

Density of 213790_at in Tothill
Density
0.0
0.1
0.2
0.3
2
4
6
8
10
Expression of 213790_at in Tothill

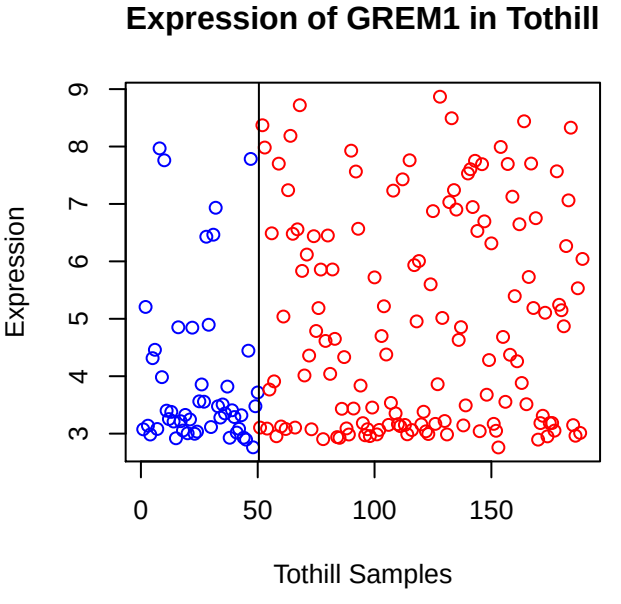

Expression of COL8A1 in TCGA
Expression
3
4
5
6
0
100
200
300
400
500
TCGA Samples

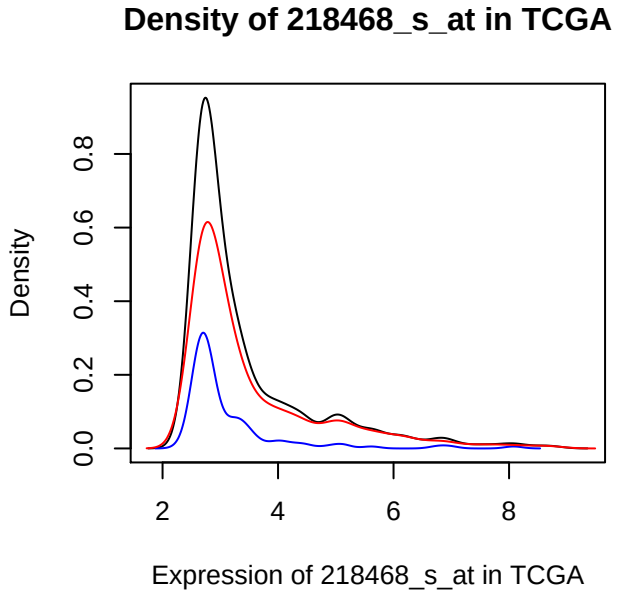

Expression of COL8A1 in Tothill
Expression
3
4
5
6
7
8
0
50
100
150
Tothill Samples

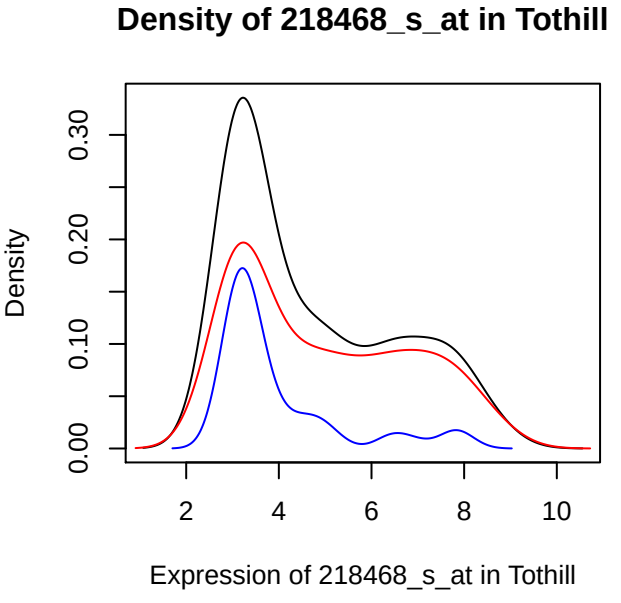

Density of 214587_at in TCGA
Density
0.0
0.5
1.0
1.5
2
3
4
5
6
7
Expression of 214587_at in TCGA

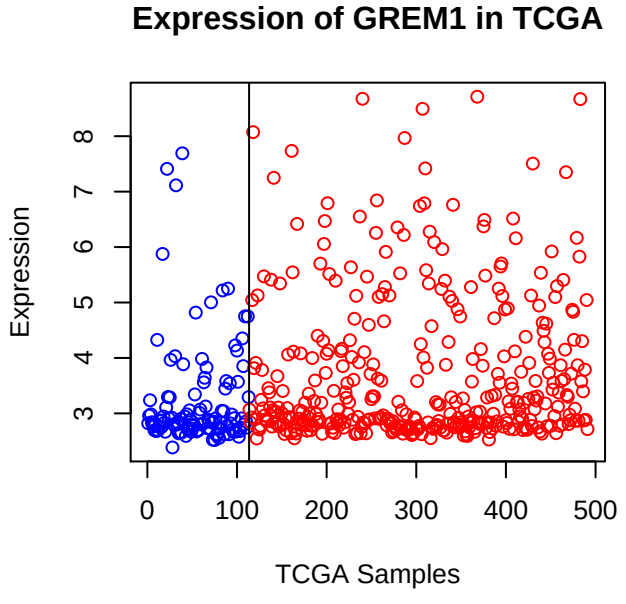

Density of 214587_at in Tothill
Density
0.0
0.1
0.2
0.3
0.4
2
4
6
8
Expression of 214587_at in Tothill

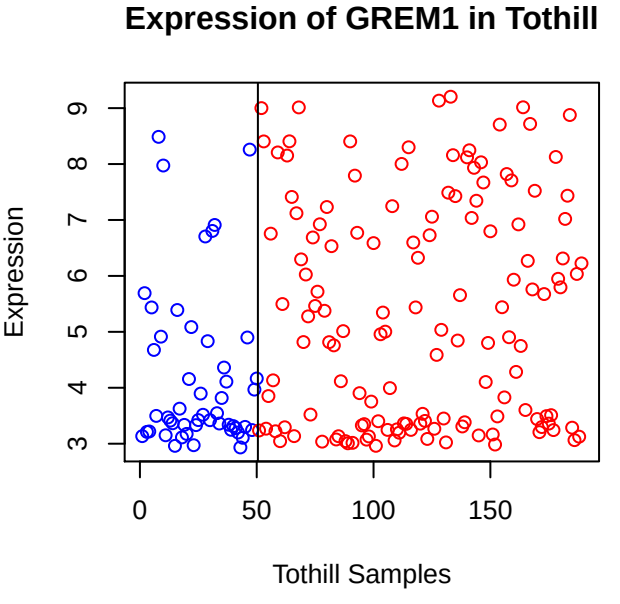

Expression of GREM1 in TCGA
Expression
3
4
5
6
7
8
9
0
100
200
300
400
500
TCGA Samples

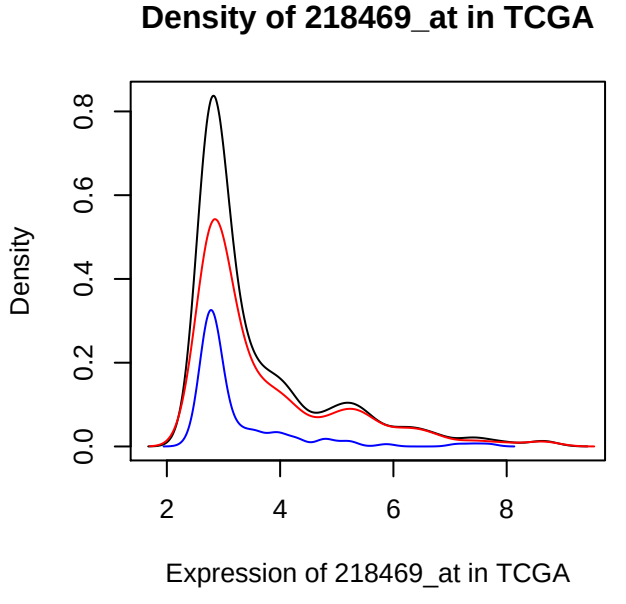

Expression of GREM1 in Tothill
Expression
3
4
5
6
7
8
9
0
50
100
150
Tothill Samples

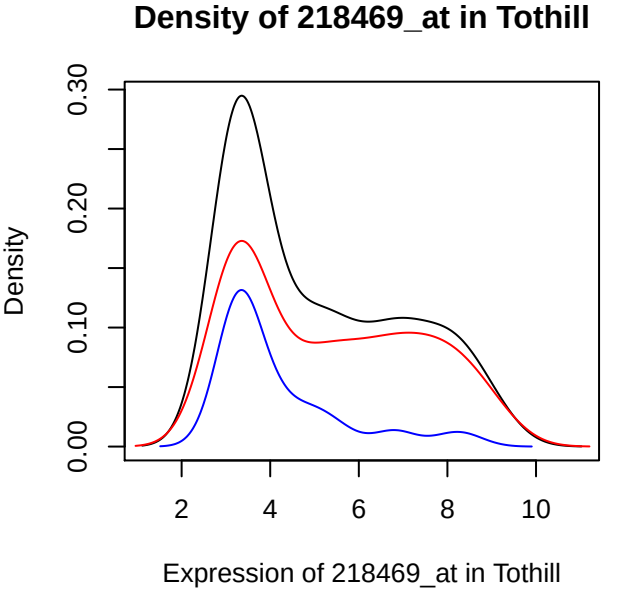

Density of 218468_s_at in TCGA
Density
0.0
0.2
0.4
0.6
0.8
2
4
6
8
Expression of 218468_s_at in TCGA

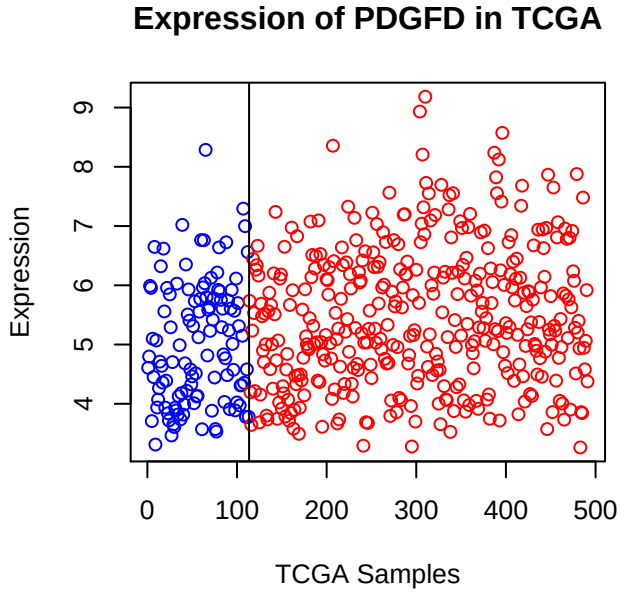

Density of 218468_s_at in Tothill
Density
0.00
0.10
0.20
0.30
2
4
6
8
10
Expression of 218468_s_at in Tothill

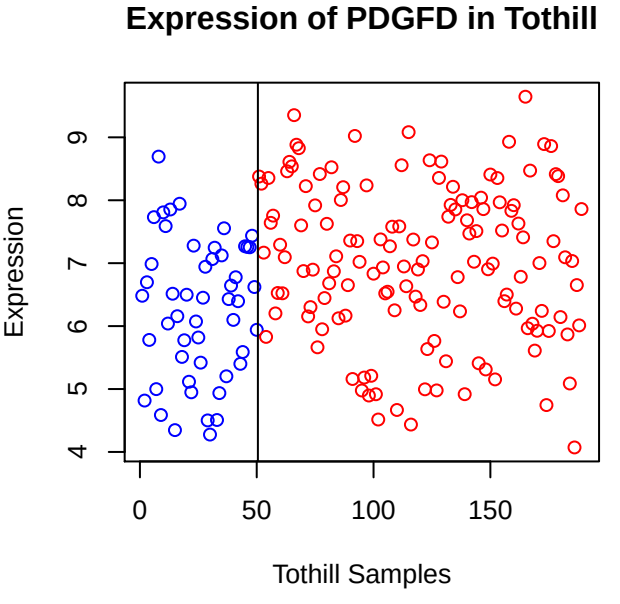

Expression of GREM1 in TCGA
Expression
3
4
5
6
7
8
0
100
200
300
400
500
TCGA Samples

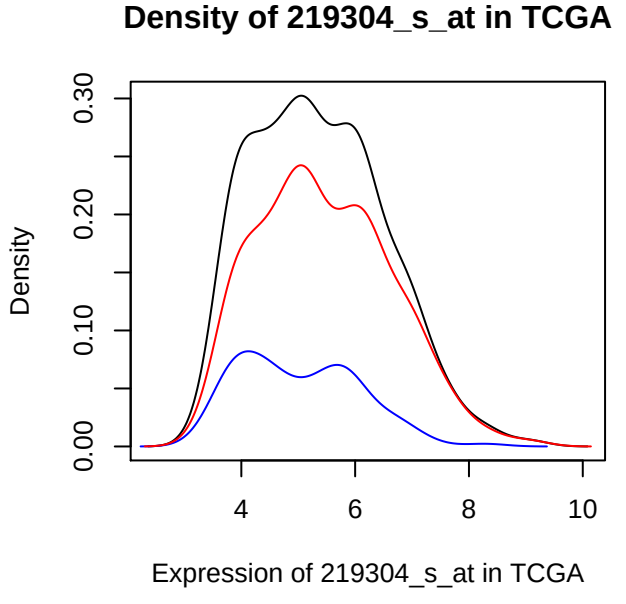

Expression of GREM1 in Tothill
Expression
3
4
5
6
7
8
9
0
50
100
150
Tothill Samples

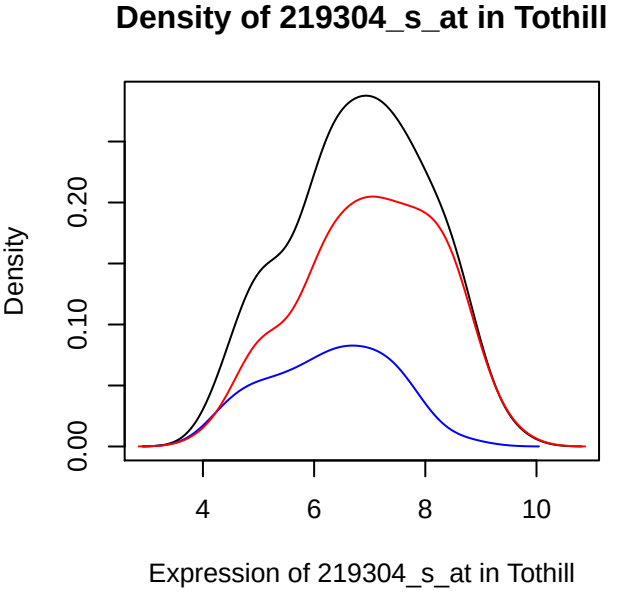

Density of 218469_at in TCGA
Density
0.0
0.2
0.4
0.6
0.8
2
4
6
8
Expression of 218469_at in TCGA

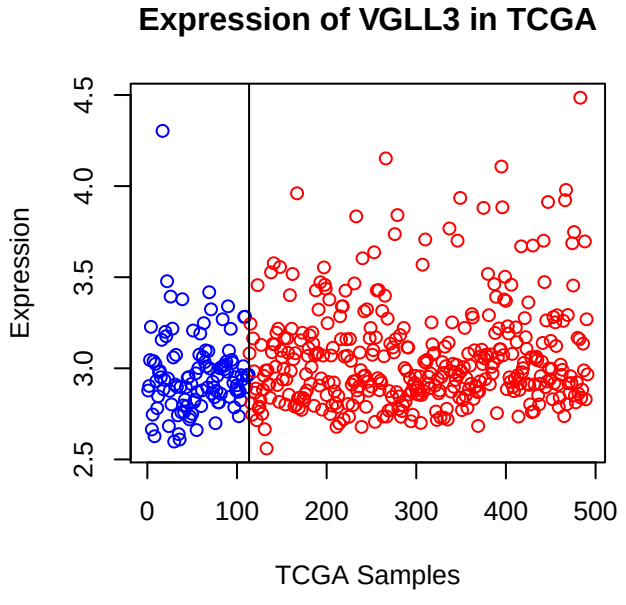

Density of 218469_at in Tothill
Density
0.00
0.10
0.20
0.30
2
4
6
8
10
Expression of 218469_at in Tothill

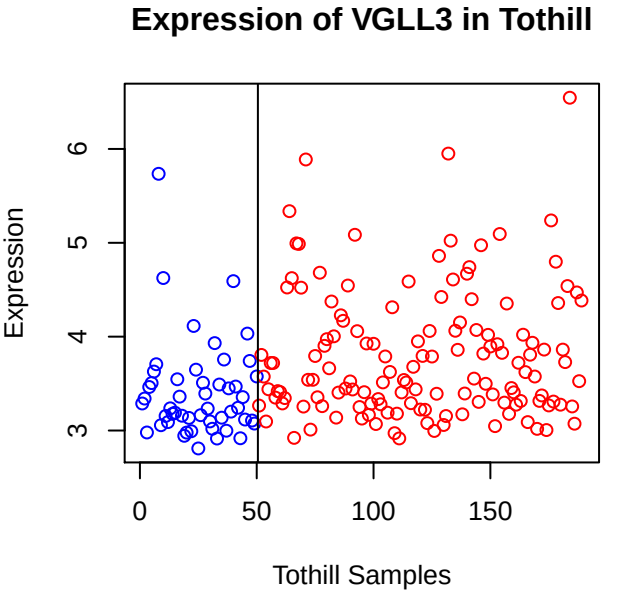

Expression of PDGFD in TCGA
Expression
4
5
6
7
8
9
0
100
200
300
400
500
TCGA Samples

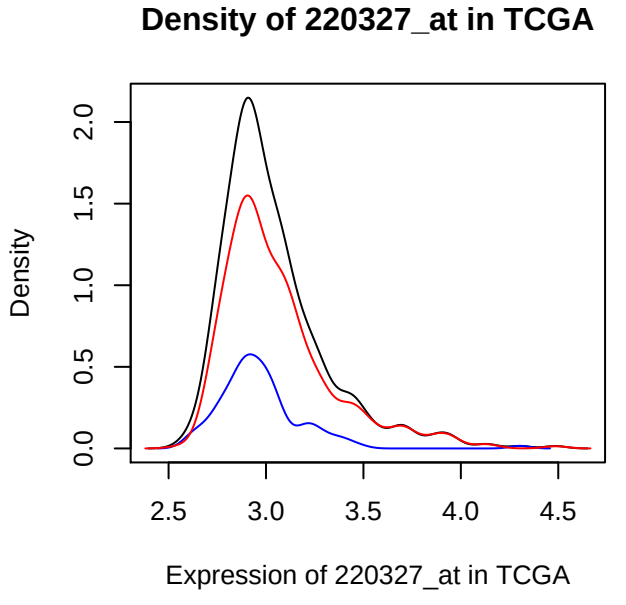

Expression of PDGFD in Tothill
Expression
4
5
6
7
8
9
0
50
100
150
Tothill Samples

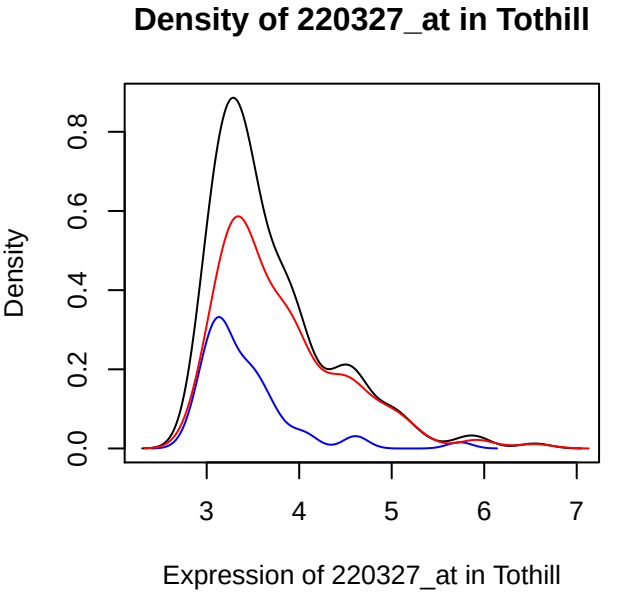

Density of 219304_s_at in TCGA
Density
0.00
0.10
0.20
0.30
4
6
8
10
Expression of 219304_s_at in TCGA

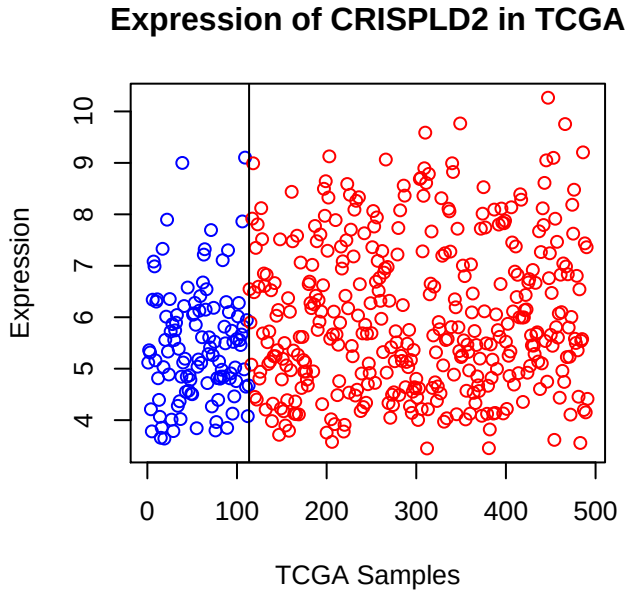

Density of 219304_s_at in Tothill
Density
0.00
0.10
0.20
4
6
8
10
Expression of 219304_s_at in Tothill

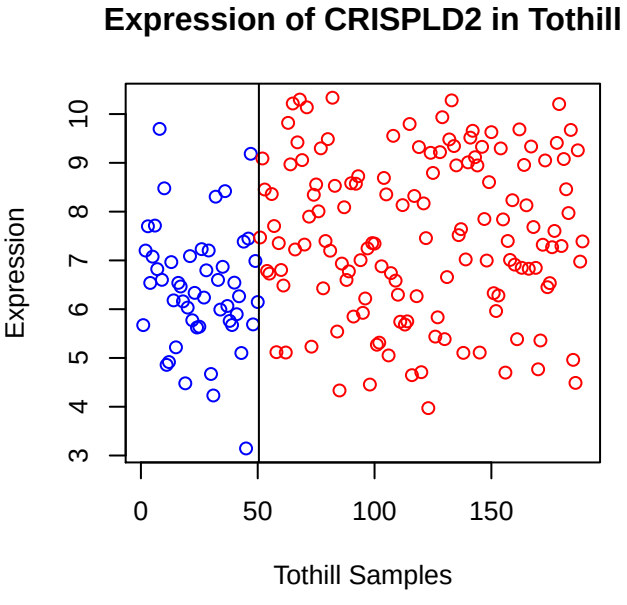

Expression of VGLL3 in TCGA
Expression
2.5
3.0
3.5
4.0
4.5
0
100
200
300
400
500
TCGA Samples

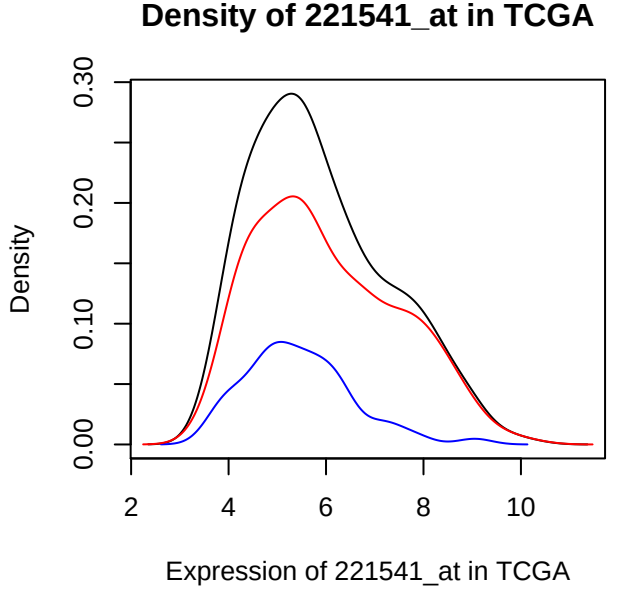

Expression of VGLL3 in Tothill
Expression
3
4
5
6
0
50
100
150
Tothill Samples

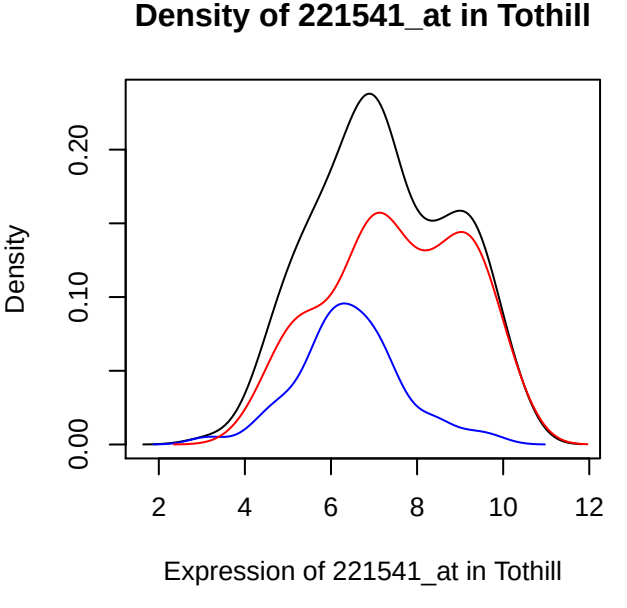

Density of 220327_at in TCGA
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
Expression of 220327_at in TCGA

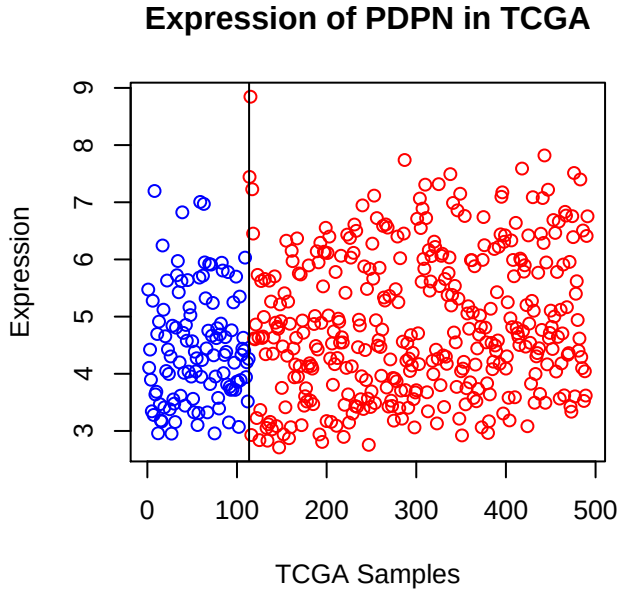

Density of 220327_at in Tothill
Density
0.0
0.2
0.4
0.6
0.8
3
4
5
6
7
Expression of 220327_at in Tothill

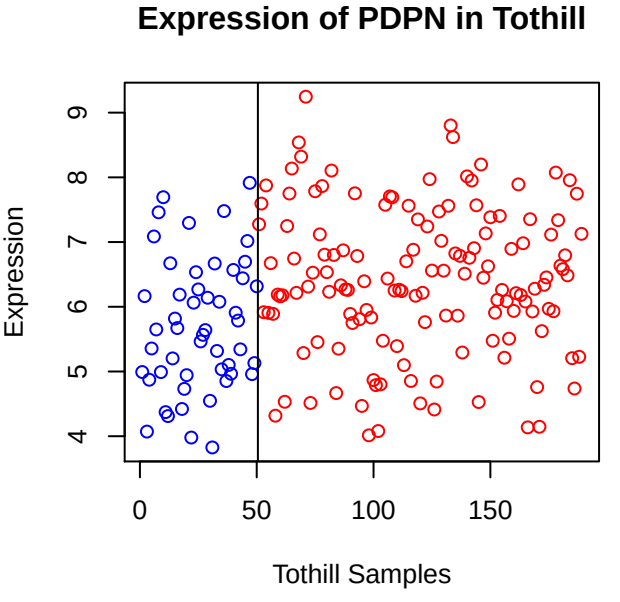

Expression of CRISPLD2 in TCGA
Expression
TCGA Samples
0
100
200
300
400
500
4
5
6
7
8
9
10

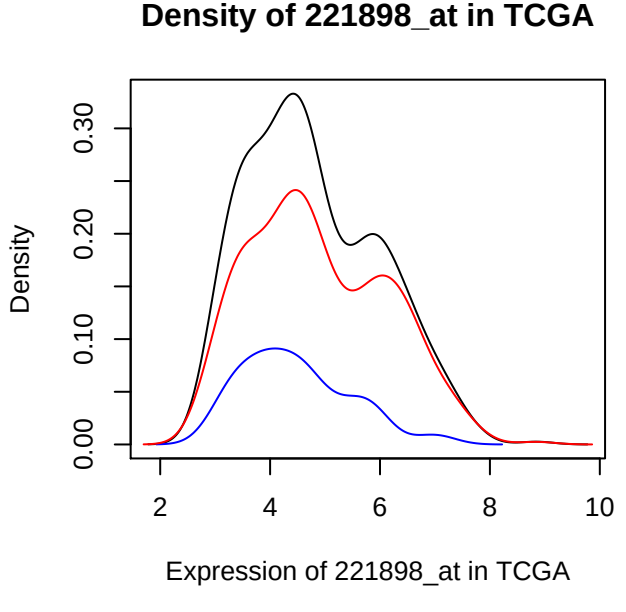

Expression of CRISPLD2 in Tothill
Expression
Tothill Samples
0
50
100
150
3
4
5
6
7
8
9
10

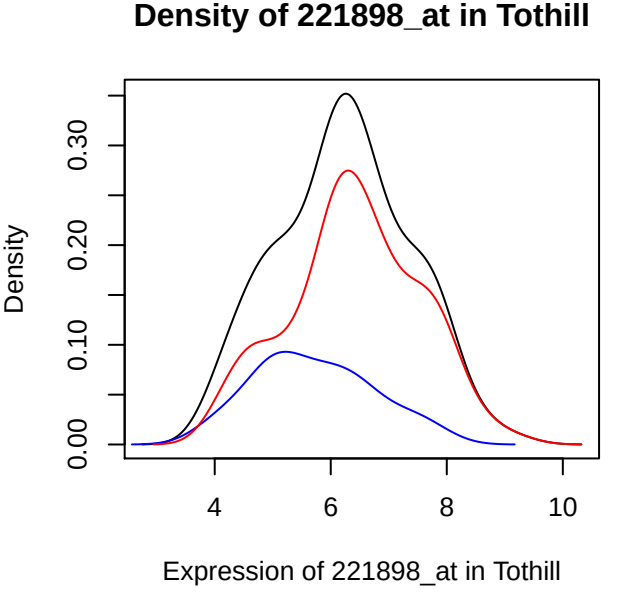

Density of 221541_at in TCGA
Density
Expression of 221541_at in TCGA
0.00
0.10
0.20
0.30
2
4
6
8
10

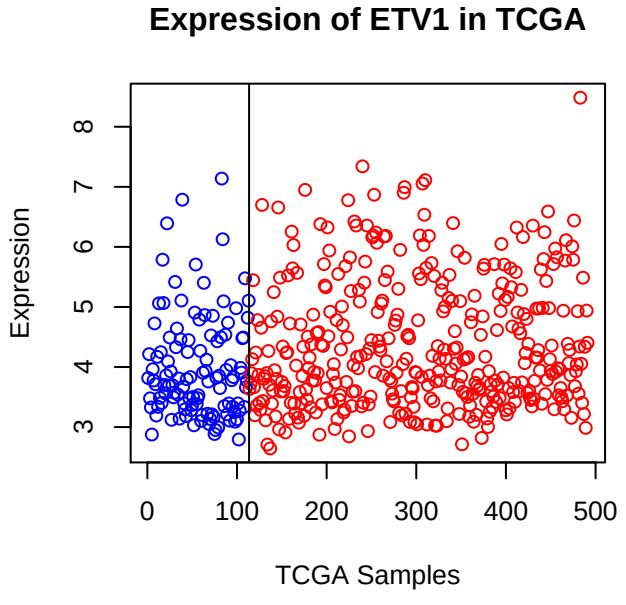

Density of 221541_at in Tothill
Density
Expression of 221541_at in Tothill
0.00
0.10
0.20
2
4
6
8
10
12

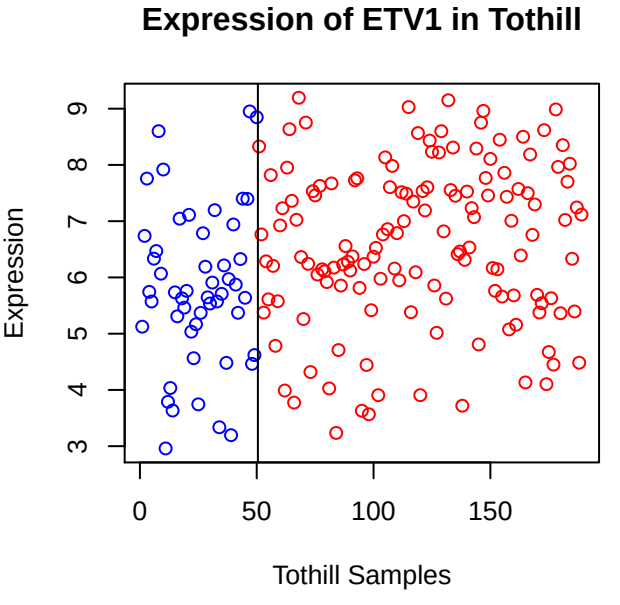

Expression of PDPN in TCGA
Expression
3
4
5
6
7
8
9
0
100
200
300
400
500
TCGA Samples

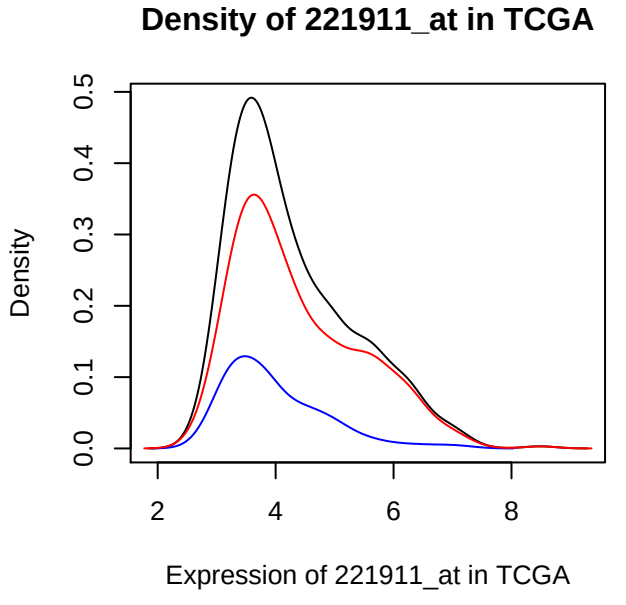

Expression of PDPN in Tothill
Expression
4
5
6
7
8
9
0
50
100
150
Tothill Samples

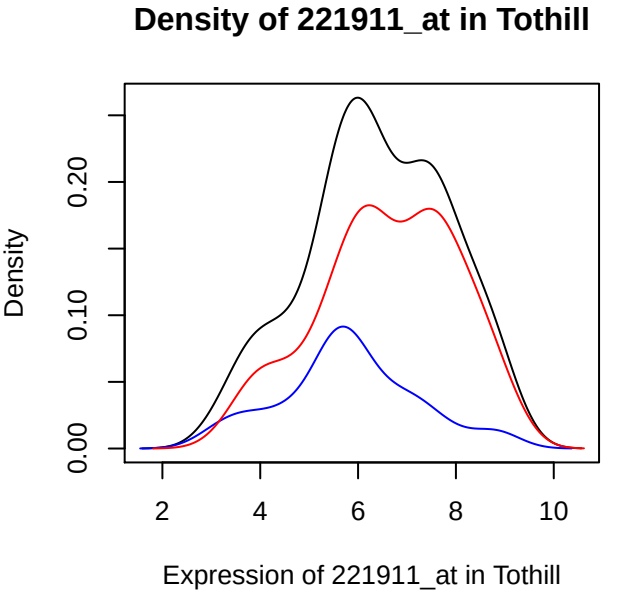

Density of 221898_at in TCGA
Density
0.00
0.10
0.20
0.30
2
4
6
8
10
Expression of 221898_at in TCGA

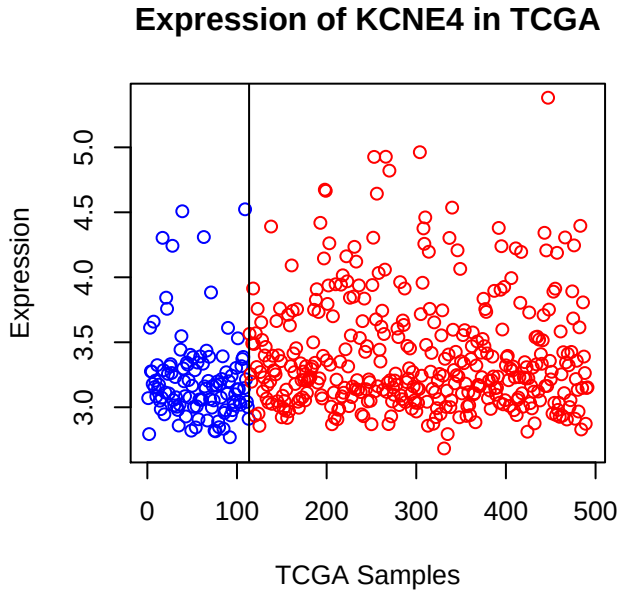

Density of 221898_at in Tothill
Density
0.00
0.10
0.20
0.30
4
6
8
10
Expression of 221898_at in Tothill

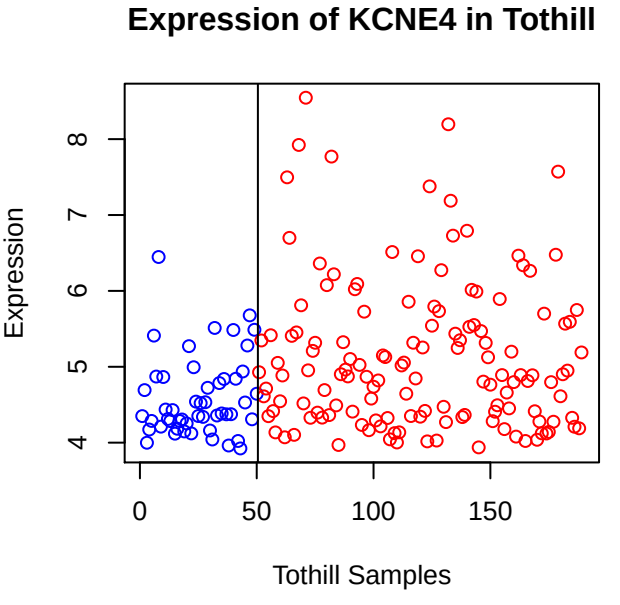

Expression of ETV1 in TCGA
Expression
3
4
5
6
7
8
0
100
200
300
400
500
TCGA Samples

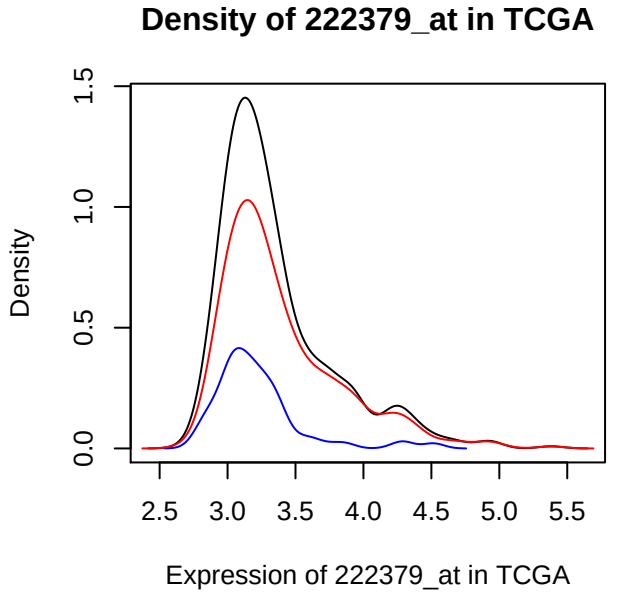

Expression of ETV1 in Tothill
Expression
3
4
5
6
7
8
9
0
50
100
150
Tothill Samples

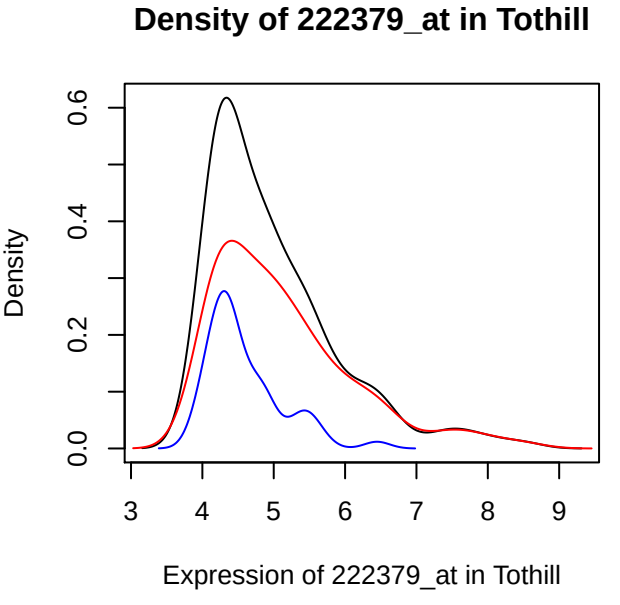

Density of 221911_at in TCGA
Density
0.0
0.1
0.2
0.3
0.4
0.5
2
4
6
8
Expression of 221911_at in TCGA

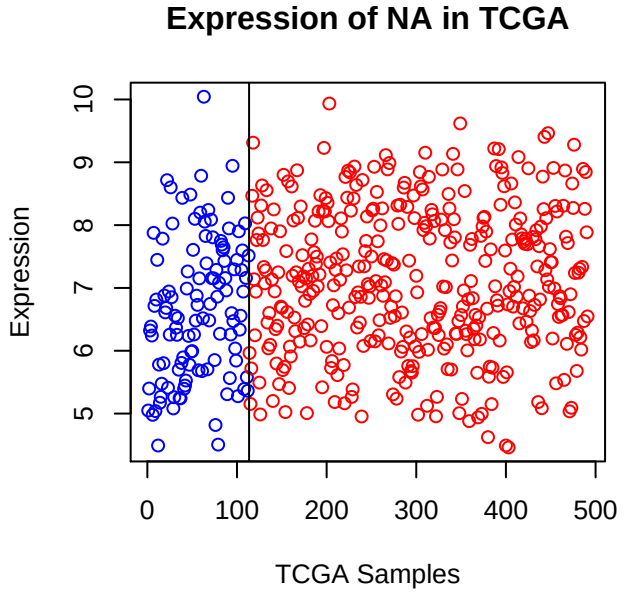

Density of 221911_at in Tothill
Density
0.00
0.10
0.20
2
4
6
8
10
Expression of 221911_at in Tothill

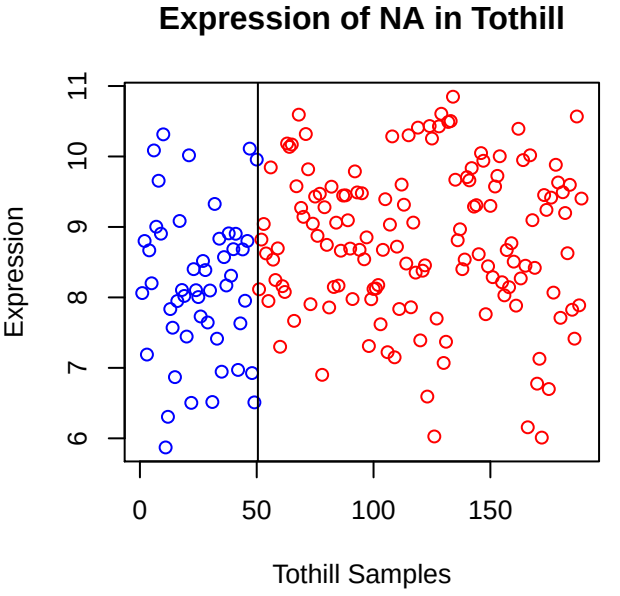

Expression of KCNE4 in TCGA
Expression
3.0
3.5
4.0
4.5
5.0
0
100
200
300
400
500
TCGA Samples

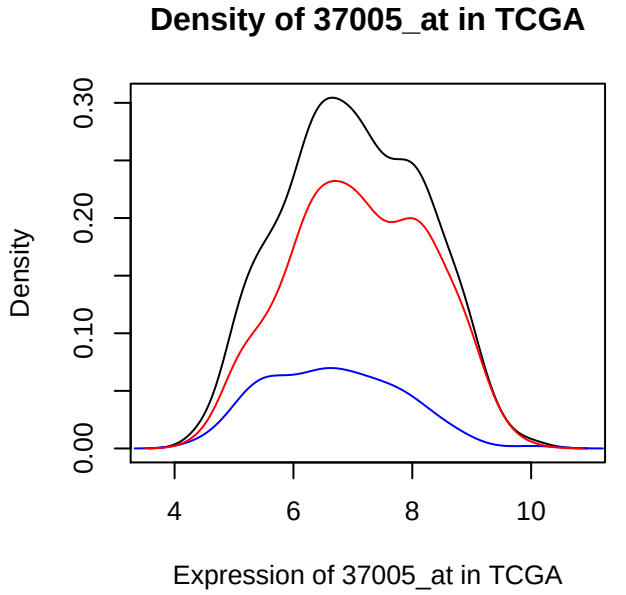

Expression of KCNE4 in Tothill
Expression
4
5
6
7
8
0
50
100
150
Tothill Samples

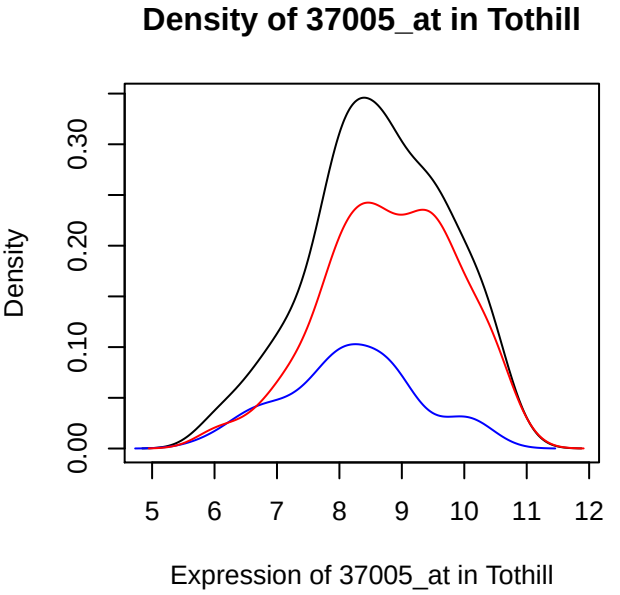

Density of 222379_at in TCGA
Density
0.0
0.5
1.0
1.5
2.5
3.0
3.5
4.0
4.5
5.0
5.5
Expression of 222379_at in TCGA

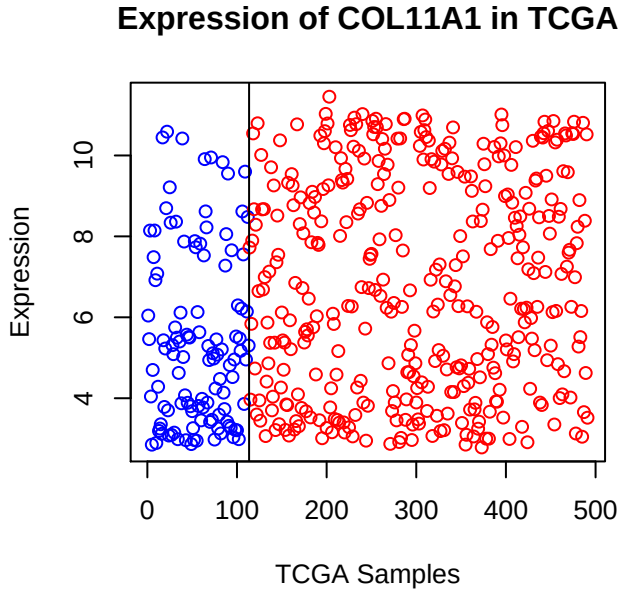

Density of 222379_at in Tothill
Density
0.0
0.2
0.4
0.6
3
4
5
6
7
8
9
Expression of 222379_at in Tothill

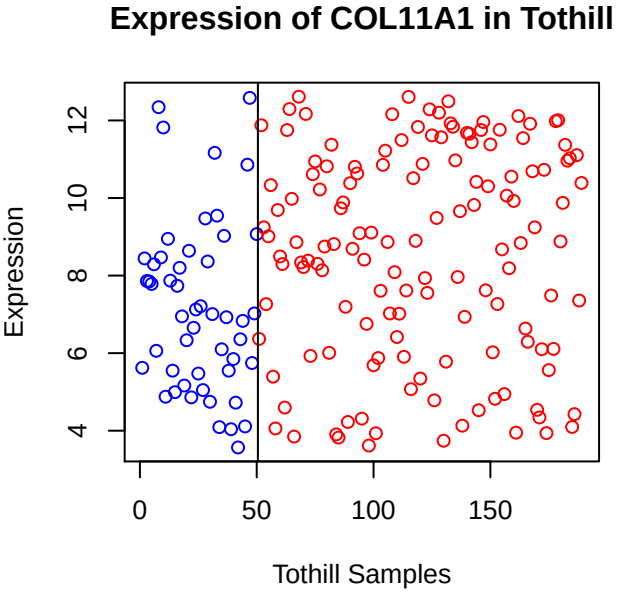

Expression of NA in TCGA
Expression
TCGA Samples
0
100
200
300
400
500
5
6
7
8
9
10

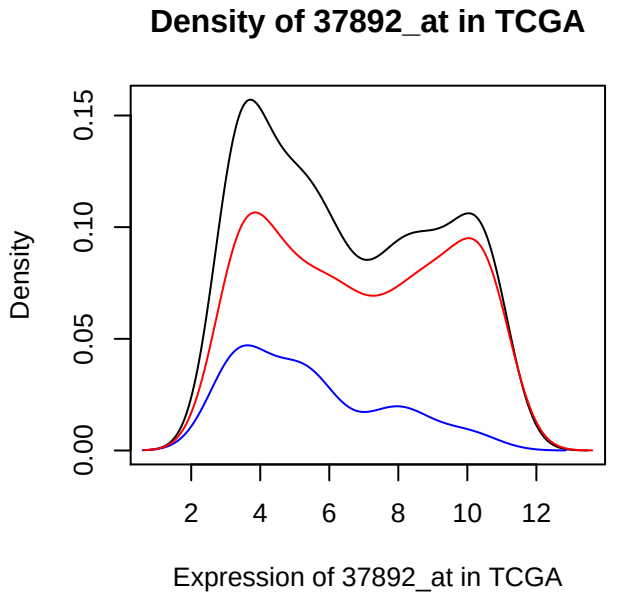

Expression of NA in Tothill
Expression
Tothill Samples
0
50
100
150
6
7
8
9
10
11

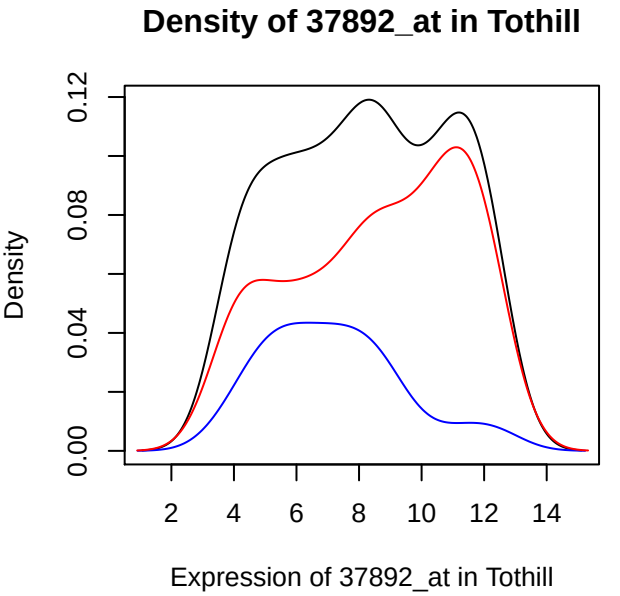

Density of 37005_at in TCGA
Density
Expression of 37005_at in TCGA
4
6
8
10
0.00
0.10
0.20
0.30


Density of 37005_at in Tothill
Density
Expression of 37005_at in Tothill
5
6
7
8
9
10
11
12
0.00
0.10
0.20
0.30


Expression of COL11A1 in TCGA
Expression
4
6
8
10
0
100
200
300
400
500
TCGA Samples


Expression of COL11A1 in Tothill
Expression
4
6
8
10
12
0
50
100
150
Tothill Samples


Density of 37892_at in TCGA
Density
0.00
0.05
0.10
0.15
2
4
6
8
10
12
Expression of 37892_at in TCGA


Density of 37892_at in Tothill
Density
0.00
0.04
0.08
0.12
2
4
6
8
10
12
14
Expression of 37892_at in Tothill

Plotting FABP4 vs ADH1B for Microarray Datasets

by Keith Baggerly

Jun 23, 2013

1 Executive Summary

1.1 Introduction

We want to examine the joint distribution of FABP4 and ADH1B in the TCGA, Tothill, Bonome, and CCLE cohorts, and to assess how much RD rates increase with expression levels.

1.2 Data and Methods

We use the array and clinical data prepared in the various “assembleData” and “assembleClinical” scripts. We produce plots for each dataset, using high/low cutoffs assessed by eye. We measure the RD rates when both genes are high for the TCGA and Tothill cohorts.

1.3 Results

We produce figures named fabp4VsAdh1b(dataset).

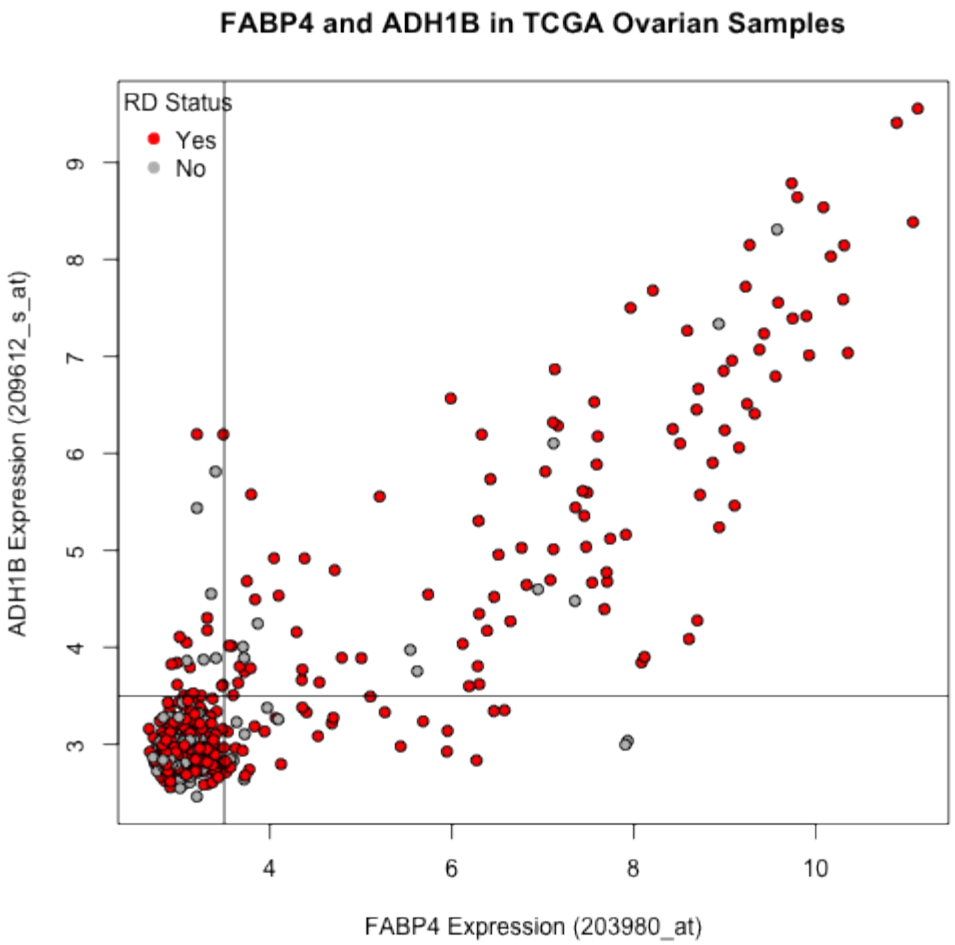
The TCGA RD rates are 97/107 in the high expression group, vs 281/384 in the complement.

The Tothill RD rates are 59/63 in the high expression group, vs 80/126 in the complement.

2. Plotting FABP4 vs ADH1B for TCGA

```
load(file.path("RDataObjects", "tcgaExpression.RData"))
load(file.path("RDataObjects", "tcgaFilteredSamples.RData"))
load(file.path("RDataObjects", "tcgaClinical.RData"))
```

```
fabp4.ps <- "203980_at"
adh1b.ps <- "209612_s_at"
bgColors <- rep("grey", length(tcgaSampleRD))
bgColors[tcgaSampleRD == "RD"] <- "red"
tcgaUsed <- tcgaFilteredSamples[, "sampleUse"] == "Used"
plot(tcgaExpression[fabp4.ps, tcgaUsed], tcgaExpression[adh1b.ps, tcgaUsed],
    pch = 21, bg = bgColors[tcgaUsed], xlab = "FABP4 Expression (203980_at)",
    ylab = "ADH1B Expression (209612_s_at)", main = "FABP4 and ADH1B in TCGA Ovarian Samples")
abline(v = 3.5, h = 3.5)
legend("topleft", c("Yes", "No"), pch = 19, col = c("red", "grey"), bty = "n",
    title = "RD Status")
```




```
## pdf
##   2
```

Now we check the RD to No RD ratios by subgroup.

```
table(tcgaSampleRD[tcgaUsed], tcgaExpression[fabp4.ps, tcgaUsed] > 3.5)
```

```
##
##         FALSE TRUE
##   No RD    94   19
##   RD      254  124
```

```
table(tcgaSampleRD[tcgaUsed], tcgaExpression[adh1b.ps, tcgaUsed] > 3.5)
```

```
##
##         FALSE TRUE
##   No RD    97   16
##   RD      266  112
```

```
table(tcgaSampleRD[tcgaUsed], (tcgaExpression[fabp4.ps, tcgaUsed] > 3.5) &
(tcgaExpression[adh1b.ps,
    tcgaUsed] > 3.5))
```

```
##
##         FALSE TRUE
##   No RD   103   10
```

```
##   RD      281   97
```

When both gene levels are high (above 3.5), the RD rate is 97/107 (90.6%), as opposed to 281/384 (73.2%).

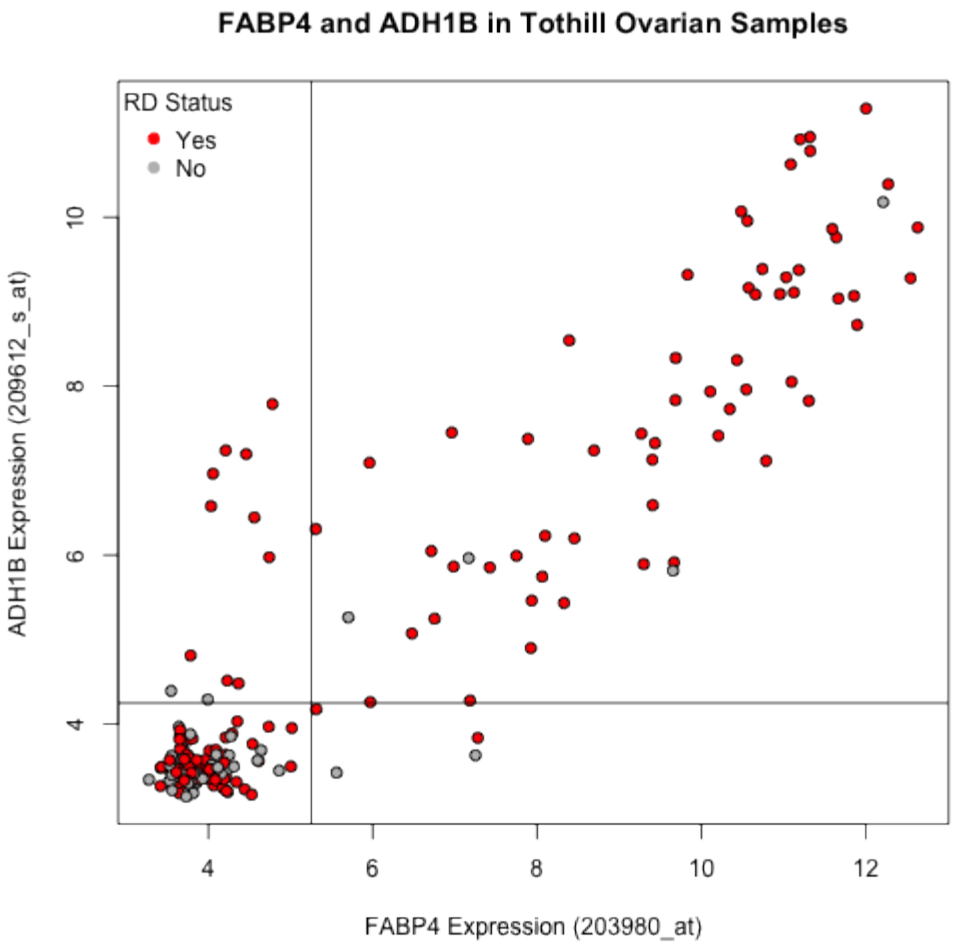
```
rm(tcgaExpression, tcgaFilteredSamples, tcgaSampleInfo, fabp4.ps, adh1b.ps,
    bgColors, tcgaDataDirs, tcgaFiles, tcgaSampleClinicalMapping, tcgaSampleRD,
    tcgaUsed)
```

3. Plotting FABP4 vs ADH1B for Tothill

```
load(file.path("RDataObjects", "tothillExpression.RData"))
load(file.path("RDataObjects", "tothillFilteredSamples.RData"))
load(file.path("RDataObjects", "tothillClinical.RData"))

tothillExpression <- tothillExpression[, rownames(tothillFilteredSamples)]
all(rownames(tothillFilteredSamples) == names(tothillRD))
```

```
## [1] TRUE
```

```
fabp4.ps <- "203980_at"
adh1b.ps <- "209612_s_at"
bgColors <- rep("grey", length(tothillRD))
bgColors[tothillRD == "RD"] <- "red"
tothillUsed <- tothillFilteredSamples[, "sampleUse"] == "Used"
plot(tothillExpression[fabp4.ps, tothillUsed], tothillExpression[adh1b.ps, tothillUsed],
    pch = 21, bg = bgColors[tothillUsed], xlab = "FABP4 Expression (203980_at)",
    ylab = "ADH1B Expression (209612_s_at)", main = "FABP4 and ADH1B in Tothill Ovarian Samples")
abline(v = 5.25, h = 4.25)
legend("topleft", c("Yes", "No"), pch = 19, col = c("red", "grey"), bty = "n",
    title = "RD Status")
```




```
## pdf
##   2
```

Now we check the RD to No RD ratios by subgroup.

```
table(tothillRD[tothillUsed], tothillExpression[fabp4.ps, tothillUsed] > 5.25)
```

```
##
##         FALSE TRUE
##   No RD    44    6
##   RD       78   61
```

```
table(tothillRD[tothillUsed], tothillExpression[adh1b.ps, tothillUsed] > 4.25)
```

```
##
##         FALSE TRUE
##   No RD    44    6
##   RD       70   69
```

```
table(tothillRD[tothillUsed], (tothillExpression[fabp4.ps, tothillUsed] > 5.25) &
    (tothillExpression[adh1b.ps, tothillUsed] > 4.25))
```

```
##
##         FALSE TRUE
##   No RD    46    4
##   RD       80   59
```

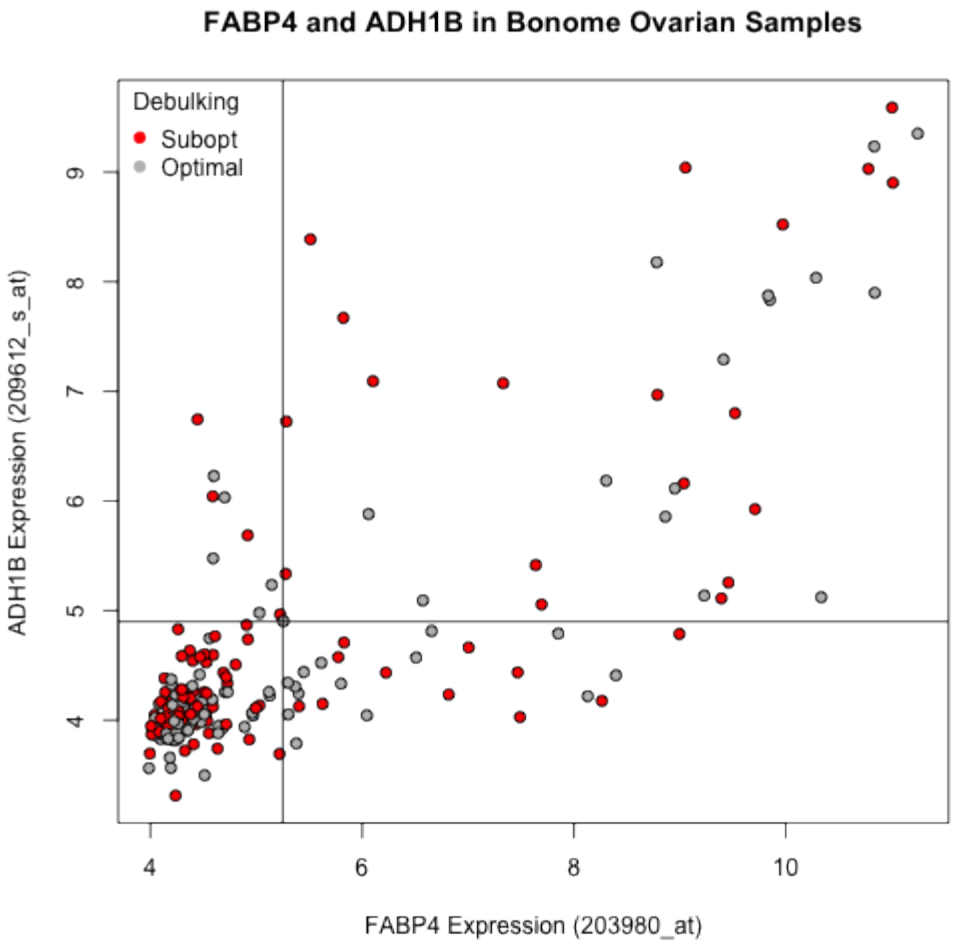
When both gene levels are high, the RD rate is 59/63 (93.7%), as opposed to 80/126 (63.5%).

```
rm(tothillClinical, tothillExpression, tothillFilteredSamples, tothillOSMos,
    tothillPFSMos, fabp4.ps, adh1b.ps, bgColors, tothillRD, tothillUsed)
```

4. Plotting FABP4 vs ADH1B for Bonome

```
load(file.path("RDataObjects", "bonomeExpression.RData"))
load(file.path("RDataObjects", "bonomeClinical.RData"))

bonomeExpression <- bonomeExpression[, rownames(bonomeClinical)]
```

```
fabp4.ps <- "203980_at"
adh1b.ps <- "209612_s_at"
bgColors <- rep("grey", nrow(bonomeClinical))
bgColors[bonomeClinical[, "SurgeryOutcome"] == "Suboptimal"] <- "red"
bonomeUsed <- bonomeClinical[, "SurgeryOutcome"] != ""  ## omit 10 normal samples
plot(bonomeExpression[fabp4.ps, bonomeUsed], bonomeExpression[adh1b.ps, bonomeUsed],
    pch = 21, bg = bgColors[bonomeUsed], xlab = "FABP4 Expression (203980_at)",
    ylab = "ADH1B Expression (209612_s_at)", main = "FABP4 and ADH1B in Bonome Ovarian Samples")
abline(v = 5.25, h = 4.9)
legend("topleft", c("Subopt", "Optimal"), pch = 19, col = c("red", "grey"),
    bty = "n", title = "Debulking")
```




```
## pdf
##   2
```

Now we check the Optimal to Suboptimal ratios by subgroup.

```
table(bonomeClinical[bonomeUsed, "SurgeryOutcome"], bonomeExpression[fabp4.ps,
    bonomeUsed] > 5.25)
```

```
##
##              FALSE TRUE
##                  0    0
##   Optimal       60   30
##   Suboptimal    65   30
```

```
table(bonomeClinical[bonomeUsed, "SurgeryOutcome"], bonomeExpression[adh1b.ps,
    bonomeUsed] > 4.9)
```

```
##
##              FALSE TRUE
##                  0    0
##   Optimal       69   21
##   Suboptimal    72   23
```

```
table(bonomeClinical[bonomeUsed, "SurgeryOutcome"], (bonomeExpression[fabp4.ps,
    bonomeUsed] > 5.25) & (bonomeExpression[adh1b.ps, bonomeUsed] > 4.9))
```

```
##
##              FALSE TRUE
##                  0    0
##   Optimal       74   16
##   Suboptimal    76   19
```

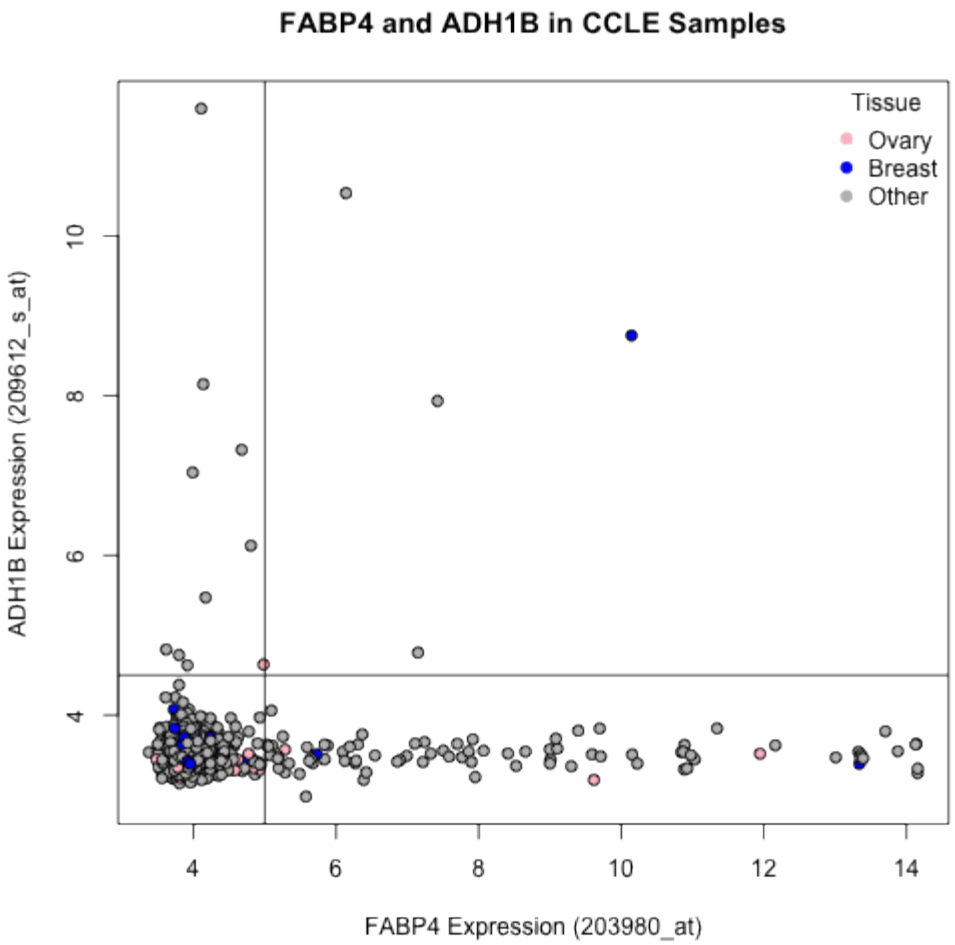
```
rm(bonomeClinical, bonomeExpression, fabp4.ps, adh1b.ps, bgColors, bonomeOSYrs,
    bonomeUsed)
```

5. Plotting FABP4 vs ADH1B for CCLE

```
load(file.path("RDataObjects", "ccleExpression.RData"))
load(file.path("RDataObjects", "ccleClinical.RData"))

all(rownames(ccleClinical) == colnames(ccleExpression))
```

```
## [1] TRUE
```

```
fabp4.ps <- "203980_at"
adh1b.ps <- "209612_s_at"
bgColors <- rep("grey", nrow(ccleClinical))
bgColors[ccleClinical[, "primarySite"] == "ovary"] <- "pink"
bgColors[ccleClinical[, "primarySite"] == "breast"] <- "blue"
plot(ccleExpression[fabp4.ps, ], ccleExpression[adh1b.ps, ], pch = 21, bg = bgColors,
    xlab = "FABP4 Expression (203980_at)", ylab = "ADH1B Expression (209612_s_at)",
    main = "FABP4 and ADH1B in CCLE Samples")
abline(v = 5, h = 4.5)
legend("topright", c("Ovary", "Breast", "Other"), pch = 19, col = c("pink",
    "blue", "grey"), bty = "n", title = "Tissue")
```




```
## pdf
##   2
```

```
rm(ccleClinical, ccleExpression, fabp4.ps, adh1b.ps, bgColors)
```

Appendix

```
getwd()
```

```
## [1] "/Users/kabagg/TCGA/RDPaper"
```

```
sessionInfo()
```

```
## R version 3.0.0 (2013-04-03)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] knitr_1.1
##
## loaded via a namespace (and not attached):
```

```
## [1] digest_0.6.3   evaluate_0.4.3 formatR_0.7    stringr_0.6.2
## [5] tools_3.0.0
```

Residual Disease Paper

Exploring Default PCR Problems

by Shelley Herbrich

May 13, 2013

1 Executive Summary

1.1 Introduction

Currently, PCR vendor software internally chooses a default threshold for each target gene on a plate (in this case, 18S, FABP4, and ADH1B). On revisiting the data obtained from Washington, we noticed that for a few of the plates the threshold chosen for FABP4 is quite low. The problem is that in going from one plate to another, these cutoffs are largely stable for 18S and ADH1B, but not for FABP4.

We suspect that the cutoffs for FABP4 may be being dragged out of place by some outlier wells, as might be the case with one or two problematic samples.

1.2 Methods

For this analysis, we focus primarily on the first plate run on Plate 2. Some of the samples were flagged as problematic (and were later rerun), but the threshold for FABP4 was applied to all samples, including those that were retained.

We suspect a better threshold for FABP4 would be near the same level as those for ADH1B and 18S. We interpolate the revised Ct values based on this proposed threshold.

1.3 Results

It appears that the software chooses the gene-specific threshold based on the first encountered curve for that particular target. This results in significantly underestimated thresholds when there are poor quality wells on a given plate. The implications of this change aren't subtle. The Ct values we would infer for FABP4 with both thresholds are different, and the mean difference is about 5.5 units (or about $2^{5.5}$ = 46-fold).

Therefore, we need to develop an alternative quantification method for our PCR data that is independent of the software-defined threshold values.

3 Assessing the Impact of Incorrect Thresholds

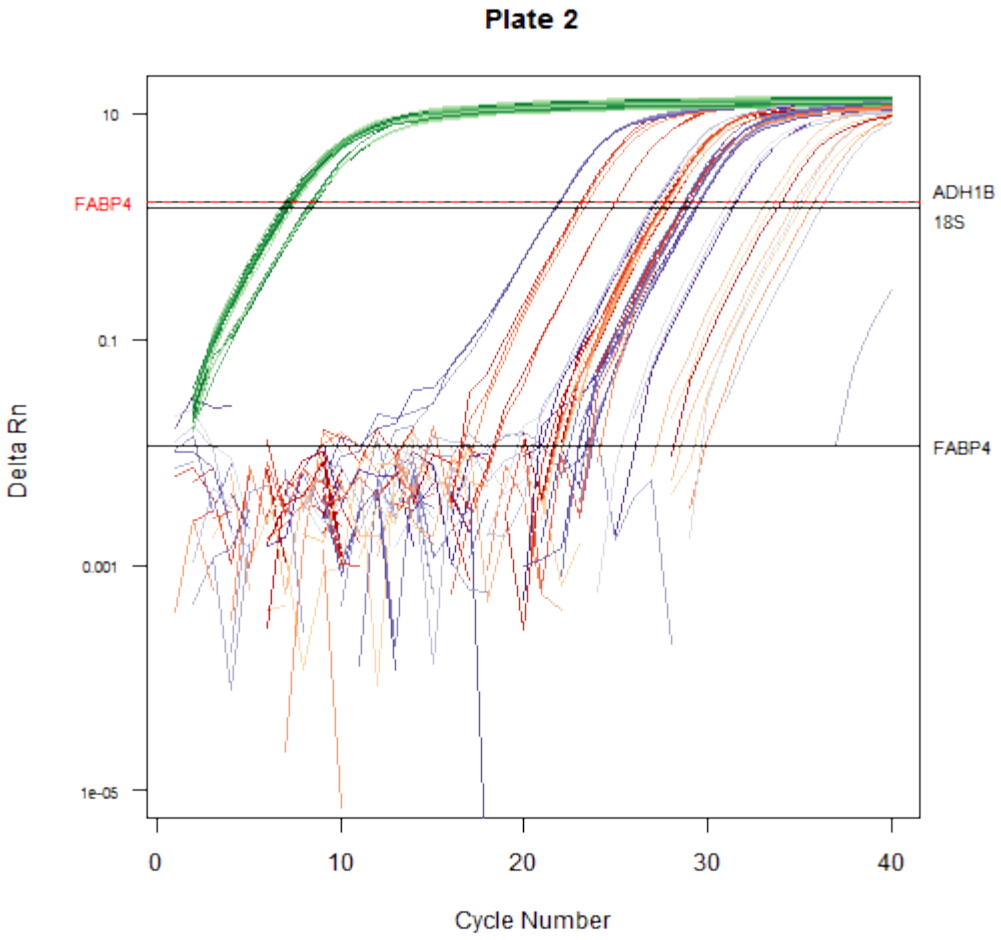
```
library(RColorBrewer)
```

```
load(file.path("RDataObjects", "rawPCRWash_Unfiltered.RData"))
load(file.path("RDataObjects", "PCRResults.RData"))
```

```
# extract annotation info
rawPCRPlate2 <- rawPCRWash[which(rawPCRWash$Plate == "Plate.2"), ]
wells <- rawPCRPlate2$Well
genes <- rawPCRPlate2$Target.Name
thresholds <- unique(rawPCRPlate2$Ct.Threshold)
names(thresholds) <- unique(genes)
```

```
colors <- array(NA, length(genes))
colors[which(genes == "ADH1B")] <- rep(brewer.pal(9, "OrRd")[3:8], 4)[1:23]
colors[which(genes == "FABP4")] <- rep(brewer.pal(9, "Purples")[3:8], 4)[1:23]
colors[which(genes == "18S")] <- rep(brewer.pal(9, "Greens")[3:8], 4)[1:23]
```

```
par(mar = c(5.1, 5.1, 4.1, 3.1))
plot(NA, xlim = c(1, 40), ylim = c(-5, 1.1), xlab = "Cycle Number", ylab = "",
    yaxt = "n", main = "Plate 2")
axis(2, at = seq(-5, 2, by = 2), labels = 10^seq(-5, 2, by = 2), las = 2, cex.axis = 0.7)
mtext("Delta Rn", side = 2, at = -2, line = 4, las = 0, cex = 1)

for (i1 in 1:length(wells)) {
    wn <- wells[i1]
    tempRn <- log10(rawPCRPlate2[which(rawPCRPlate2$Well == wn), grep("Cycle",
        colnames(rawPCRPlate2))])
    missing <- is.na(tempRn)
    points(c(1:40), tempRn, col = colors[i1], type = "l")
}

# add thresholds
abline(h = log10(thresholds))
mtext(names(thresholds), side = 4, at = log10(thresholds) + c(0.1, 0, -0.1),
    line = 0.5, las = 2, cex = 0.75)

# add proposed threshold
otherFABP4 <- c(1.674, 1.674, 1.631, 1.589)
estimatedThreshold <- mean(otherFABP4)
abline(h = log10(estimatedThreshold), col = "red", lty = 2)
mtext("FABP4", side = 2, at = log10(estimatedThreshold), line = 0.5, las = 2,
    cex = 0.75, col = "red")
```



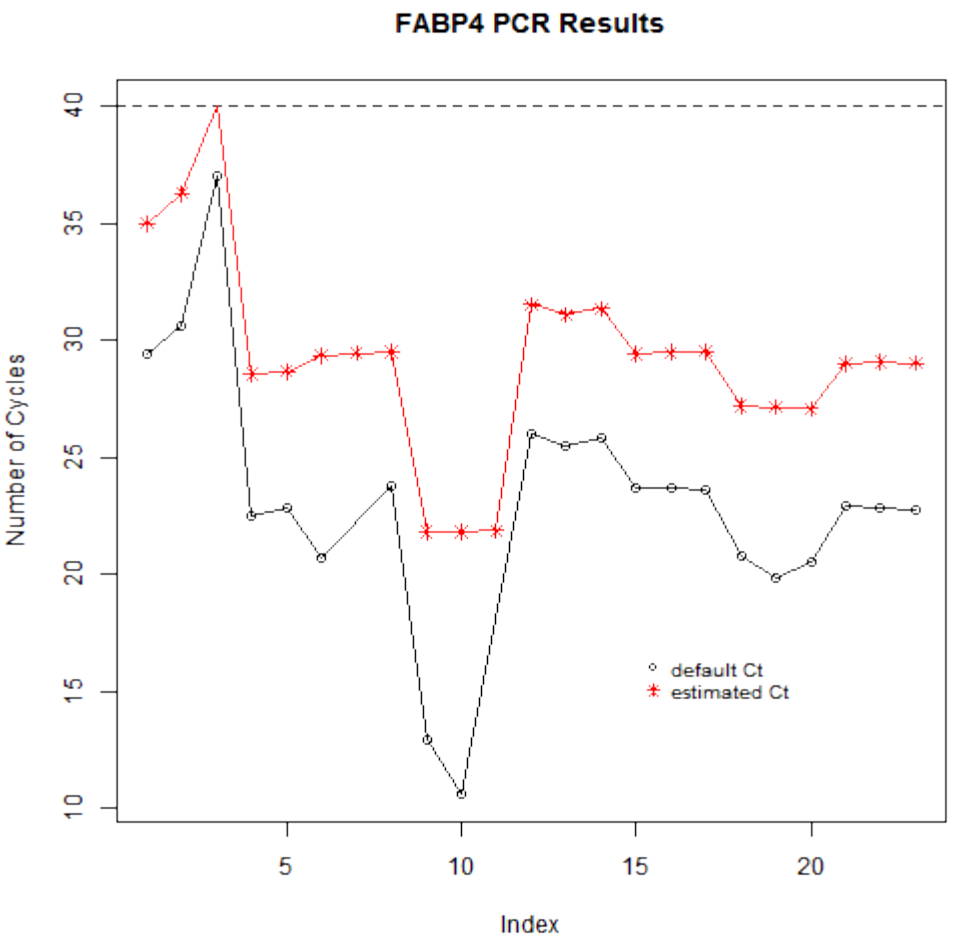

```
goi <- "FABP4"
fabp4Wells <- wells[which(genes == goi)]
defaultCt <- as.numeric(rawPCRPlate2$Ct[which(genes == goi)])
mean(defaultCt, na.rm = TRUE)
```

```
## [1] 23.26
```

```
estimatedCt <- array(NA, length(fabp4Wells))
for (i1 in 1:length(fabp4Wells)) {
    wellID <- fabp4Wells[i1]
    subRn <- as.numeric(rawPCRPlate2[which(rawPCRPlate2$Well == wellID), grep("Cycle",
        colnames(rawPCRPlate2))])
    ub <- which(subRn > estimatedThreshold)[1]
    m <- (subRn[ub] - subRn[ub - 1])
    b <- subRn[ub - 1]
    estimatedCt[i1] <- (ub - 1) + (estimatedThreshold - b)/m
}
mean(estimatedCt, na.rm = TRUE)
```

```
## [1] 28.79
```

```
plot(defaultCt, ylim = c(min(c(defaultCt, estimatedCt), na.rm = TRUE), 40),
    ylab = "Number of Cycles", main = "FABP4 PCR Results")
abline(h = 40, lty = 2)
points(c(1:23)[!is.na(defaultCt)], defaultCt[!is.na(defaultCt)], type = "l")

points(estimatedCt, pch = 8, col = "red")
imputeCt <- estimatedCt
imputeCt[is.na(imputeCt)] <- 40
points(imputeCt, type = "l", col = "red")

legend(15, 17, c("default Ct", "estimated Ct"), cex = 0.8, col = c("black",
    "red"), pch = c(1, 8), bty = "n")
```



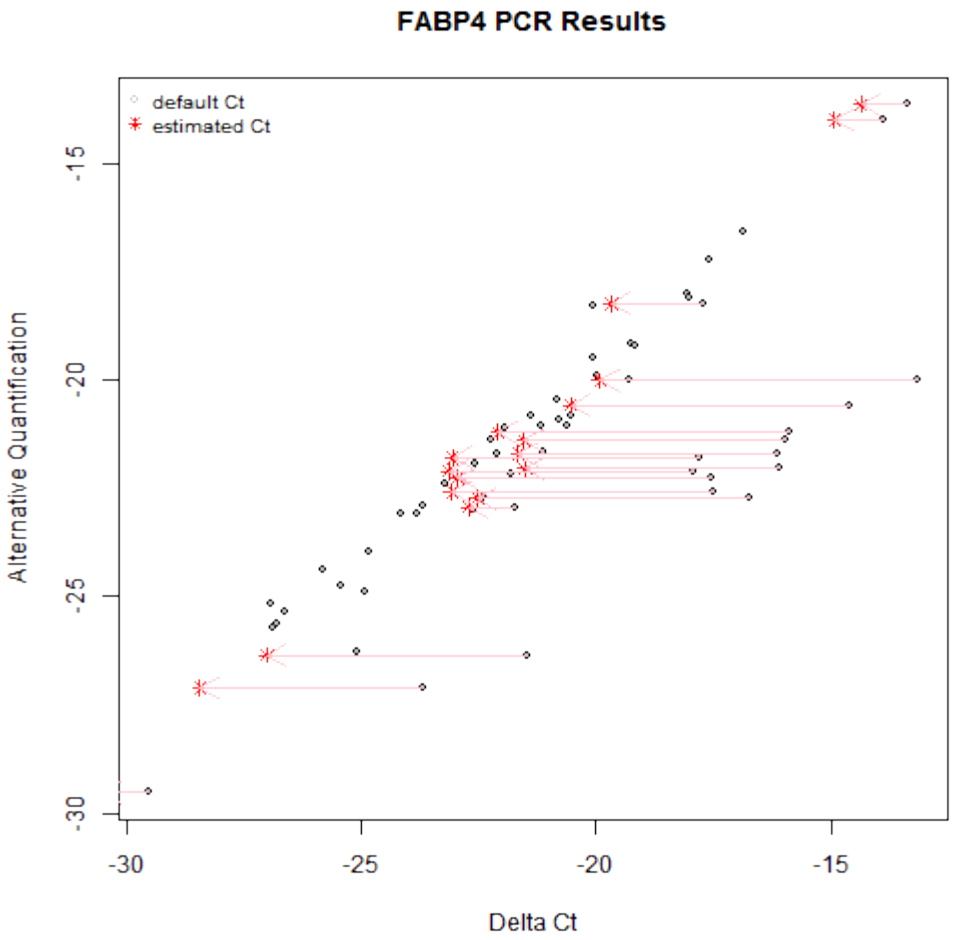
```
# flagging bad rows
goodRow <- rep(TRUE, nrow(rawPCRWash))
## Plate Plate.1 '- 2nd set 021613' all fine Plate Plate.2 '021213', Rerun
## samples
goodRow[(rawPCRWash$Sample.Name == "W37") & (rawPCRWash$Plate == "Plate.2")] <- FALSE
goodRow[(rawPCRWash$Sample.Name == "W32") & (rawPCRWash$Plate == "Plate.2")] <- FALSE
goodRow[(rawPCRWash$Sample.Name == "W20") & (rawPCRWash$Plate == "Plate.2")] <- FALSE
## Plate Plate.3 '021613'
goodRow[(rawPCRWash$Well == "A6") & (rawPCRWash$Plate == "Plate.3")] <- FALSE
goodRow[(rawPCRWash$Well == "B6") & (rawPCRWash$Plate == "Plate.3")] <- FALSE
goodRow[(rawPCRWash$Well == "F3") & (rawPCRWash$Plate == "Plate.3")] <- FALSE
goodRow[(rawPCRWash$Well == "G2") & (rawPCRWash$Plate == "Plate.3")] <- FALSE
## Plate Plate.4 '021813'; all 80-4 (rerun)
goodRow[(rawPCRWash$Well == "C2") & (rawPCRWash$Plate == "Plate.4")] <- FALSE
goodRow[(rawPCRWash$Well == "C3") & (rawPCRWash$Plate == "Plate.4")] <- FALSE
goodRow[(rawPCRWash$Well == "C5") & (rawPCRWash$Plate == "Plate.4")] <- FALSE
goodRow[(rawPCRWash$Well == "C6") & (rawPCRWash$Plate == "Plate.4")] <- FALSE
## Plate Plate.5 '3rd batch 201313'
goodRow[(rawPCRWash$Well == "F6") & (rawPCRWash$Plate == "Plate.5")] <- FALSE
## Plate W6 '4th set 021413' all fine Plate W7 'second machine' row B
## (sample 81-2) may be weak overall
rawPCRWash <- rawPCRWash[goodRow, ]
```

```
sampleCt <- function(x) {
    geneMeans <- sapply(split(x, factor(x$Target.Name)), function(y) mean(as.numeric(y$Ct),
        na.rm = TRUE))
    geneThresh <- unique(x$Ct.Threshold)
    names(geneThresh) <- unique(x$Target.Name)
    rv <- geneMeans["18S"] - geneMeans["FABP4"]
    if (geneThresh["FABP4"] < 1) {
        tmp <- x[which(x$Target.Name == "FABP4"), ]
        estimatedCt <- array(NA, nrow(tmp))
        for (i1 in 1:length(estimatedCt)) {
            subRn <- as.numeric(tmp[i1, grep("Cycle", colnames(tmp))])
            ub <- which(subRn > estimatedThreshold)[1]
            m <- (subRn[ub] - subRn[ub - 1])
            b <- subRn[ub - 1]
            estimatedCt[i1] <- (ub - 1) + (estimatedThreshold - b)/m
        }
        rv <- c(rv, geneMeans["18S"] - mean(estimatedCt, na.rm = TRUE))
    } else {
        rv <- c(rv, NA)
    }
    rv
}
```

```
ct <- NULL
for (i1 in 1:length(unique(rawPCRWash$Plate))) {
    tmp <- rawPCRWash[which(rawPCRWash$Plate == unique(rawPCRWash$Plate)[i1]),
        ]
    tmpCt <- t(sapply(split(tmp, factor(tmp$Sample.Name)), sampleCt))
    ct <- rbind(ct, tmpCt)
}
```

```
plot(ct[PCRResults$Sample.Name[which(PCRResults$Source == "Washington")], 1],
    PCRResults$FABP4[which(PCRResults$Source == "Washington")], main = "FABP4 PCR Results",
    xlab = "Delta Ct", ylab = "Alternative Quantification", pch = 21, bg = "grey",
    cex = 0.75)
points(ct[PCRResults$Sample.Name[which(PCRResults$Source == "Washington")],
    2], PCRResults$FABP4[which(PCRResults$Source == "Washington")], col = "red",
    pch = 8)
arrows(ct[PCRResults$Sample.Name[which(PCRResults$Source == "Washington")],
    1], PCRResults$FABP4[which(PCRResults$Source == "Washington")], x1 =
ct[PCRResults$Sample.Name[which(PCRResults$Source ==
    "Washington")], 2], col = "pink", length = 0.15)
legend("topleft", c("default Ct", "estimated Ct"), cex = 0.8, col = c("grey",
```

```
    "red"), pch = c(1, 8), bty = "n")
```



Appendix

```
getwd()
```

```
## [1] "\\\\mdadqsfs02/workspace/kabagg/RDPaper/Webpage/ResidualDisease"
```

```
sessionInfo()
```

```
## R version 2.15.3 (2013-03-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] RColorBrewer_1.0-5 knitr_1.2
##
## loaded via a namespace (and not attached):
```

```
## [1] digest_0.6.3   evaluate_0.4.3 formatR_0.7    stringr_0.6.2
## [5] tools_2.15.3
```

Residual Disease Paper

Quantifying Public PCR Data

by Shelley Herbrich

1 Executive Summary

1.1 Introduction

In this report, we present the script used to generate the final PCR quantification summaries for ADH1B and FABP4.

1.2 Data & Methods

We use the “rawPCRData.RData” file which contains the well-specific source, plate, randomized sample identifier, target gene, baseline-corrected fluorescence measurements for 40 cycles, and baseline coordiantes. For each sample, we have 2 to 3 technical replicates for both ADH1B and FABP4 as well as the control, 18S. Some wells were manually filtered for poor quality.

We chose to use a “window-of-linearity” method, as introduced by Ramakers et al (2003), to quantify the PCR results from the baseline-corrected fluorescence measurements obtained from the vendor software. This method fits a linear model to the cycle number versus log fluorescence within a sliding window of defined size within a fixed border. The optimal model is that with the maximum log-linear range. Based on this model, the intercept corresponds to the initial template fluorescence and the slope is an estimate of the PCR efficiency. This algorithm is implemented in **sliwin** (part of the qpcr package).

We require that the model not be fit within the baseline by forcing the lower bound of the border to be greater than the “take-off point” of exponential growth. Because the 18S fluorescence takes off almost immediately, we allow for a smaller window size to fit the optimal linear model.

1.3 Results

We generate the ADH1B and FABP4 PCR values that correspond to the “PCRResults.RData” object.

2 Options and Libraries

We load the libraries we will use in this report.

```
library(qpcR)
```

3 Loading Data

Next, we load the deidentified raw PCR data.

```
load(file.path("RDataObjects", "rawPCRData.RData"))
```

4 Quantifying PCR Data

We quantify the PCR measurements for FABP4 and ADH1B using the method described above and display the results.

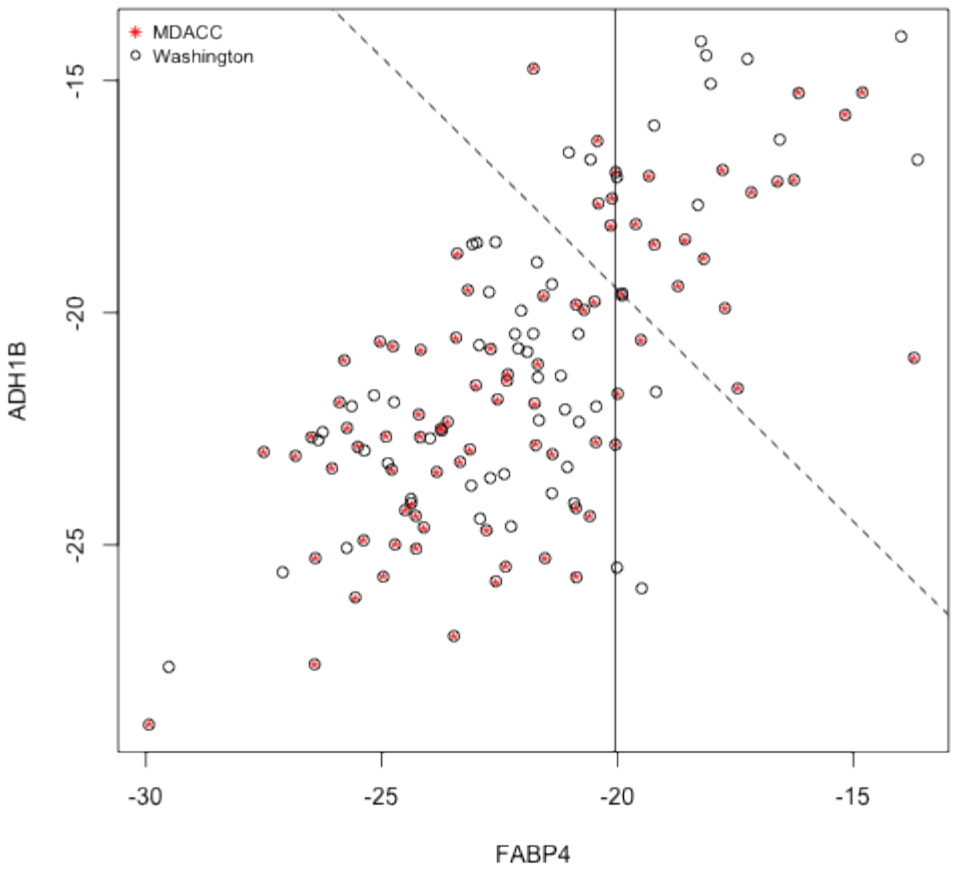
```
sampleID <- unique(rawPCRData$Sample.Name)
PCRquantifications <- data.frame(Source = rep("", length(sampleID)), Plate = rep("",
    length(sampleID)), Sample.Name = sampleID, FABP4 = rep(NA, length(sampleID)),
    ADH1B = rep(NA, length(sampleID)), stringsAsFactors = FALSE)
```

```
for (i1 in 1:length(sampleID)) {
    tmp <- rawPCRData[which(rawPCRData$Sample.Name == sampleID[i1]), ]
    tempNo <- array(NA, length(tmp$Well))
    for (i2 in 1:length(tmp$Well)) {
        subDRn <- cbind(1:40, t(tmp[i2, grep("Cycle", colnames(tmp))]))
        m <- pcrfit(subDRn, cyc = 1, fluo = 2, model = l3)
        to <- tryCatch(takeoff(m)$top, error = function(e) NA)
        border <- c(0, 0)
        if (to < tmp[i2, "Baseline.End"] & !is.na(to))
            border <- c(tmp[i2, "Baseline.End"] - to, 0)
        if (tmp$Target.Name[i2] == "18S") {
            sw <- tryCatch(sliwin(m, border = border, wsize = 4:6, plot = FALSE),
                error = function(e) NULL)
        } else {
            sw <- tryCatch(sliwin(m, border = border, plot = FALSE), error = function(e) NULL)
        }
        if (!is.null(sw))
            tempNo[i2] <- unlist(sw["init"])
    }
    targetMeans <- sapply(split(tempNo, tmp$Target.Name), mean, na.rm = TRUE)
    PCRquantifications[i1, "Source"] <- as.character(tmp$Source[1])
    PCRquantifications[i1, "Plate"] <- tmp$Plate[1]
    PCRquantifications[i1, "FABP4"] <- log2(targetMeans["FABP4"]) - log2(targetMeans["18S"])
    PCRquantifications[i1, "ADH1B"] <- log2(targetMeans["ADH1B"]) - log2(targetMeans["18S"])
}
PCRquantifications <- PCRquantifications[order(PCRquantifications$FABP4, decreasing = TRUE),
    ]
```

```
head(PCRquantifications)
```

```
##         Source    Plate Sample.Name  FABP4  ADH1B
## 25  Washington  Plate.4         W20 -13.64 -16.71
## 61       MDACC  Plate.9         M33 -13.71 -20.97
## 26  Washington  Plate.4         W46 -13.99 -14.05
## 95       MDACC Plate.13         M80 -14.81 -15.26
## 139      MDACC Plate.29         M71 -15.18 -15.74
## 72       MDACC Plate.10         M61 -16.16 -15.27
```

```
plot(PCRquantifications$FABP4, PCRquantifications$ADH1B, xlab = "FABP4", ylab = "ADH1B")
points(PCRquantifications$FABP4[which(PCRquantifications$Source == "MDACC")],
    PCRquantifications$ADH1B[which(PCRquantifications$Source == "MDACC")], pch = "*",
    col = "red")
legend("topleft", c("MDACC", "Washington"), pch = c(8, 1), col = c("red", "black"),
    bty = "n", cex = 0.8)
abline(v = -20.05)
abline(a = -39.5, b = -1, lty = 2)
```



5 Appendix

```
getwd()
```

```
## [1] "/Users/slt/SLT WORKSPACE/EXEMPT/OVARIAN/Ovarian residual disease study 2012/RD
manuscript/Web page for paper/Webpage"
```

```
sessionInfo()
```

```
## R version 3.0.2 (2013-09-25)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] qpcR_1.3-7.1      robustbase_0.9-10 rgl_0.93.963      minpack.lm_1.1-8
## [5] MASS_7.3-29       knitr_1.5
##
## loaded via a namespace (and not attached):
## [1] evaluate_0.5.1 formatR_0.9    stringr_0.6.2  tools_3.0.2
```

Residual Disease Paper

Comparing Ovary and Omentum Expression

by Shelley Herbrich

May 13, 2013

1 Executive Summary

1.1 Introduction

In this report, we compare the PCR expression of our target genes between ovary and omental samples.

We also compare FABP4 exression between ovary and omental samples from patients who had both available.

1.2 Data & Methods

In comparing PCR expression of our target genes between ovary and omental samples, we use PCR quantification summaries of samples run on the 17 plates containing tissue from both ovary and omentum.

1.3 Results

We produce density plots of ADH1B and FABP4 expression by tissue.

We see that for both genes, omental levels are markedly higher than ovary.

We produce a scatterplot of paired FABP4 values for 4 cases with data from both ovary and omentum available.

The measurement from omentum is higher than that from ovary in all 4 cases, with percent differences of 2%, 85%, 110% and 131%.

2 Loading PCR Data

```
library(RColorBrewer)
```

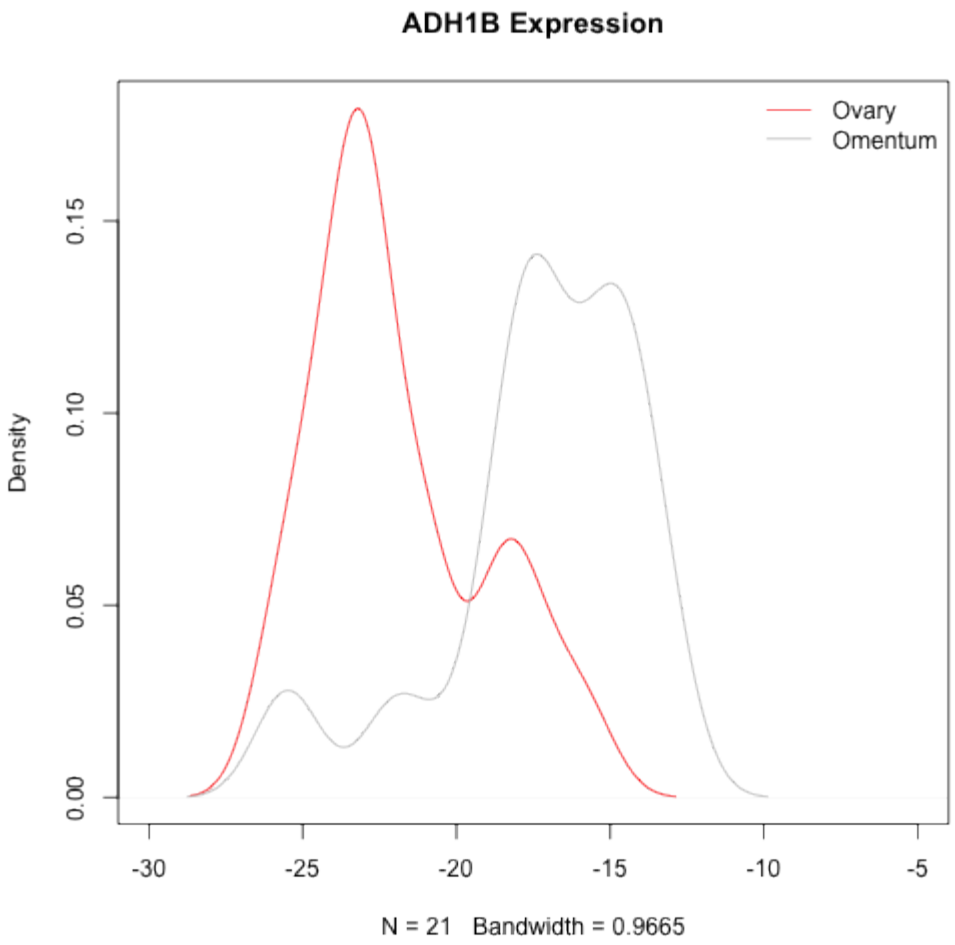
```
load(file.path("RDataObjects", "OMResults.RData"))
load(file.path("RDataObjects", "PCRResults.RData"))
```

```
# match plates
plate <- unique(OMResults$Plate)
ovFABP4 <- PCRResults$FABP4[which(PCRResults$Plate %in% plate)]
ovADH1B <- PCRResults$ADH1B[which(PCRResults$Plate %in% plate)]
```

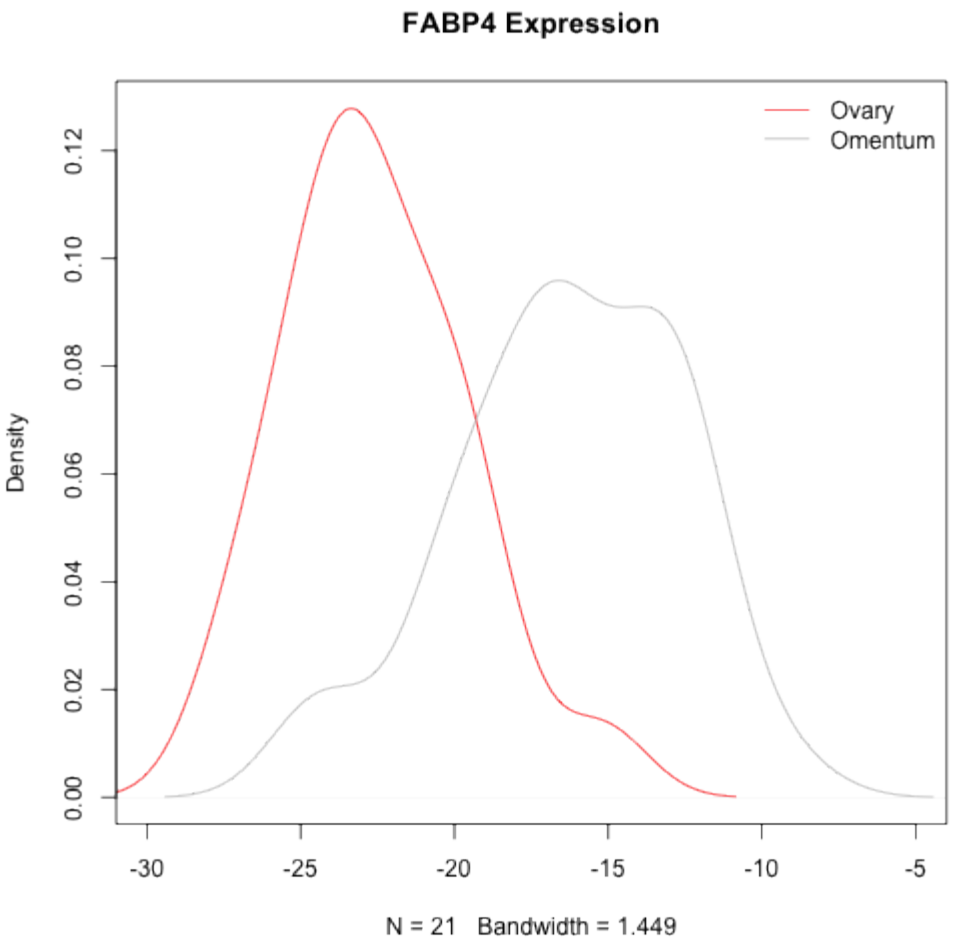
3 Analyses

3.1 Plotting Densities of ADH1B and FABP4 Expression

```
# pdf(file='OVOMExpression.pdf',paper='USr')
plot(density(ovADH1B), col = "red", xlim = c(-30, -5), main = "ADH1B Expression")
lines(density(OMResults$ADH1B), col = "grey")
legend("topright", c("Ovary", "Omentum"), col = c("red", "grey"), lty = 1, bty = "n")
```




```
plot(density(ovFABP4), col = "red", xlim = c(-30, -5), main = "FABP4 Expression")
lines(density(OMResults$FABP4), col = "grey")
legend("topright", c("Ovary", "Omentum"), col = c("red", "grey"), lty = 1, bty = "n")
```



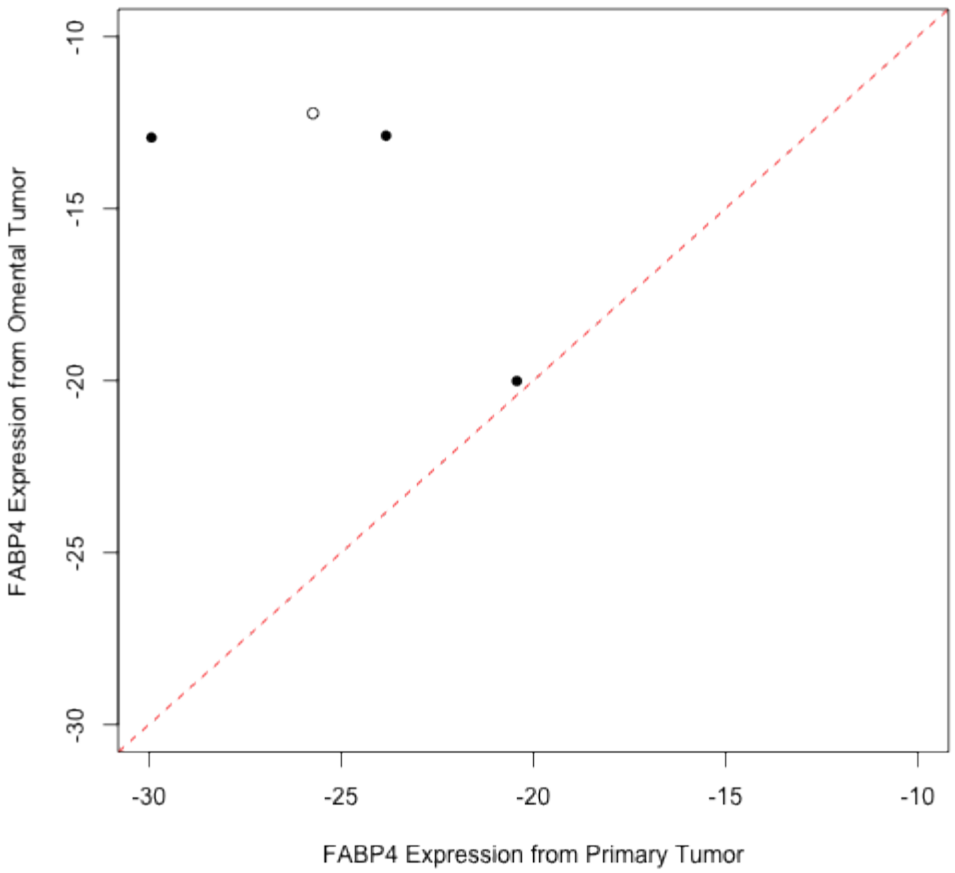

```
# dev.off()
```

3.2 Scatterplot of FABP4 and ADH1B from omentum versus primary

The following scatterplot shows FABP4 values in primary tumor and omentum for the 4 patients in the validation cohort who had both types of tumor tissue assayed.

Solid symbols show cases with RD. The dashed line indicates equality.

```
pairsOV <- c("M20", "M59", "M65", "M34")
pairsOM <- c("OM1", "OM20", "OM40", "OM3")
sym <- c(16, 16, 1, 16)

ovPaired <- c()
omPaired <- c()

for (i in 1:4) {
    ovPaired <- c(ovPaired, PCRResults$FABP4[which(PCRResults$Sample.Name ==
        pairsOV[i])])
    omPaired <- c(omPaired, OMResults$FABP4[which(OMResults$Sample.Name == pairsOM[i])])
}

plot(ovPaired, omPaired, xlab = "FABP4 Expression from Primary Tumor", ylab = "FABP4 Expression
from Omental Tumor",
    xlim = c(-30, -10), ylim = c(-30, -10), pch = sym)

abline(0, 1, col = "red", lty = 2)
```




```
ovPaired/omPaired
```

```
## [1] 1.850 1.021 2.104 2.314
```

Appendix

```
getwd()
```

```
## [1] "/Users/slt/SLT WORKSPACE/EXEMPT/OVARIAN/Ovarian residual disease study 2012/RD
manuscript/Web page for paper/Webpage"
```

```
sessionInfo()
```

```
## R version 3.0.2 (2013-09-25)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] RColorBrewer_1.0-5 knitr_1.5
##
## loaded via a namespace (and not attached):
```

```
## [1] evaluate_0.5.1 formatR_0.10   stringr_0.6.2  tools_3.0.2
```

Power Calculations for Assigning Patients to RD Risk Groups in the Validation Cohort

Susan L. Tucker

1 Executive Summary

1.1 Introduction

The goal of this analysis is to perform power calculations needed to select a method for assigning patients to high-risk versus lower-risk residual disease (RD) groups in our validation study.

1.2 Data & Methods

1.2.1 Planned validation test

In our validation study, we measured FABP4 expression levels by qRT-PCR for 139 tumor samples.

To test our hypothesis that RD occurs more frequently among patients with high FABP4 levels, we will compare RD incidence in subgroups of patients predicted to be at high risk versus lower risk of RD.

Specifically, we will assign a pre-determined number of patients (nHigh) to the high risk group, and the remaining patients (139 - nHigh) to the low risk group. A 1-sided Fisher's exact test will be used to compare the incidence of RD in the two risk groups.

Therefore, the specific goal of this report is to select a value for nHigh, based on power calculations.

1.2.2 Power calculations

The power calculation proceeds as follows.

Using a binomial random-number generator, we randomly generate outcomes (RD versus no RD) based on assumed probabilities of RD in the high-risk and low-risk groups.

The observed (randomly generated) RD rates in the high and low risk groups are compared using the 1-sided Fisher exact test. The test is considered a success if $P<0.05$.

The two steps described above are repeated for a total of 10,000 simulations, and the proportion of successful tests (the estimated power of the test) is recorded.

1.2.3 Selecting probabilities

In performing the power calculations, we assume that the overall probability of RD in the validation cohort (pRD) is 75%, based on the observed RD rates of 77% (378/491) and 73% (136/186) in the TCGA and Tothill data sets, respectively.

To determine an appropriate value for the probability of RD in the high-risk group (pRDhigh), we compute the positive predictive value (PPV = proportion of patients with RD in the high-risk group) in the TCGA and Tothill data sets as a function of the call rate (the proportion of patients assigned to the high-risk group). PPV will serve as an estimate of pRDhigh.

We note that the probability of RD in the low-risk group (pRDlow) can be computed from pRD and pRDhigh, as described later in this report, so it need not be specified separately in performing the power calculations.

1.2.4 Data used

To compute PPV as a function of call rate in the TCGA and Tothill data sets, we use the RData objects containing clinical and gene expression data that were created in previous reports (assembleTCGAClinical, assembleTothillClinical). Patients are filtered as described previously (filterTCGASamples, filterTothillSamples).

1.3 Results

The power calculations show that assigning 20%, 25% or 30% of patients to the high-risk group would likely provide very high power (>90%) of achieving success in our validation study provided PPV values are at the high end of the range seen in the TCGA and Tothill data sets.

For PPV values on the low end of the range seen for TCGA and Tothill, call rates of 20% or 30% would likely provide relatively low power (50-60%).

For a call rate of 25%, the calculations suggest that the power will be reasonably high (>76%) for any value of PPV falling within the range observed for TCGA and Tothill.

1.4 Conclusion

Based on the power calculations, we elect to assign the 25% of patients with the highest FABP4 values to our predicted high-risk group.

2 Loading & Filtration of Data

The TCGA and Tothill data objects created previously are loaded.

```
load(file.path("RDataObjects", "tcgaExpression.RData"))
load(file.path("RDataObjects", "tcgaClinical.RData"))
load(file.path("RDataObjects", "tcgaFilteredSamples.RData"))

load(file.path("RDataObjects", "tothillExpression.RData"))
load(file.path("RDataObjects", "tothillClinical.RData"))
load(file.path("RDataObjects", "tothillFilteredSamples.RData"))
```

Filtrations are applied to the TCGA data.

```
tcgaSampleUseLong <- rownames(tcgaFilteredSamples[which(tcgaFilteredSamples[,
    "sampleUse"] == "Used"), ])
tcgaSampleUse <- substr(tcgaSampleUseLong, 1, 12)
tcgaRDUse <- tcgaRD[tcgaSampleUse]
tcgaExprUse <- tcgaExpression[, tcgaSampleUseLong]
colnames(tcgaExprUse) <- tcgaSampleUse
```

Filtrations are applied to the Tothill data.

```
tothillSampleUse <- rownames(tothillFilteredSamples[which(tothillFilteredSamples[,
    "sampleUse"] == "Used"), ])
tothillRDUse <- tothillRD[tothillSampleUse]
tothillExprUse <- tothillExpression[, tothillSampleUse]
```

3 Analyses

3.1 Positive predictive value (PPV) in TCGA and Tothill

Before performing power calculations, we calculate PPV in the TCGA and Tothill data sets as a function of the proportion of patients assigned to the high risk group based on FABP4 expression levels.

We first obtain the vectors of FABP4 expression values for the two data sets.

```
library(hthgu133a.db)
probeFABP4 <- names(which(unlist(mget(rownames(tcgaExprUse), hthgu133aSYMBOL)) ==
    "FABP4"))
probeFABP4
```

```
## [1] "203980_at"
```

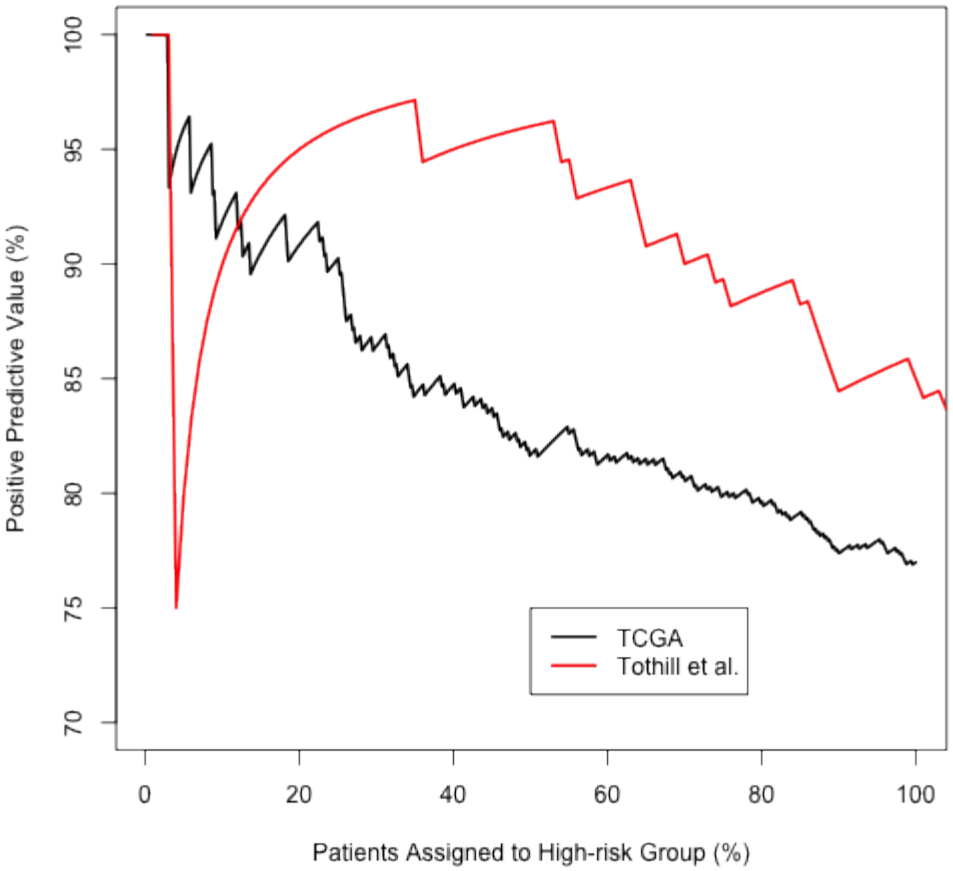
```
tcgaFABP4 <- tcgaExprUse[probeFABP4, ]
tothillFABP4 <- tothillExprUse[probeFABP4, ]
```

We next sort the vectors by FABP4 expression level.

```
tcgaFABP4sorted <- tcgaFABP4[order(tcgaFABP4)]
tcgaRDsorted <- tcgaRDUse[order(tcgaFABP4)]

tothillFABP4sorted <- tothillFABP4[order(tothillFABP4)]
tothillRDsorted <- tothillRDUse[order(tothillFABP4)]
```

PPV is computed and plotted as a function of call rate.

```
tcgaNPts <- length(tcgaFABP4)
tcgaPctCall <- seq(from = tcgaNPts, to = 1, by = -1)
tcgaPPV <- c()
for (i in 1:tcgaNPts) {
    tcgaPPV <- c(tcgaPPV, length(which(tcgaRDsorted[i:tcgaNPts] == "RD")))
}
tcgaPPV <- 100 * tcgaPPV/tcgaPctCall
tcgaPctCall <- 100 * tcgaPctCall/tcgaNPts

tothillNPts <- length(tothillFABP4)
tothillPctCall <- seq(from = tothillNPts, to = 1, by = -1)
tothillPPV <- c()
for (i in 1:tothillNPts) {
    tothillPPV <- c(tothillPPV, length(which(tothillRDsorted[i:tothillNPts] ==
        "RD")))
}
tothillPPV <- 100 * tothillPPV/tothillPctCall
tothillPCall <- 100 * tothillPctCall/tothillNPts

plot(tcgaPctCall, tcgaPPV, type = "l", xlab = "Patients Assigned to High-risk Group (%)",
    ylab = "Positive Predictive Value (%)", ylim = c(70, 100), lwd = 2)
lines(tothillPctCall, tothillPPV, col = c("red"), lwd = 2)
legend(x = 50, y = 75, legend = c("TCGA", "Tothill et al."), lty = c(1, 1),
    lwd = 2, col = c("black", "red"))
```



3.2 Power calculations

The following notation is used here:

- pRD = overall probability of RD in the validation cohort
- nHigh = number of patients assigned to the high-risk group
- pRDhigh = expected probability of RD among patients assigned to the high-risk group
- pRDlow = expected probability of RD among patients assigned to the lower-risk group
- pCall = call rate = proportion of patients assigned to the high-risk group = nHigh / 139

We observe that pRD = pRDhigh * pCall + pRDlow * (1 - pCall). Therefore pRDlow can be determined once pRD, pCall and pRDhigh are specified.

We begin by defining the function that performs the power calculation.

```
powerCalc <- function(nSam, nCall, pRD, pRDhigh, sig, nits) {
    # nSam = number of samples (to be fixed here at 139) nCall = number called
    # with RD sig = significance level defining success of the 1-sided Fisher's
    # exact test (to be fixed here at P=0.05) nits = number of iterations in the
    # numerical simulation pCall and pRD as defined above

    pCall <- nCall/nSam
    pRDlow = (pRD - pRDhigh * pCall)/(1 - pCall)

    callRD <- c(rep(1, nCall), rep(0, nSam - nCall))

    fisherP <- c()

    for (i in 1:nits) {
        observedRD <- c()
```

```
        for (j in 1:nCall) {
            observedRD <- c(observedRD, rbinom(1, 1, pRDhigh))
        }
        for (j in 1:(nSam - nCall)) {
            observedRD <- c(observedRD, rbinom(1, 1, pRDlow))
        }
        fisherP <- c(fisherP, fisher.test(callRD, observedRD, or = 1, alternative = "greater",
            conf.int = FALSE)$p.value)
    }
    power <- sum(fisherP < sig)/nits
    return(power)
}
```

We now run the power calculations. For each calculation, we fix the number of samples at nSam = 139, the overall probability of RD at 75%, and the number of iterations at 10,000.

We assume that the number of patients predicted to be in the high-risk group (nCall) is 28, 35 or 42, corresponding to call rates of about 20%, 25% or 30%, respectively.

Based on the PPV plots obtained from the TCGA and Tothill data, we assume that pRDhigh = 90, 92.5 or 95% when the call rate is 20-25%, and we assume that pRDhigh = 85, 90 or 95% when the call rate is 30%. These represent approximately upper and lower limits for the PPV, as well as an intermediate value.

```
powerEst <- matrix(0, nrow = 9, ncol = 3)
colnames(powerEst) <- c("nCall", "pRDhigh", "Power")
rownames(powerEst) <- rep("", nrow(powerEst))

set.seed(5913)
powerEst[1, ] <- c(28, 0.95, powerCalc(139, 28, 0.75, 0.95, 0.05, 10000))
powerEst[2, ] <- c(28, 0.925, powerCalc(139, 28, 0.75, 0.925, 0.05, 10000))
powerEst[3, ] <- c(28, 0.9, powerCalc(139, 28, 0.75, 0.9, 0.05, 10000))

powerEst[4, ] <- c(35, 0.95, powerCalc(139, 35, 0.75, 0.95, 0.05, 10000))
powerEst[5, ] <- c(35, 0.925, powerCalc(139, 35, 0.75, 0.925, 0.05, 10000))
powerEst[6, ] <- c(35, 0.9, powerCalc(139, 35, 0.75, 0.9, 0.05, 10000))

powerEst[7, ] <- c(42, 0.95, powerCalc(139, 42, 0.75, 0.95, 0.05, 10000))
powerEst[8, ] <- c(42, 0.9, powerCalc(139, 42, 0.75, 0.9, 0.05, 10000))
powerEst[9, ] <- c(42, 0.85, powerCalc(139, 42, 0.75, 0.85, 0.05, 10000))
```

We list the results of the power calculations.

```
powerEst
```

```
##  nCall pRDhigh  Power
##     28   0.950 0.9098
##     28   0.925 0.7925
##     28   0.900 0.6224
##     35   0.950 0.9750
##     35   0.925 0.8981
##     35   0.900 0.7622
##     42   0.950 0.9916
##     42   0.900 0.8606
##     42   0.850 0.5015
```

4 Appendix

4.1 File Location

```
getwd()
```

```
## [1] "/Users/slt/SLT WORKSPACE/EXEMPT/OVARIAN/Ovarian residual disease study 2012/RD
```

```
manuscript/Web page for paper/Webpage"
```

4.2 SessionInfo

```
sessionInfo()
```

```
## R version 3.0.2 (2013-09-25)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] hthgu133a.db_2.9.0   org.Hs.eg.db_2.9.0   RSQLite_0.11.4
## [4] DBI_0.2-7            AnnotationDbi_1.22.6 Biobase_2.20.1
## [7] BiocGenerics_0.6.0   knitr_1.5
##
## loaded via a namespace (and not attached):
## [1] evaluate_0.5.1 formatR_0.9    IRanges_1.18.4 stats4_3.0.2
## [5] stringr_0.6.2  tools_3.0.2
```

Residual Disease Paper

Comparing RD Results

by Shelley Herbrich

1 Executive Summary

1.1 Introduction

Using the true residual disease (RD) status for the validation cohort, we are interested to check our predictions using FABP4 and ADH1B.

1.2 Data and Methods

We work with the results dataset, *PCRResults*.

For both target genes, we define our subset of patients with enriched proportion of residual disease as those with the top 25% of expression (this corresponds to the top 35 samples).

1.3 Results

We plot the sorted log2 FABP4 and ADH1B values based on our quantification method. We also plot ADH1B against FABP4.

2 Loading Libraries and Quantification Data

We load the PCR results, containing our quantification summaries and true RD status.

```
library(qpcR)
library(gdata)
```

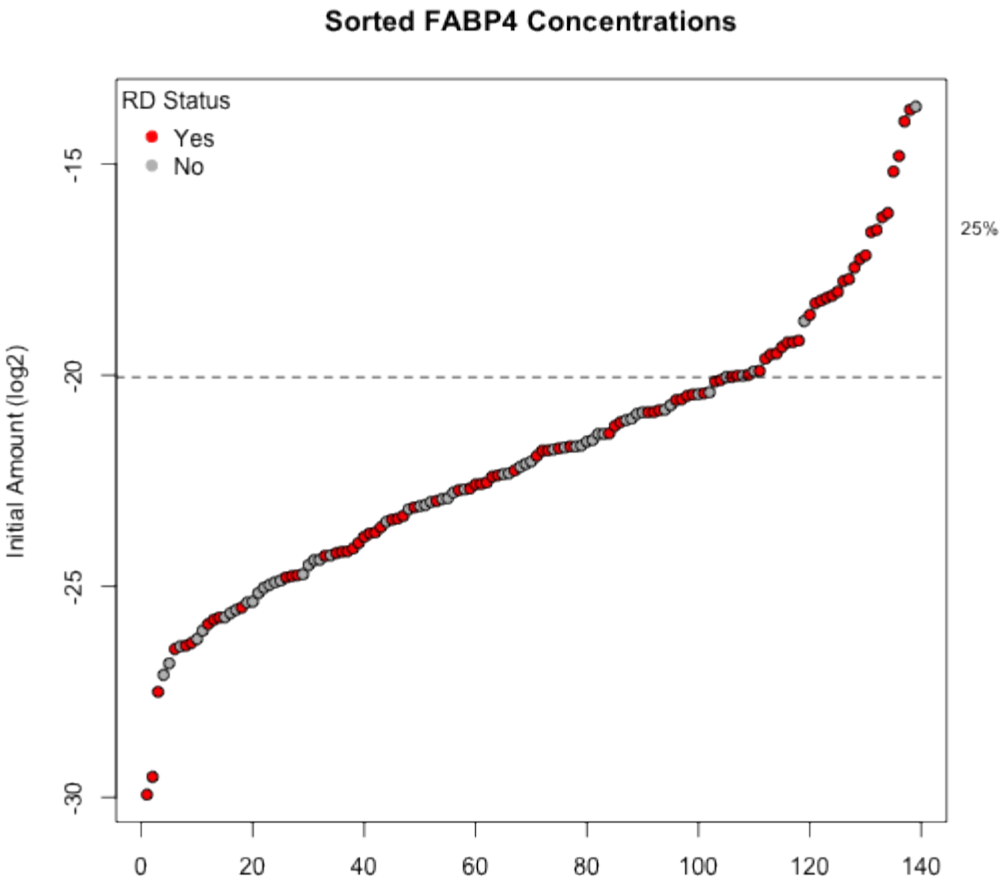
```
load(file.path("RDataObjects", "PCRResults.RData"))
load(file.path("RDataObjects", "rawPCRData.RData"))
```

```
sampleID <- PCRResults$Sample.Name
rd <- PCRResults$RDStatus
names(rd) <- sampleID
```

3 Flagging RD Using FABP4

First, we graphically examine our cutoff of the top 25th percentile based on levels of FABP4.

```
plot(rev(PCRResults$FABP4), ylab = "Initial Amount (log2)", xlab = "", pch = 21,
    bg = c("grey", "red")[rev(factor(rd))], main = "Sorted FABP4 Concentrations")
abline(h = -20.05, lty = 2)
mtext("25%", side = 4, at = -16.5, las = 2, line = 0.5, cex = 0.8)
legend("topleft", c("Yes", "No"), pch = 19, col = c("red", "grey"), bty = "n",
    title = "RD Status")
```



We do see a subgroup with an enriched proportion of residual disease that is associated with high FABP4. In our cohort where the overall percentage of patients with residual disease is 60%, we are able to identify a subgroup with 86% residual disease.

```
table(rd[1:35])/sum(table(rd[1:35]))
```

```
##
##     No    Yes
## 0.1429 0.8571
```

```
table(rd[36:139])/sum(table(rd[36:139]))
```

```
##
##     No    Yes
## 0.4808 0.5192
```

```
fisher.test(matrix(c(30, 5, 54, 50), ncol = 2), alternative = "greater")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  matrix(c(30, 5, 54, 50), ncol = 2)
## p-value = 0.0002489
## alternative hypothesis: true odds ratio is greater than 1
## 95 percent confidence interval:
##  2.191   Inf
## sample estimates:
## odds ratio
```

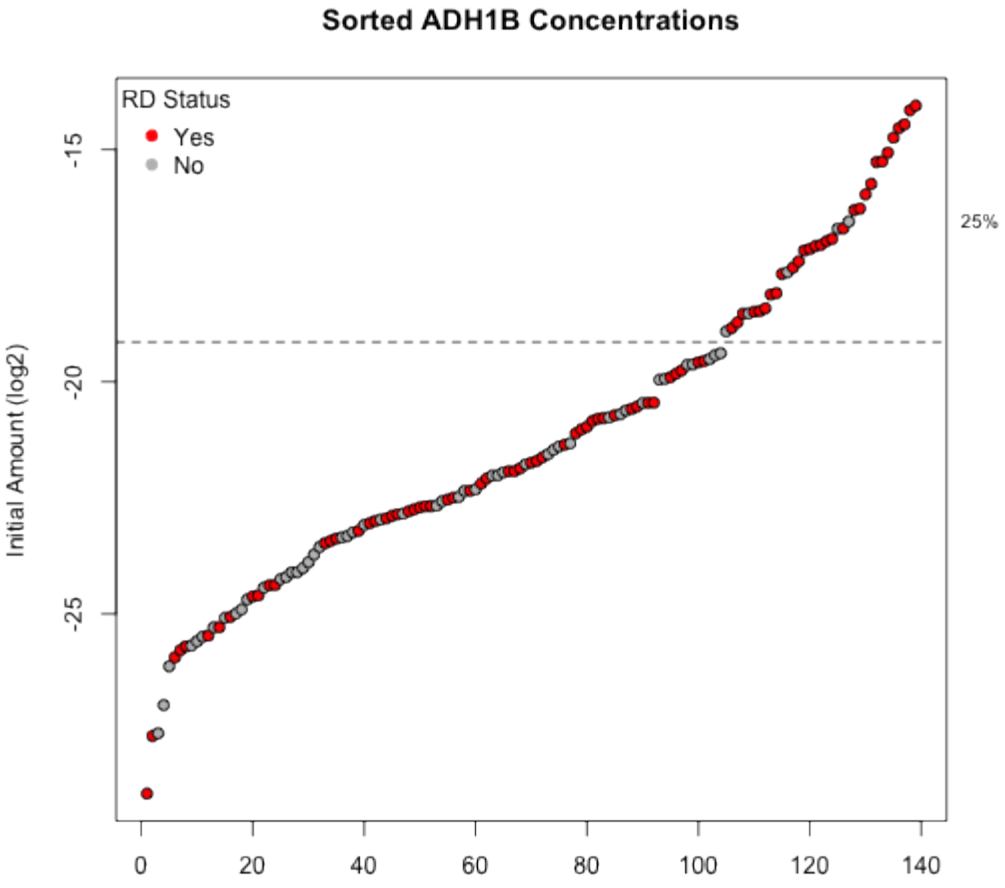
```
##      5.494
```

Based on a one-sided Fisher's Exact test, the difference in proportion of residual disease is significantly higher for those with elevated FABP4.

4 Flagging RD Using ADH1B

Now, we look at the top 25th percentile based on ADH1B.

```
orderADH1B <- order(PCRResults$ADH1B)

plot(PCRResults$ADH1B[orderADH1B], ylab = "Initial Amount (log2)", xlab = "",
    pch = 21, bg = c("grey", "red")[factor(rd[orderADH1B])], main = "Sorted ADH1B
Concentrations")
abline(h = -19.15, lty = 2)
mtext("25%", side = 4, at = -16.5, las = 2, line = 0.5, cex = 0.8)
legend("topleft", c("Yes", "No"), pch = 19, col = c("red", "grey"), bty = "n",
    title = "RD Status")
```



Using ADH1B alone, we are also able to define a subgroup with an enriched proportion (86%) of residual disease.

```
table(rd[rev(orderADH1B)[1:35]])/sum(table(rd[rev(orderADH1B)[1:35]]))
```

```
##
##     No    Yes
## 0.1429 0.8571
```

```
table(rd[rev(orderADH1B)[36:139]])/sum(table(rd[rev(orderADH1B)[36:139]]))
```

```
##
##     No    Yes
## 0.4808 0.5192
```

```
fisher.test(matrix(c(30, 5, 54, 50), ncol = 2), alternative = "greater")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  matrix(c(30, 5, 54, 50), ncol = 2)
## p-value = 0.0002489
## alternative hypothesis: true odds ratio is greater than 1
## 95 percent confidence interval:
##  2.191   Inf
## sample estimates:
## odds ratio
##      5.494
```

Again, we see the difference in proportion of residual disease is significantly higher for those with elevated ADH1B.

```
byBoth <- intersect(PCRResults$Sample.Name[1:35], PCRResults$Sample.Name[rev(orderADH1B)[1:35]])
rd[byBoth]
```

```
##   W20   W46   M80   M71   M61   M22   W38   M64   M24   W34   M54    W4
##  "No" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes"
##   W44   M52   W28   W55   M81   M32   W22   M76   M18   W16   M40
## "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes" "Yes"
```

Of the 35 samples flagged by either marker, 23 were flagged by both (22 RD, 1 no RD).

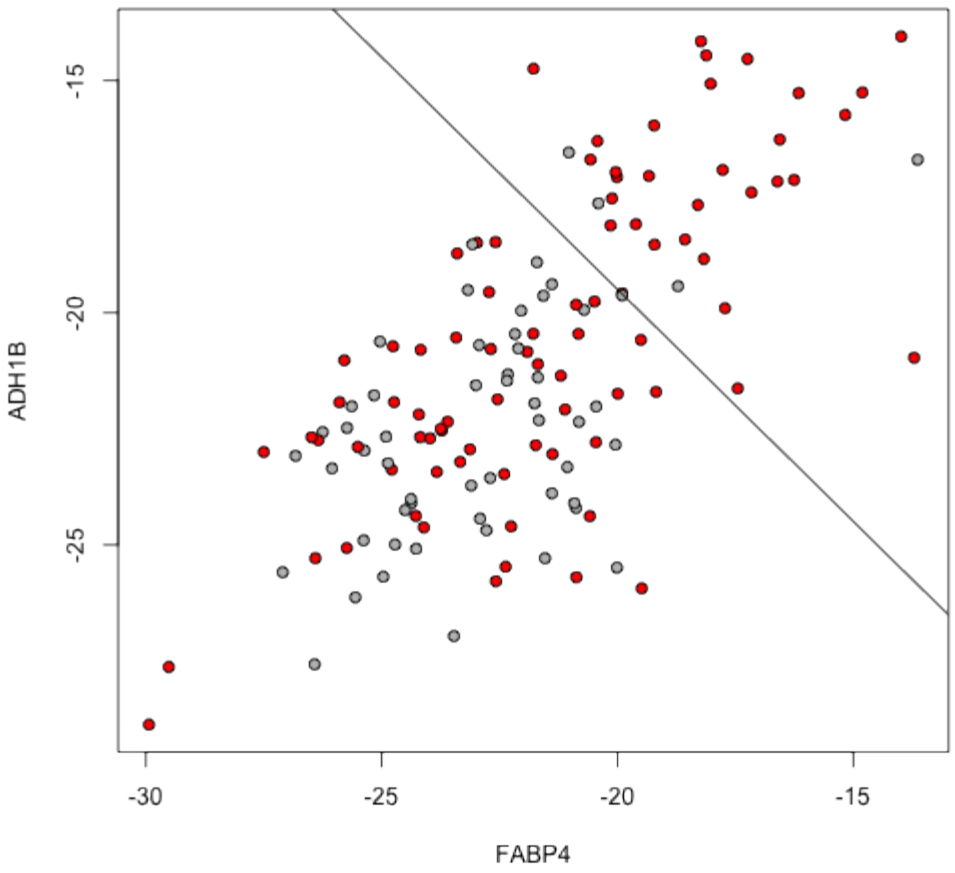
```
rawPCRData[which(rawPCRData$Sample.Name == "W20"), 1:5]
```

```
##         Source   Plate Well Sample.Name Target.Name
## 205 Washington Plate.4   C1         W20       ADH1B
## 208 Washington Plate.4   C4         W20       FABP4
## 211 Washington Plate.4   C7         W20         18S
## 212 Washington Plate.4   C8         W20         18S
## 213 Washington Plate.4   C9         W20         18S
```

Here, we note that for the single sample with RD two wells for both ADH1B and FABP4 were removed due to poor PCR quality leaving only a single replicate to quantify each target gene.

5 Flagging RD Using Both ADH1B and FABP4

```
plot(PCRResults$FABP4, PCRResults$ADH1B, ylab = "ADH1B", xlab = "FABP4", pch = 21,
    bg = c("grey", "red")[factor(rd)], main = "")
abline(a = -39.5, b = -1)
```




```
sum(-39.5 - PCRResults$FABP4 < PCRResults$ADH1B, na.rm = TRUE)
```

```
## [1] 35
```

```
byBothSim <- PCRResults$Sample.Name[which(-39.5 - PCRResults$FABP4 < PCRResults$ADH1B)]
table(rd[byBothSim])
```

```
##
##  No Yes
##   4  31
```

By using both markers simultaneously, we improve our enriched subgroup to 89% residual disease.

Appendix

```
getwd()
```

```
## [1] "/Users/slt/SLT WORKSPACE/EXEMPT/OVARIAN/Ovarian residual disease study 2012/RD
manuscript/Web page for paper/Webpage"
```

```
sessionInfo()
```

```
## R version 3.0.2 (2013-09-25)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
```

```
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] gdata_2.13.2      qpcR_1.3-7.1      robustbase_0.9-10 rgl_0.93.963
## [5] minpack.lm_1.1-8  MASS_7.3-29       knitr_1.5
##
## loaded via a namespace (and not attached):
## [1] evaluate_0.5.1 formatR_0.9    gtools_3.1.0   stringr_0.6.2
## [5] tools_3.0.2
```

Residual Disease Paper

Plotting RD Status

by Shelley Herbrich

May 8, 2013

1 Executive Summary

1.1 Introduction

In this report, we present the script used to display RD by ADH1B and FABP4.

1.2 Data & Methods

We use the "PCRResults.RData" file which contains the sample-specific source, plate, randomized sample identifier, ADH1B and FABP4 quantifications, RD call, and true RD status.

1.3 Results

We generate a plot of the ADH1B versus FABP4 PCR values indicating samples with RD by red points. The solid vertical line corresponds to the final threshold used to identify samples with high FABP4 (top 25%). The dashed line shows the partitioning of samples using both FABP4 and ADH1B.

2 Loading Data

First, we load the deidentified PCR results.

```
load(file.path("RDataObjects", "PCRResults.RData"))
```

3 Plotting PCR Data

We plot the PCR measurements for ADH1B against FABP4 indicating RD in red.

```
pdf(file = "plottingRD.pdf", paper = "USr")
rd <- factor(PCRResults$RDStatus)
plot(PCRResults$FABP4, PCRResults$ADH1B, pch = 21, bg = c("grey", "red")[rd],
    xlab = "FABP4", ylab = "ADH1B")
legend("topleft", c("RD", "no RD"), pch = 19, col = c("red", "grey"), bty = "n",
    cex = 0.8)
abline(v = -20.05)
abline(a = -39.5, b = -1, lty = 2)
dev.off()
```

```
## pdf
##   2
```

4 Appendix

```
getwd()
```

```
## [1] "\\\\mdadqsfs02/workspace/kabagg/RDPaper/Webpage/ResidualDisease"
```

```
sessionInfo()
```

```
## R version 2.15.3 (2013-03-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] knitr_1.2
##
## loaded via a namespace (and not attached):
## [1] digest_0.6.3   evaluate_0.4.3 formatR_0.7    stringr_0.6.2
## [5] tools_2.15.3
```

Logistic Model of Residual Disease by FABP4 Expression Levels in the Validation Cohort

Susan L. Tucker

```
opts_chunk$set(tidy = TRUE, message = TRUE)
```

1 Executive Summary

1.1 Introduction

The goal of this analysis is to use illustrate the relationship between FABP4 level and incidence of residual disease (RD) in the validation cohort using logistic regression.

1.2 Data & Methods

We load the RData object containing the results of the validation study, including PCR measurements of FABP4 and RD status for each patient.

A logistic regression model is fitted to the data, describing RD as a function of FABP4.

Patients are grouped into 4 groups of 34-35 patients each, sorted by FABP4 values. The mean and standard deviation of FABP4 is computed for each group. The incidence of RD per group is computed and the standard deviation is estimated using binomial statistics.

The grouped data are plotted, with the fit of the logistic model shown for comparison.

1.3 Results

The incidence of RD in the 4 groups, in order of increasing FABP4, is 14/34 (41%), 22/35 (63%), 18/35 (51%) and 30/35 (86%), respectively.

1.4 Conclusion

The plot indicates a continuous trend toward increasing incidence of RD over the entire range of FABP4 values, with an estimated incidence of about 30% at the lowest values of FABP4 observed.

2 Loading & Processing Data

The data object containing the PCR values and RD information is loaded.

```
load(file.path("RDataObjects", "PCRResults.RData"))
```

The FABP4 and RD information is extracted.

```
fabp4 <- PCRResults$FABP4
RD <- rep(0, length(fabp4))
RD[PCRResults$RDStatus == "Yes"] <- 1
```

The data are sorted by increasing FABP4 values.

```
fabp4Sorted <- fabp4[order(fabp4)]
RDSorted <- RD[order(fabp4)]
```

The patients are grouped into 4 groups of 34-35 patients each.

```
numGp <- 4
```

```
nGp <- c(34, 35, 35, 35)
gp <- c(rep(1, 34), rep(2, 35), rep(3, 35), rep(4, 35))
```

3 Analyses

We compute the mean and standard deviation of FABP4 values per group. We also determine the incidence of RD per group and compute its standard deviation using binomial statistics.

```
meanPCR <- c()
sdPCR <- c()
kRD <- c()

for (i in 1:numGp) {
    meanPCR <- c(meanPCR, mean(fabp4Sorted[gp == i]))
    sdPCR <- c(sdPCR, sd(fabp4Sorted[gp == i]))
    kRD <- c(kRD, sum(RDSorted[gp == i] == 1))
}

sdRD <- sqrt(kRD * (nGp - kRD)/nGp)/(nGp)
incRD <- kRD/nGp
```

A logistic model is fitted to the data.

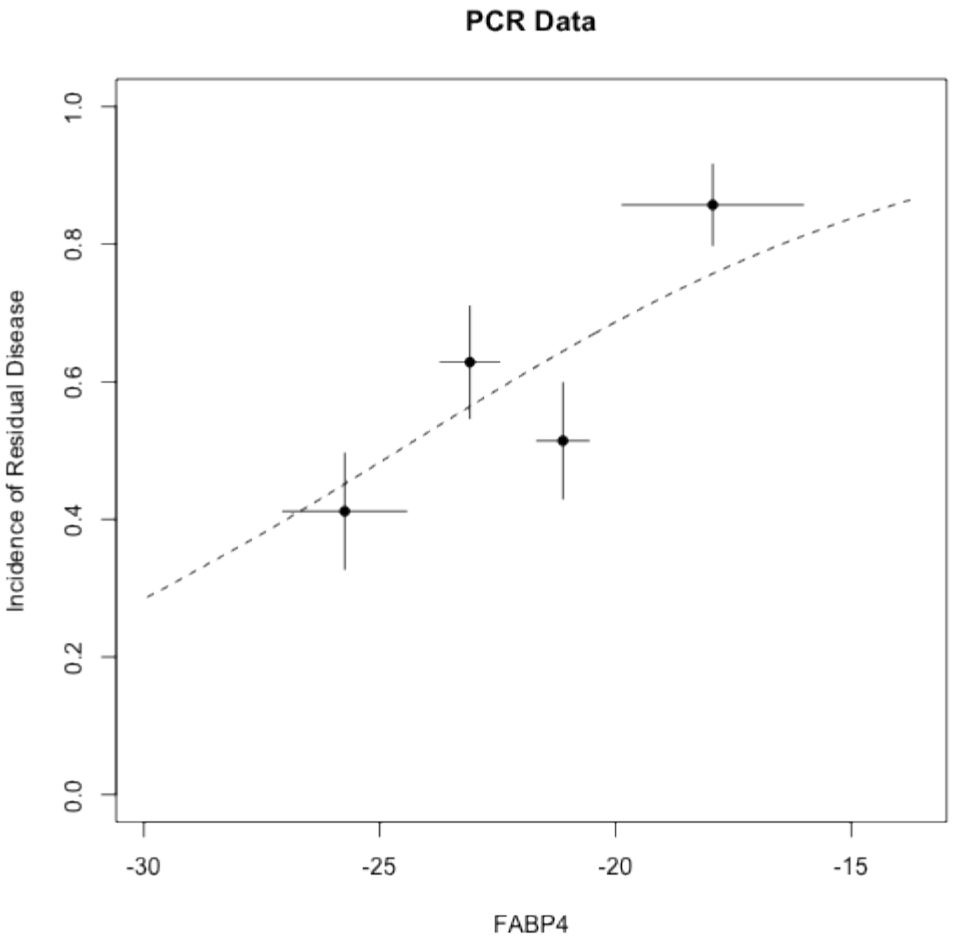
```
fitPCR <- glm(RDSorted ~ fabp4Sorted, family = "binomial")
```

A plot is produced showing incidence of RD as a function of FABP4 assayed by qRT-PCR.

The points show the observed incidence of RD in each group, plotted at the mean value of FABP4 per group. Horizontal error bars represent +/- 1 standard deviation of the FABP4 values per group. Vertical error bars represent +/- 1 standard deviation of the incidence, computed using binomial statistics.

The dashed curve shows the fit of the logistic model to the ungrouped data.

```
plot(meanPCR, incRD, pch = 16, ylim = c(0, 1), xlim = c(min(fabp4), max(fabp4)),
    xlab = "FABP4", ylab = "Incidence of Residual Disease", main = "PCR Data")

points(fabp4Sorted, fitPCR$fitted.values, type = "l", lty = 2)

for (i in 1:numGp) {
    x <- c(meanPCR[i] - sdPCR[i], meanPCR[i] + sdPCR[i])
    y <- c(incRD[i], incRD[i])
    points(x, y, type = "l", lty = 1)
}

for (i in 1:numGp) {
    x <- c(meanPCR[i], meanPCR[i])
    y <- c(incRD[i] - sdRD[i], incRD[i] + sdRD[i])
    points(x, y, type = "l", lty = 1)
}
```



4 Appendix

4.1 File Location

```
getwd()
```

```
## [1] "/Users/slt/SLT WORKSPACE/EXEMPT/OVARIAN/Ovarian residual disease study 2012/RD
manuscript/Web page for paper/Webpage"
```

4.2 SessionInfo

```
sessionInfo()
```

```
## R version 3.0.2 (2013-09-25)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] knitr_1.5
##
## loaded via a namespace (and not attached):
```

```
## [1] evaluate_0.5.1 formatR_0.10   stringr_0.6.2  tools_3.0.2
```

Correlation Between FABP4 and ADH1B Expression and Protein Levels Measured by RPPA

Susan L. Tucker

```
opts_chunk$set(tidy = TRUE, message = TRUE)
```

1 Executive Summary

1.1 Introduction

The goal of this analysis is to determine whether expression levels of FABP4 or ADH1B in the TCGA ovarian cohort are correlated with protein levels measured using reverse-phase protein arrays (RPPA).

1.2 Data & Methods

Level 3 RPPA data from ovarian samples were downloaded from the TCGA website on March 20, 2014.

Samples are identified having both gene expression data and RPPA data available.

Spearman correlation analysis is performed between FABP4 expression and protein levels for each protein represented in the RPPA data. The same analysis is performed for ADH1B expression using the probeset of interest identified in previous analyses.

Permutation analyses are performed in which the FABP4 and ADH1B values are randomly permuted among patients 100 times and correlation analyses repeated.

The distributions of P-values are investigated and compared to those obtained from the permutation tests.

A heatmap of selected proteins is produced to investigate patterns in protein levels.

Scatterplots between FABP4 expression and protein levels, and between ADH1B expression and protein levels, are produced for proteins that may be associated with expression level of either of these genes.

1.3 Results

RPPA data are available from 165 proteins in 412 samples. There are 354 patients with both RRPA data and expression data available.

Correlation coeffients range from -0.353 to 0.378. Comparison of P-values with the results of permutation analyses suggest that more proteins are correlated with FABP4 and/or ADH1B than expected by chance. The heatmap also suggests that some of the proteins investigated may be associated with differences in RD.

The scatterplots illustrate that the “significant” correlations between FABP4 and ADH1B expression and protein levels are weak, as reflected in the low correlation coefficients.

1.4 Conclusion

There appear to be some associations between FABP4 and ADH1B expression values and levels of some proteins measured using RPPA.

There is also some suggestion from the heatmap of associations between protein levels and incidence of RD.

These associations are weak in the current data set, but they warrent further investigation. They may help to elucidate biological mechanisms explaining the association between high FABP4 and ADH1B levels and significantly increased risk of RD after primary cytoreduction in high-grade serous ovarian cancer.

2 Loading & Filtration of Data

The relevant TCGA data objects (created previously) are loaded.

```
load(file.path("RDataObjects", "tcgaFilteredSamples.RData"))
load(file.path("RDataObjects", "tcgaExpression.RData"))
load("ovRPPA.RData")
```

Previously described sample filtrations are applied to the TCGA data.

```
rownames(tcgaFilteredSamples)[1:2]
```

```
## [1] "TCGA-13-0758-01A-01R-0362-01" "TCGA-09-0364-01A-02R-0362-01"
```

```
tcgaSampleUseLong <- rownames(tcgaFilteredSamples[which(tcgaFilteredSamples[,
    "sampleUse"] == "Used"), ])
tcgaSampleUse <- substr(tcgaSampleUseLong, 1, 12)
length(tcgaSampleUse)
```

```
## [1] 491
```

```
length(unique(tcgaSampleUse))
```

```
## [1] 491
```

```
tcgaRD <- tcgaSampleRD[tcgaSampleUseLong]
names(tcgaRD) <- tcgaSampleUse
```

We identify the subset of patients having both RPPA data and expression data available.

```
colnames(ovRPPA)[1:2]
```

```
## [1] "TCGA-04-1335-01A-21-20" "TCGA-04-1336-01A-21-20"
```

```
dim(ovRPPA)
```

```
## [1] 165 412
```

```
length(unique(substr(colnames(ovRPPA), 1, 12)))
```

```
## [1] 412
```

```
colnames(ovRPPA) <- substr(colnames(ovRPPA), 1, 12)

ptList <- intersect(colnames(ovRPPA), tcgaSampleUse)
length(ptList)
```

```
## [1] 354
```

```
rppaUse <- ovRPPA[, ptList]
rdUse <- tcgaRD[ptList]
```

We extract the FABP4 and ADH1B expression values from the expression matrix. We know which probesets these genes correspond to from earlier analyses.

```
colnames(tcgaExpression[, 1:2])
```

```
## [1] "TCGA-13-0758-01A-01R-0362-01" "TCGA-09-0364-01A-02R-0362-01"
```

```
tcgaExpressionUse <- tcgaExpression[, tcgaSampleUseLong]
colnames(tcgaExpressionUse) <- tcgaSampleUse

fabp4 <- tcgaExpressionUse["203980_at", ptList]
adh1b <- tcgaExpressionUse["209613_s_at", ptList]
```

3 Analyses

3.1 Correlation analyses

We perform Spearman correlation analyses between FABP4 expression levels and RPPA values.

```
fabp4CorP <- apply(rppaUse, 1, function(x) {
    cor.test(fabp4, x, method = "spearman", exact = FALSE)$p.value
})

fabp4CorRho <- apply(rppaUse, 1, function(x) {
    cor.test(fabp4, x, method = "spearman", exact = FALSE)$estimate
})

summary(fabp4CorP)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0112  0.0932  0.2360  0.4130  0.9670
```

```
summary(fabp4CorRho)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.3530 -0.0882 -0.0121  0.0017  0.0895  0.3680
```

We do the same for ADH1B.

```
adh1bCorP <- apply(rppaUse, 1, function(x) {
    cor.test(adh1b, x, method = "spearman", exact = FALSE)$p.value
})

adh1bCorRho <- apply(rppaUse, 1, function(x) {
    cor.test(adh1b, x, method = "spearman", exact = FALSE)$estimate
})

summary(adh1bCorP)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0033  0.0979  0.2400  0.4020  0.9900
```

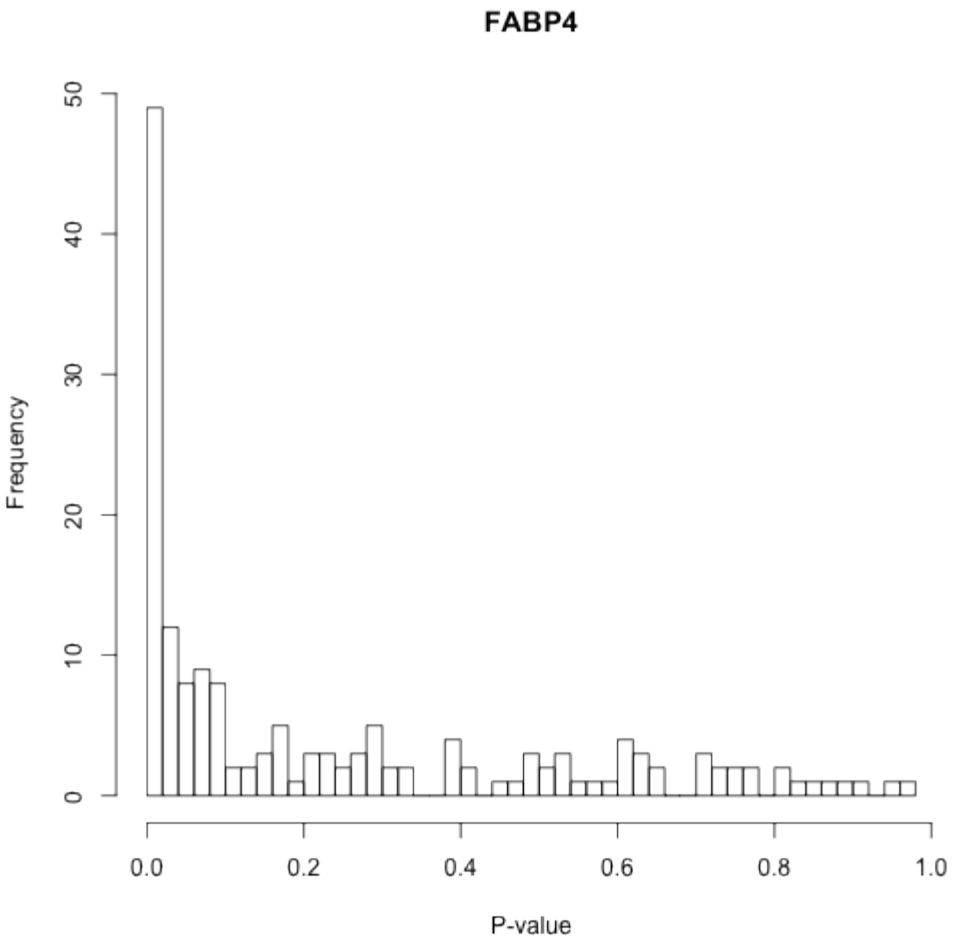
```
summary(adh1bCorRho)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.3270 -0.0881 -0.0014 -0.0005  0.0871  0.3780
```

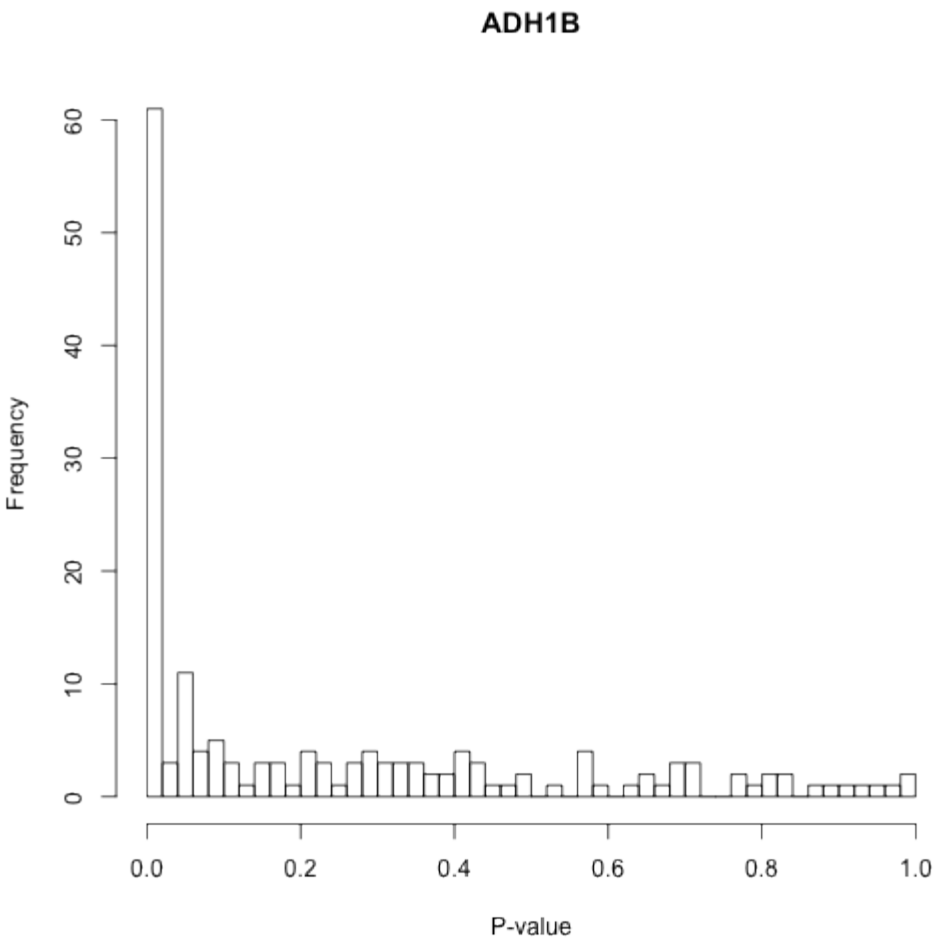
3.2 Distributions of P-values from the correlation analyses

We look at the histogram of P-values (BUM plots) obtained for each analysis. The distributions are markedly non-uniform, suggesting that more proteins are significantly associated with FABP4 and ADH1B than would be expected by chance.

```
hist(fabp4CorP, breaks = 50, xlab = "P-value", main = "FABP4")
```




```
hist(adh1bCorP, breaks = 50, xlab = "P-value", main = "ADH1B")
```



3.3 Permutation tests

We wish to confirm that we see more small P-values than expected by chance. Therefore, we perform a permutation test in which randomly shuffled FABP4 and ADH1B values are compared to the RPPA data using correlation analysis.

```
nPerms <- 100
proteinList <- rownames(rppaUse)

if (file.exists("rppaPerm.RData")) {
    load("rppaPerm.RData")
} else {

    starttimePerm <- date()
    set.seed(22)

    pMxPerm <- matrix(0, nrow = length(proteinList), ncol = nPerms)
    rownames(pMxPerm) <- proteinList
    rhoMxPerm <- pMxPerm
    pMxPerm2 <- pMxPerm
    rhoMxPerm2 <- pMxPerm

    nPt <- length(ptList)

    for (i1 in 1:nPerms) {

        fabp4Fake <- fabp4[sample(nPt)]
        adh1bFake <- adh1b[sample(nPt)]

        fakeP <- apply(rppaUse, 1, function(x) {
            cor.test(fabp4Fake, x, method = "spearman", exact = FALSE)$p.value
```

```
        })
        fakeRho <- apply(rppaUse, 1, function(x) {
            cor.test(fabp4Fake, x, method = "spearman", exact = FALSE)$estimate
        })

        fakeP2 <- apply(rppaUse, 1, function(x) {
            cor.test(adh1bFake, x, method = "spearman", exact = FALSE)$p.value
        })
        fakeRho2 <- apply(rppaUse, 1, function(x) {
            cor.test(adh1bFake, x, method = "spearman", exact = FALSE)$estimate
        })

        pMxPerm[, i1] <- fakeP
        rhoMxPerm[, i1] <- fakeRho

        pMxPerm2[, i1] <- fakeP2
        rhoMxPerm2[, i1] <- fakeRho2

    }

    stoptimePerm <- date()

    save(pMxPerm, rhoMxPerm, pMxPerm2, rhoMxPerm2, starttimePerm, stoptimePerm,
        file = "rppaPerm.RData")
}

c(starttimePerm, stoptimePerm)
```

```
## [1] "Fri Mar 21 10:39:09 2014" "Fri Mar 21 10:39:47 2014"
```

3.4 Comparing results to those of permutation tests

For each of several significance levels, we plot the numbers of proteins significantly associated with FABP4 and compare them to the numbers expected by chance.

The findings indicate that 23 proteins are correlated with FABP4 at a significance level of P < 0.001, while at most one would be expected by chance.

```
min(fabp4CorP)
```

```
## [1] 8.282e-13
```

```
pList <- c(1e-11, 1e-10, 1e-09, 1e-08, 1e-07, 1e-06, 1e-05, 1e-04, 0.001, 0.01)
numTrue <- unlist(lapply(pList, function(x) {
    sum(fabp4CorP < x)
}))

meanPerm <- c()
sdPerm <- c()
upPerm <- c()
downPerm <- c()

for (i in 1:length(pList)) {

    numPerm <- apply(pMxPerm, 2, function(x) {
        sum(x < pList[i])
    })
    meanPerm <- c(meanPerm, mean(numPerm))
    sdPerm <- c(sdPerm, sd(numPerm))

    upPerm <- c(upPerm, quantile(numPerm, c(0.95)))
    downPerm <- c(downPerm, quantile(numPerm, c(0.05)))

}
```

```
numTrue
```

```
##  [1]  3  3  3  3  4  5 12 14 23 39
```

```
meanPerm
```

```
##  [1] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.09 1.28
```

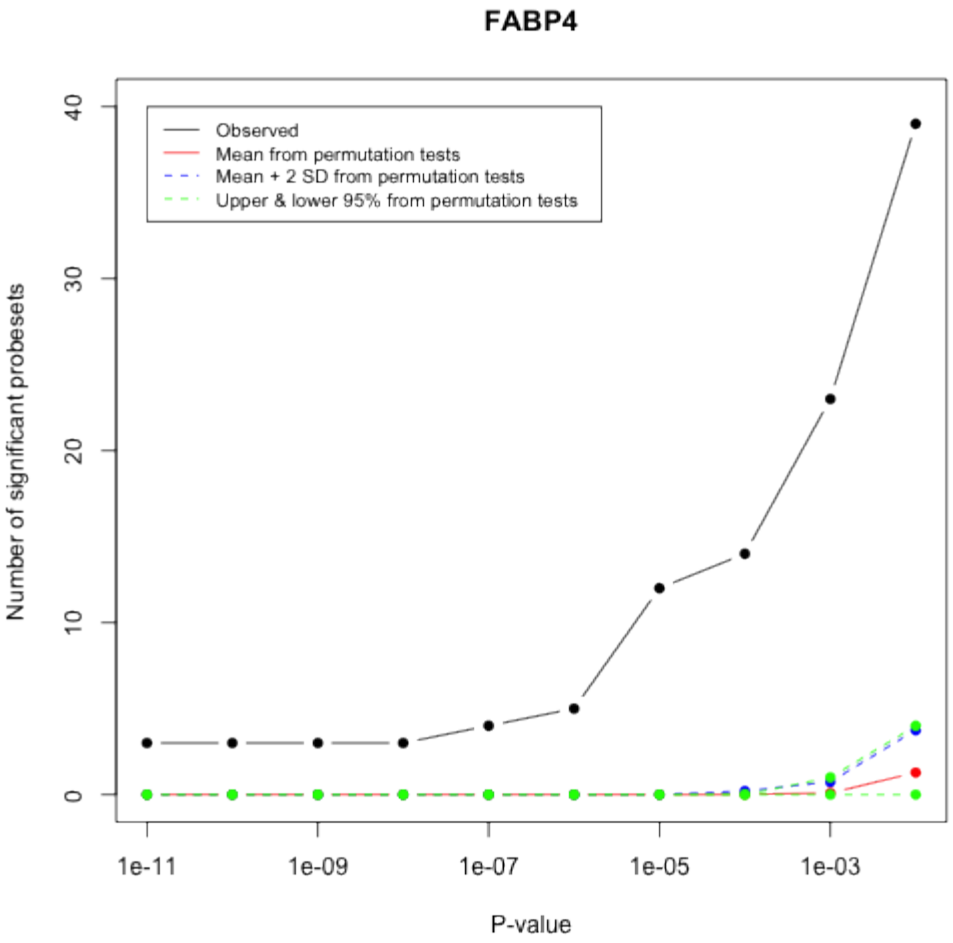
```
meanPerm + 2 * sdPerm
```

```
##  [1] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.2100 0.7317 3.7431
```

```
upPerm
```

```
## 95% 95% 95% 95% 95% 95% 95% 95% 95% 95%
##   0   0   0   0   0   0   0   0   1   4
```

```
plot(pList, numTrue, type = "b", xlab = "P-value", ylab = "Number of significant probesets",
    pch = 16, ylim = c(0, 40), main = "FABP4", log = "x")

lines(pList, meanPerm, type = "b", col = "red", pch = 16)
lines(pList, meanPerm + 2 * sdPerm, type = "b", col = "blue", lty = 2, pch = 16)

lines(pList, upPerm, type = "b", col = "green", lty = 2, pch = 16)
lines(pList, downPerm, type = "b", col = "green", lty = 2, pch = 16)

legend(1e-11, 40, legend = c("Observed", "Mean from permutation tests", "Mean + 2 SD from
permutation tests",
    "Upper & lower 95% from permutation tests"), col = c("black", "red", "blue",
    "green"), lty = c(1, 1, 2, 2), cex = 0.8)
```



We do the same thing for ADH1B and find that there are 30 proteins correlated with ADH1B at a signficance level of P < 0.001, while at most one would be expected by chance.

```
min(adh1bCorP)
```

```
## [1] 1.87e-13
```

```
numTrue2 <- unlist(lapply(pList, function(x) {
    sum(adh1bCorP < x)
}))

meanPerm2 <- c()
sdPerm2 <- c()
upPerm2 <- c()
downPerm2 <- c()

for (i in 1:length(pList)) {

    numPerm2 <- apply(pMxPerm2, 2, function(x) {
        sum(x < pList[i])
    })
    meanPerm2 <- c(meanPerm2, mean(numPerm2))
    sdPerm2 <- c(sdPerm2, sd(numPerm2))

    upPerm2 <- c(upPerm2, quantile(numPerm2, c(0.95)))
    downPerm2 <- c(downPerm2, quantile(numPerm2, c(0.05)))

}

numTrue2
```

```
##  [1]  2  2  3  4  5  7 13 20 30 53
```

```
meanPerm2
```

```
##  [1] 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.08 1.37
```

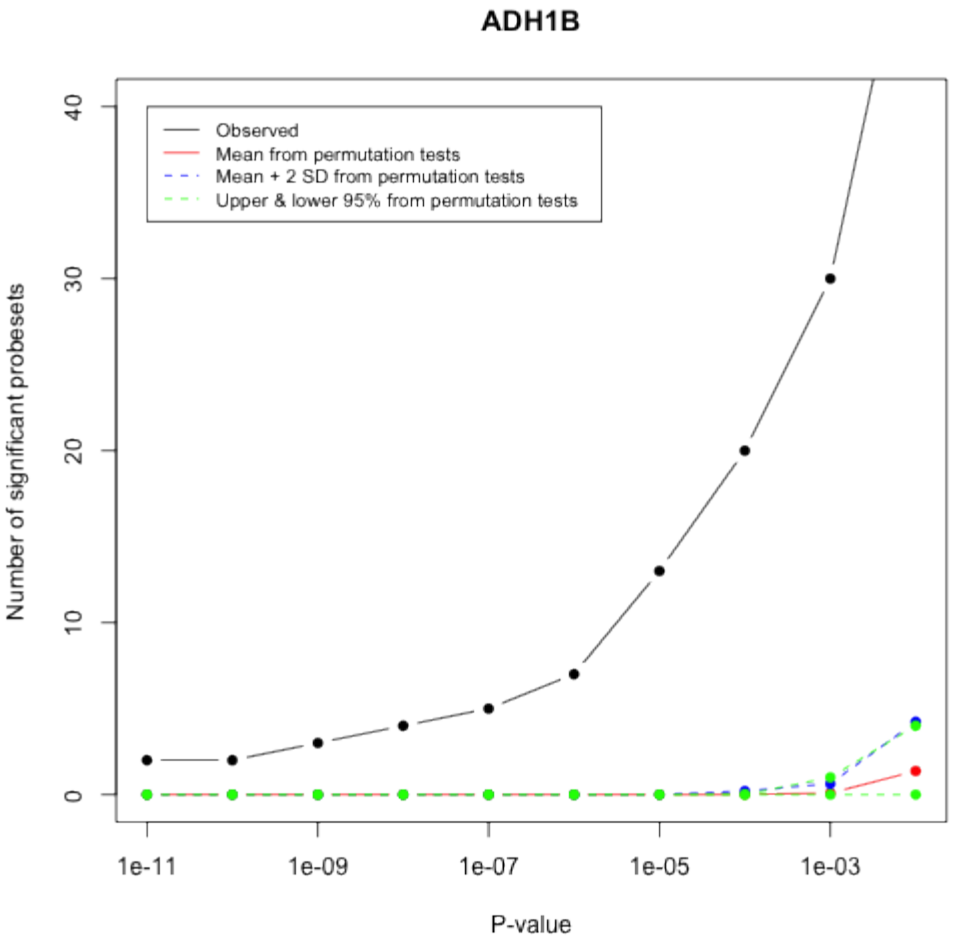
```
meanPerm2 + 2 * sdPerm2
```

```
##  [1] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.2100 0.6253 4.2361
```

```
upPerm2
```

```
## 95% 95% 95% 95% 95% 95% 95% 95% 95% 95%
##   0   0   0   0   0   0   0   0   1   4
```

```
plot(pList, numTrue2, type = "b", xlab = "P-value", ylab = "Number of significant probesets",
    pch = 16, ylim = c(0, 40), main = "ADH1B", log = "x")

lines(pList, meanPerm2, type = "b", col = "red", pch = 16)
lines(pList, meanPerm2 + 2 * sdPerm2, type = "b", col = "blue", lty = 2, pch = 16)

lines(pList, upPerm2, type = "b", col = "green", lty = 2, pch = 16)
lines(pList, downPerm2, type = "b", col = "green", lty = 2, pch = 16)

legend(1e-11, 40, legend = c("Observed", "Mean from permutation tests", "Mean + 2 SD from
permutation tests",
    "Upper & lower 95% from permutation tests"), col = c("black", "red", "blue",
    "green"), lty = c(1, 1, 2, 2), cex = 0.8)
```



We identify the 23 proteins correlated with FABP4 at a signficance level of P < 0.001 as well as their correlation coefficients, first for the proteins having a positive correlation with FABP4 (N=14) and next for the ones having a negative corelation (N=9).

```
fabp4Proteins <- names(fabp4CorP)[fabp4CorP < 0.001]
fabp4ProteinsPos <- names(fabp4CorP)[fabp4CorP < 0.001 & fabp4CorRho > 0]
fabp4ProteinsNeg <- names(fabp4CorP)[fabp4CorP < 0.001 & fabp4CorRho < 0]
length(fabp4ProteinsPos)
```

```
## [1] 14
```

```
length(fabp4ProteinsNeg)
```

```
## [1] 9
```

```
fabp4CorRho[fabp4ProteinsPos]
```

```
##        Caveolin-1-R-V            c-Kit-R-V       Collagen_VI-R-V
##                0.3682               0.1751                0.1974
##              Dvl3-R-V       Fibronectin-R-C FOXO3a_pS318_S321-R-C
##                0.2312               0.3643                0.2431
##              JNK2-R-C              p21-R-C          Paxillin-R-V
##                0.1943               0.1994                0.2395
##            Pea-15-R-V         PKC-alpha-M-V              PTCH-R-C
##                0.1824               0.1780                0.2377
##  Transglutaminase-M-V             VASP-R-C
##                0.1993               0.2507
```

```
fabp4CorRho[fabp4ProteinsNeg]
```

```
##     Chk1_pS345-R-C      Chk2_pT68-R-C      Claudin-7-R-V
##            -0.2948            -0.3528            -0.2019
##   c-Met_pY1235-R-C    C-Raf_pS338-R-C     E-Cadherin-R-V
##            -0.2775            -0.2266            -0.2464
## ER-alpha_pS118-R-V      p27_pT157-R-C      Src_pY416-R-C
##            -0.2409            -0.2399            -0.1795
```

We do the same for ADH1B. There are 13 proteins positively associated with ADH1B at a significance level P < 0.001, and 17 that are negatively associated.

```
adh1bProteins <- names(adh1bCorP)[adh1bCorP < 0.001]
adh1bProteinsPos <- names(adh1bCorP)[adh1bCorP < 0.001 & adh1bCorRho > 0]
adh1bProteinsNeg <- names(adh1bCorP)[adh1bCorP < 0.001 & adh1bCorRho < 0]
length(adh1bProteinsPos)
```

```
## [1] 13
```

```
length(adh1bProteinsNeg)
```

```
## [1] 17
```

```
adh1bCorRho[adh1bProteinsPos]
```

```
##         Annexin_I-R-V        Caveolin-1-R-V              Dvl3-R-V
##                0.1978                0.3744               0.1812
##       Fibronectin-R-C FOXO3a_pS318_S321-R-C          Paxillin-R-V
##                0.3778                0.2886               0.2040
##            Pea-15-R-V             PTCH-R-C              PTEN-R-V
##                0.1951                0.2600               0.2100
##     S6_pS235_S236-R-V     S6_pS240_S244-R-V  Transglutaminase-M-V
##                0.2646                0.2208               0.2094
##              VASP-R-C
##                0.2470
```

```
adh1bCorRho[adh1bProteinsNeg]
```

```
##          53BP1-R-C  alpha-Catenin-M-V   beta-Catenin-R-V
##            -0.1976            -0.2194            -0.2149
##           CD31-M-V     Chk1_pS345-R-C      Chk2_pT68-R-C
##            -0.1934            -0.2206            -0.3268
##      Claudin-7-R-V   c-Met_pY1235-R-C    C-Raf_pS338-R-C
##            -0.2077            -0.2454            -0.1907
##     E-Cadherin-R-V ER-alpha_pS118-R-V          Ku80-R-C
##            -0.3144            -0.2485            -0.2031
##           MSH2-M-C           MSH6-R-C           mTOR-R-V
##            -0.2479            -0.2365            -0.1792
##      p27_pT157-R-C   PARP_cleaved-M-C
##            -0.2436            -0.1766
```

We verify that proteins correlated with both FABP4 and ADH1B at a significance level of P < 0.001 have the same sign of the correlation coefficient in each case.

```
bothProteins <- intersect(fabp4Proteins, adh1bProteins)
table(sign(fabp4CorRho[bothProteins]) == sign(adh1bCorRho[bothProteins]))
```

```
##
## TRUE
```

```
##   17
```

The proteins that are positively correlated with both FABP4 and ADH1B expression are as follows.

```
unique(intersect(fabp4ProteinsPos, adh1bProteinsPos))
```

```
## [1] "Caveolin-1-R-V"        "Dvl3-R-V"              "Fibronectin-R-C"
## [4] "FOXO3a_pS318_S321-R-C" "Paxillin-R-V"          "Pea-15-R-V"
## [7] "PTCH-R-C"              "Transglutaminase-M-V"  "VASP-R-C"
```

The proteins that are negatively correlated with both FABP4 and ADH1B are as follows.

```
unique(intersect(fabp4ProteinsNeg, adh1bProteinsNeg))
```

```
## [1] "Chk1_pS345-R-C"     "Chk2_pT68-R-C"      "Claudin-7-R-V"
## [4] "c-Met_pY1235-R-C"   "C-Raf_pS338-R-C"    "E-Cadherin-R-V"
## [7] "ER-alpha_pS118-R-V" "p27_pT157-R-C"
```

3.5 Heatmap

The following heatmap shows the levels of the 36 proteins associated with FABP4 and/or ADH1B at P < 0.001. The color bar at the top represents patients with and without RD in red and blue, respectively. The color bar at the side shows proteins positively and negatively correlated with FABP4 and/or ADH1B in red and blue, respectively.

```
library(gplots)
```

```
## KernSmooth 2.23 loaded
## Copyright M. P. Wand 1997-2009
##
## Attaching package: 'gplots'
##
## The following object is masked from 'package:stats':
##
##     lowess
```

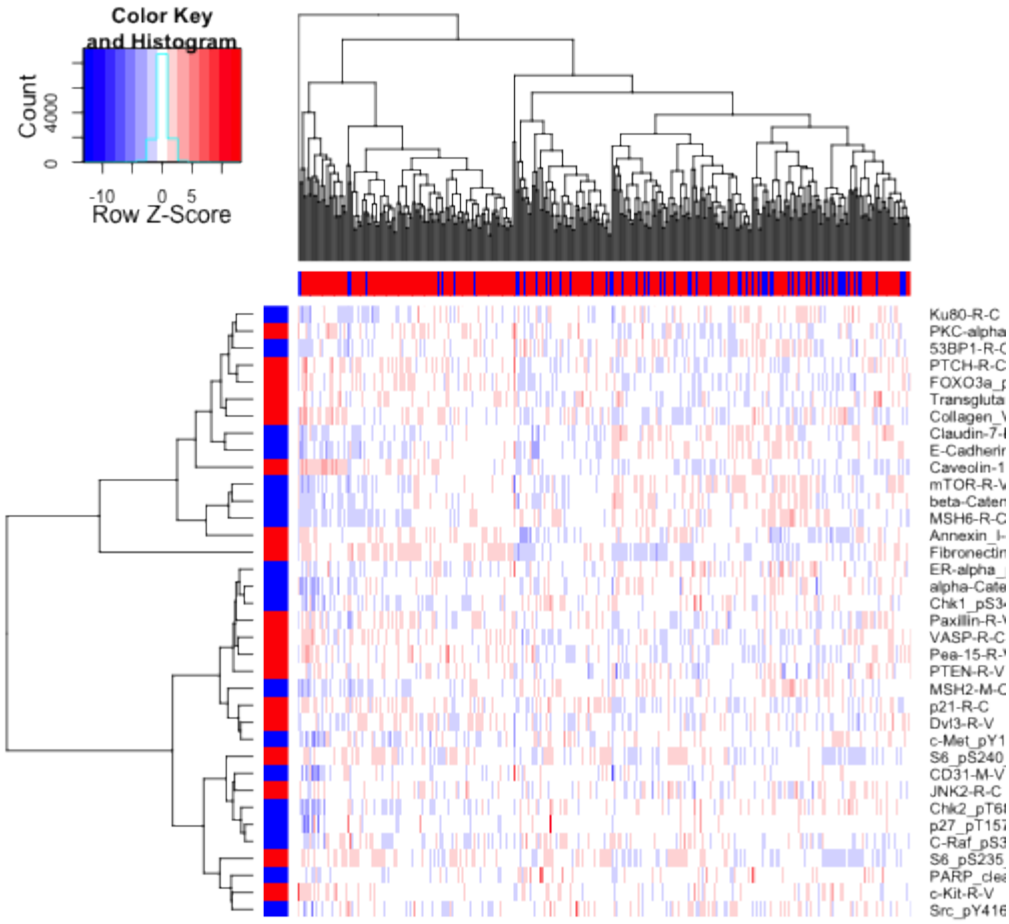
```
table(rdUse)
```

```
## rdUse
## No RD    RD
##    81   273
```

```
colVec <- rep("blue", length(ptList))
colVec[rdUse == "RD"] <- "red"

allProteins <- unique(c(fabp4Proteins, adh1bProteins))
length(allProteins)
```

```
## [1] 36
```

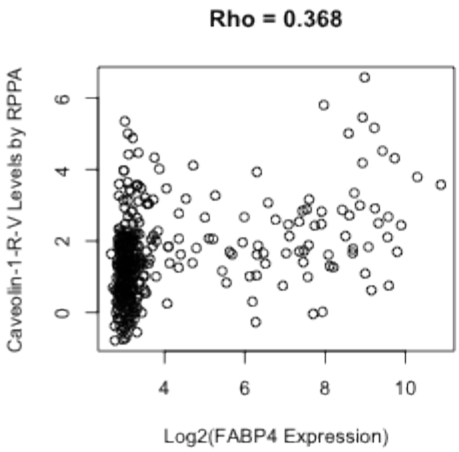
```
rowVec <- rep("blue", length(allProteins))
rowVec[sign(fabp4CorRho[allProteins]) > 0] <- "red"

heatmap.2(rppaUse[allProteins, ], scale = "row", trace = "none", labRow = allProteins,
    labCol = "", col = bluered, ColSideColors = colVec, RowSideColors = rowVec,
    xlab = "", cexRow = 1)
```

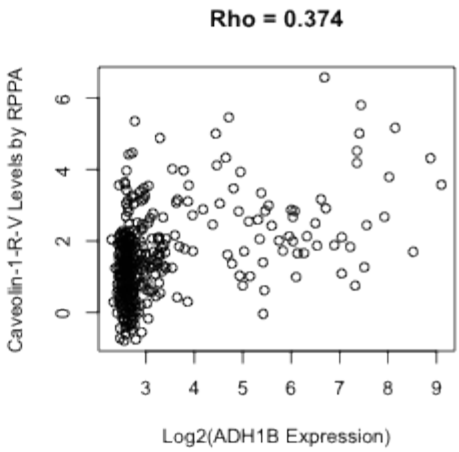


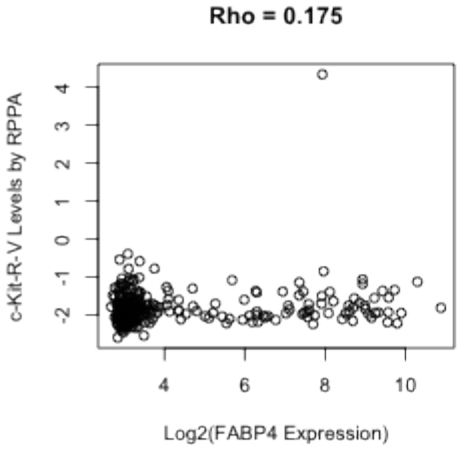
3.6 Scatterplots

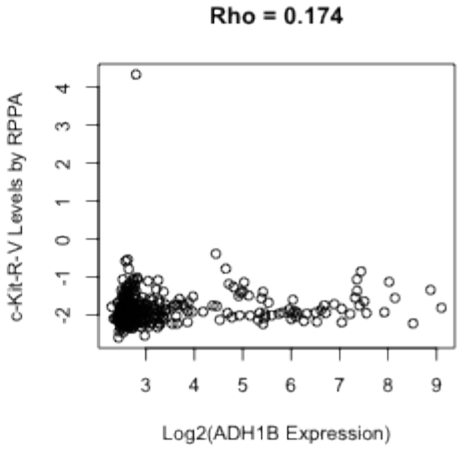
Scatterplots are produced showing the relationship between FABP4 and ADH1B expression levels with each of the 36 proteins whose levels are significantly correlated with expression levels of one or both genes at P < 0.001. We begin with the 18 positively associated proteins, then show the plots for the 12 negatively associated proteins. Spearman correlation coefficients (Rho) are shown above each plot.

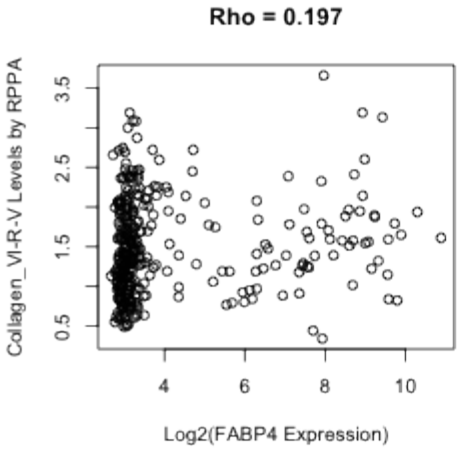
```
proteinsPos <- unique(c(fabp4ProteinsPos, adh1bProteinsPos))
proteinsNeg <- unique(c(fabp4ProteinsNeg, adh1bProteinsNeg))
proteinList <- c(proteinsPos, proteinsNeg)

par(mfrow = c(2, 2))

for (i1 in 1:length(proteinList)) {

    rho1 = round(1000 * fabp4CorRho[proteinList[i1]])/1000
    rho2 = round(1000 * adh1bCorRho[proteinList[i1]])/1000

    plot(fabp4, rppaUse[proteinList[i1], ], xlab = "Log2(FABP4 Expression)",
        ylab = paste(proteinList[i1], " Levels by RPPA", sep = ""), main = paste("Rho = ",
            rho1, sep = ""))

    plot(adh1b, rppaUse[proteinList[i1], ], xlab = "Log2(ADH1B Expression)",
        ylab = paste(proteinList[i1], " Levels by RPPA", sep = ""), main = paste("Rho = ",
            rho2, sep = ""))

}
```

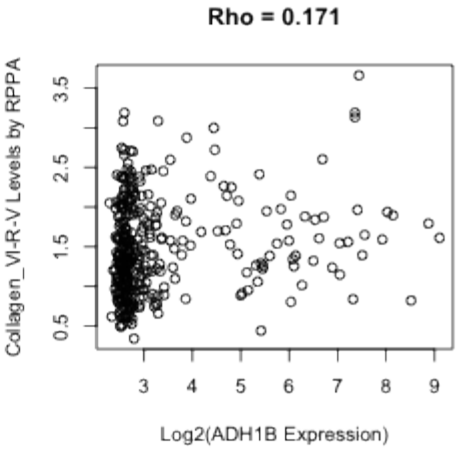

Rho = 0.368
Caveolin-1-R-V Levels by RPPA
Log2(FABP4 Expression)

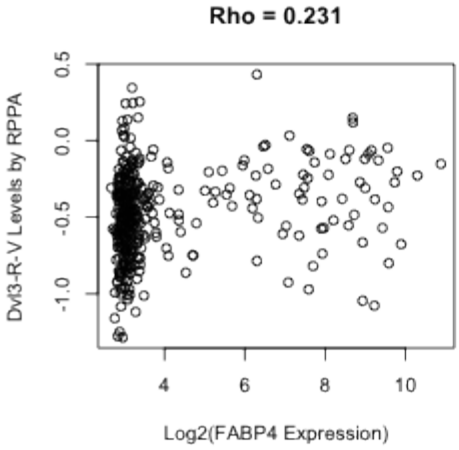

Rho = 0.374
Caveolin-1-R-V Levels by RPPA
Log2(ADH1B Expression)

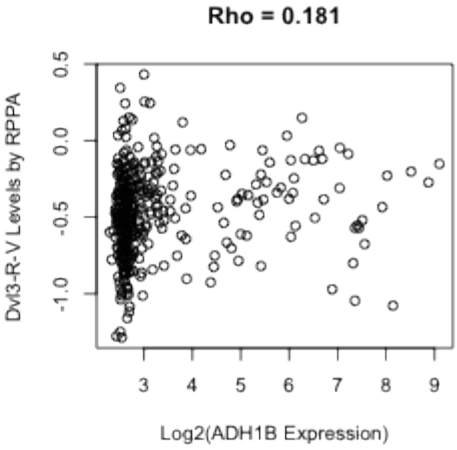

Rho = 0.175
c-Kit-R-V Levels by RPPA
Log2(FABP4 Expression)

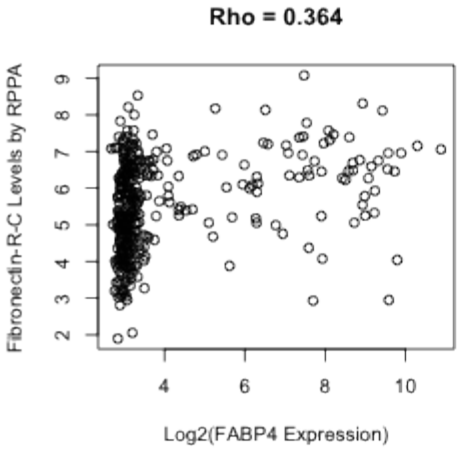

Rho = 0.174
c-Kit-R-V Levels by RPPA
Log2(ADH1B Expression)

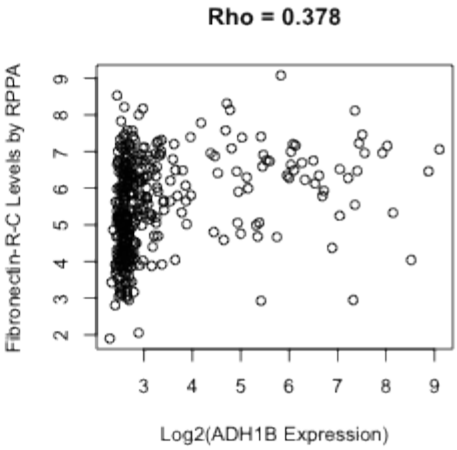

Rho = 0.197
Collagen_VI-R-V Levels by RPPA
0.5
1.5
2.5
3.5
4
6
8
10
Log2(FABP4 Expression)

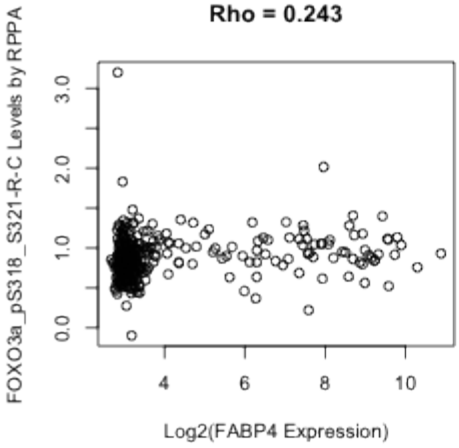

Rho = 0.171
Collagen_VI-R-V Levels by RPPA
0.5
1.5
2.5
3.5
3
4
5
6
7
8
9
Log2(ADH1B Expression)

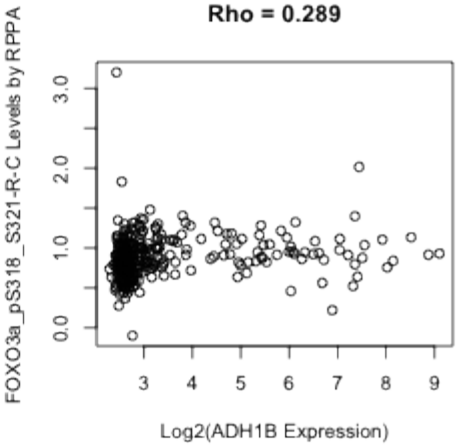

Rho = 0.231
Dvl3-R-V Levels by RPPA
-1.0
-0.5
0.0
0.5
4
6
8
10
Log2(FABP4 Expression)

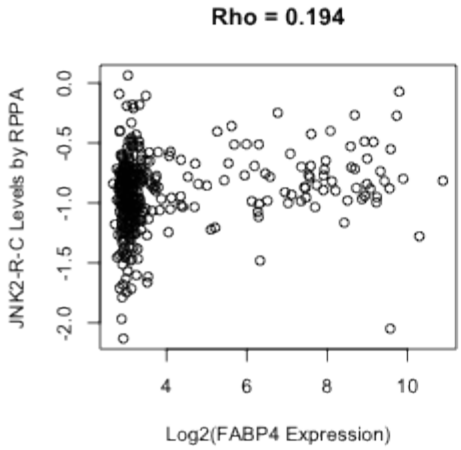

Rho = 0.181
Dvl3-R-V Levels by RPPA
-1.0
-0.5
0.0
0.5
3
4
5
6
7
8
9
Log2(ADH1B Expression)

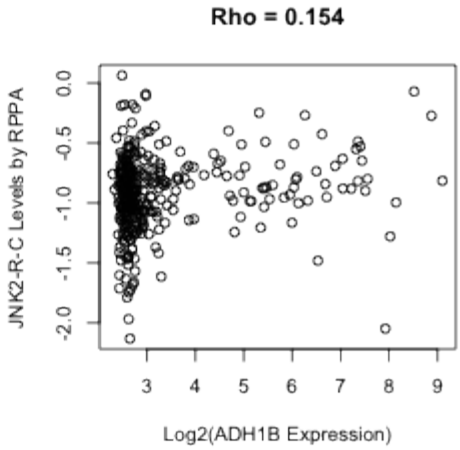

Rho = 0.364
Fibronectin-R-C Levels by RPPA
Log2(FABP4 Expression)

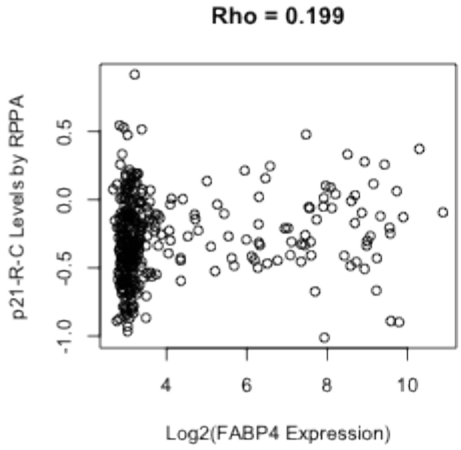

Rho = 0.378
Fibronectin-R-C Levels by RPPA
Log2(ADH1B Expression)

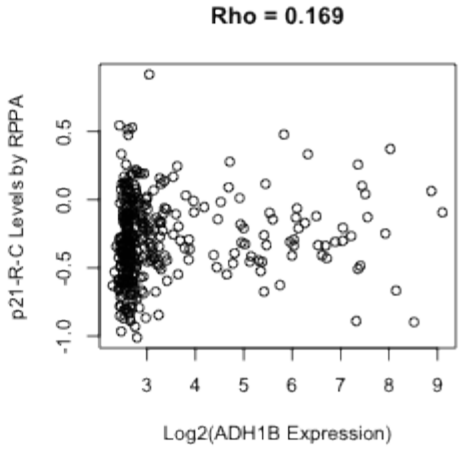

Rho = 0.243
FOXO3a_pS318_S321-R-C Levels by RPPA
Log2(FABP4 Expression)

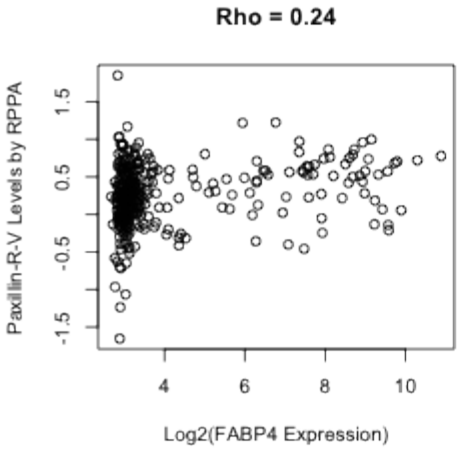

Rho = 0.289
FOXO3a_pS318_S321-R-C Levels by RPPA
Log2(ADH1B Expression)

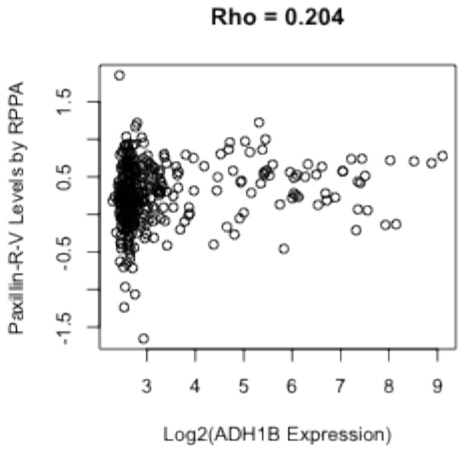

Rho = 0.194
JNK2-R-C Levels by RPPA
0.0
-0.5
-1.0
-1.5
-2.0
4
6
8
10
Log2(FABP4 Expression)

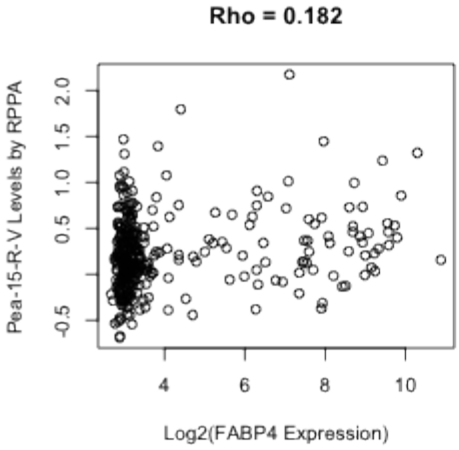

Rho = 0.154
JNK2-R-C Levels by RPPA
0.0
-0.5
-1.0
-1.5
-2.0
3
4
5
6
7
8
9
Log2(ADH1B Expression)

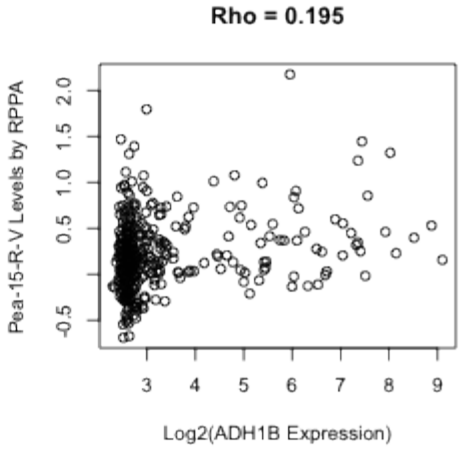

Rho = 0.199
p21-R-C Levels by RPPA
0.5
0.0
-0.5
-1.0
4
6
8
10
Log2(FABP4 Expression)

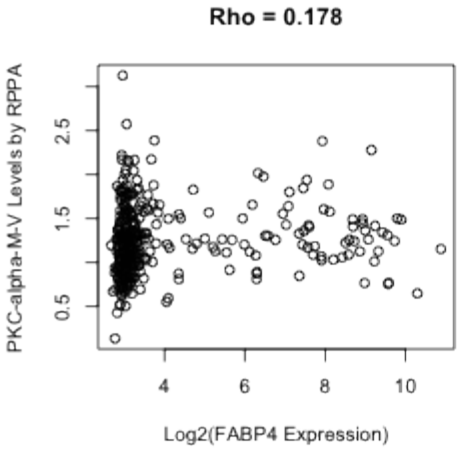

Rho = 0.169
p21-R-C Levels by RPPA
0.5
0.0
-0.5
-1.0
3
4
5
6
7
8
9
Log2(ADH1B Expression)

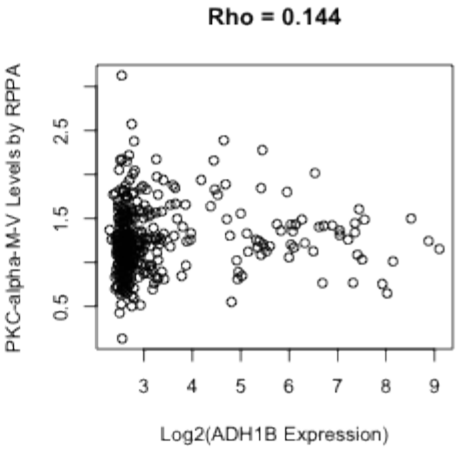

Rho = 0.24
Paxillin-R-V Levels by RPPA
1.5
0.5
-0.5
-1.5
4
6
8
10
Log2(FABP4 Expression)

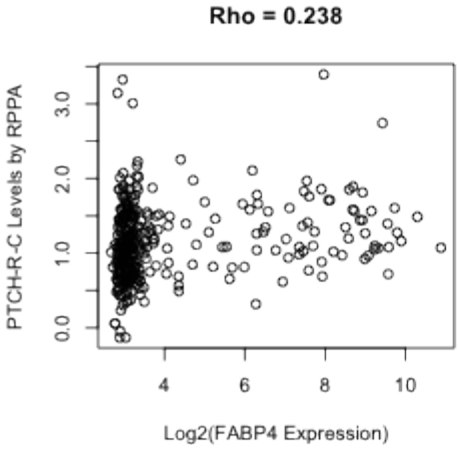

Rho = 0.204
Paxillin-R-V Levels by RPPA
1.5
0.5
-0.5
-1.5
3
4
5
6
7
8
9
Log2(ADH1B Expression)

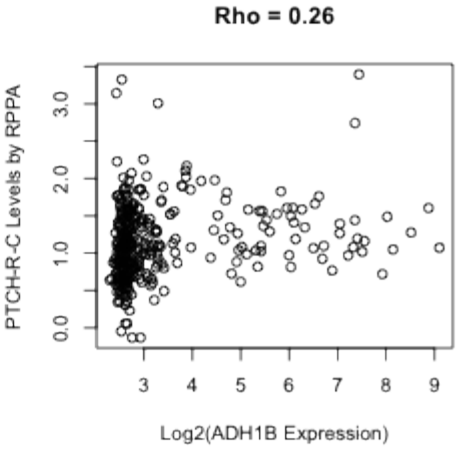

Rho = 0.182
Pea-15-R-V Levels by RPPA
2.0
1.5
1.0
0.5
-0.5
4
6
8
10
Log2(FABP4 Expression)

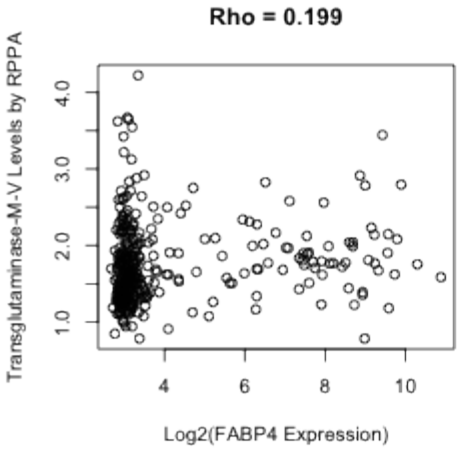

Rho = 0.195
Pea-15-R-V Levels by RPPA
2.0
1.5
1.0
0.5
-0.5
3
4
5
6
7
8
9
Log2(ADH1B Expression)

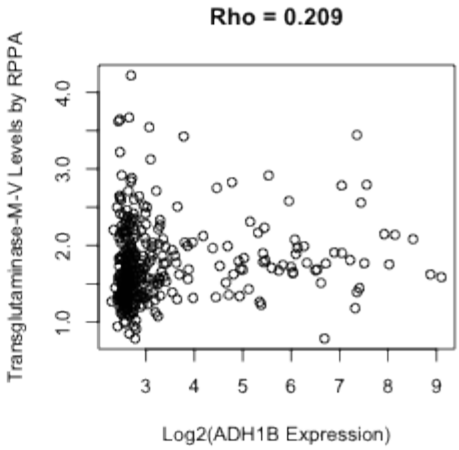

Rho = 0.178
PKC-alpha-M-V Levels by RPPA
0.5
1.5
2.5
4
6
8
10
Log2(FABP4 Expression)

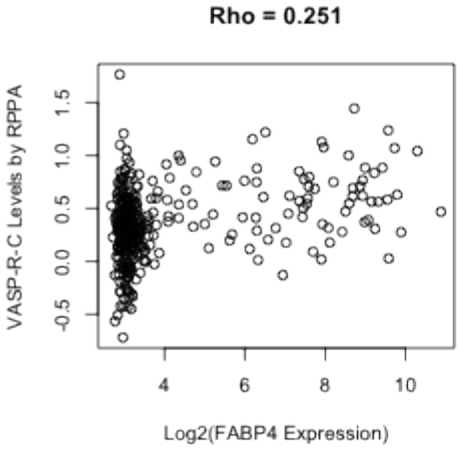

Rho = 0.144
PKC-alpha-M-V Levels by RPPA
0.5
1.5
2.5
3
4
5
6
7
8
9
Log2(ADH1B Expression)

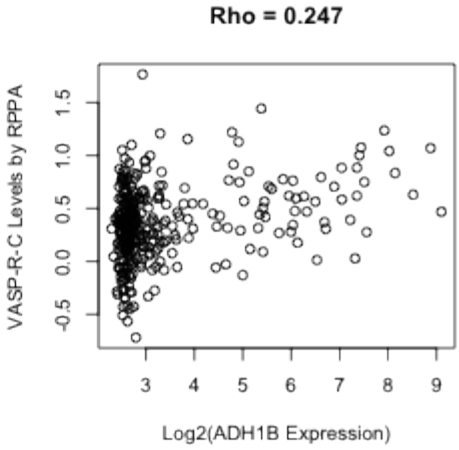

Rho = 0.238
PTCH-R-C Levels by RPPA
0.0
1.0
2.0
3.0
4
6
8
10
Log2(FABP4 Expression)

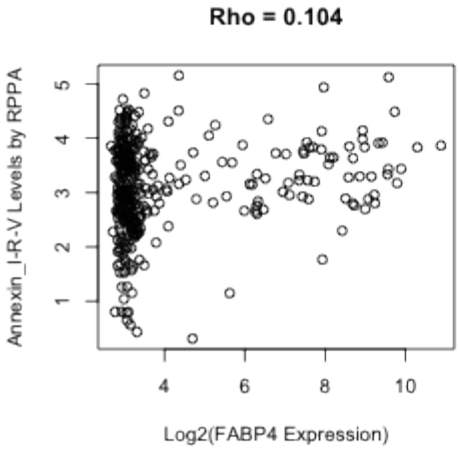

Rho = 0.26
PTCH-R-C Levels by RPPA
0.0
1.0
2.0
3.0
3
4
5
6
7
8
9
Log2(ADH1B Expression)

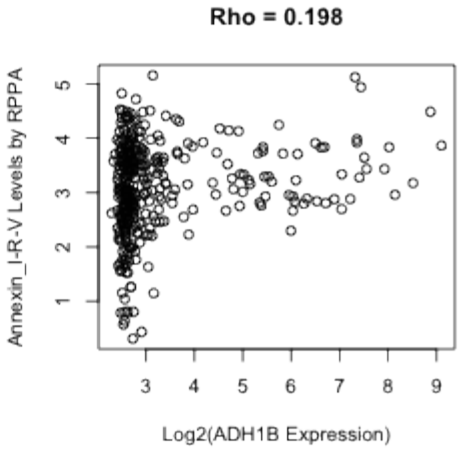

Rho = 0.199
Transglutaminase-M-V Levels by RPPA
1.0
2.0
3.0
4.0
4
6
8
10
Log2(FABP4 Expression)

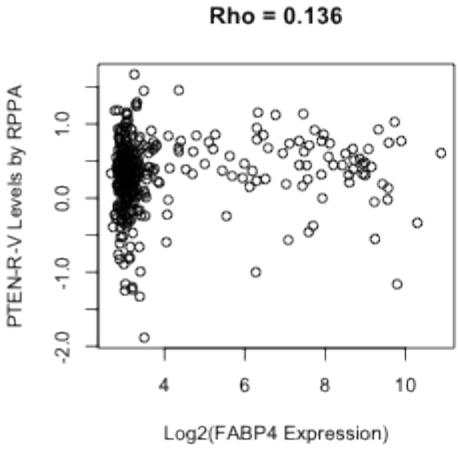

Rho = 0.209
Transglutaminase-M-V Levels by RPPA
1.0
2.0
3.0
4.0
3
4
5
6
7
8
9
Log2(ADH1B Expression)

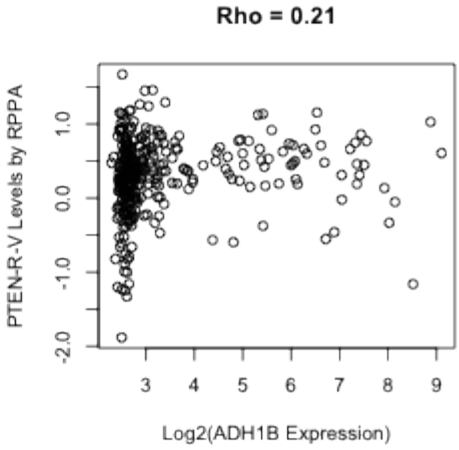

Rho = 0.251
VASP-R-C Levels by RPPA
-0.5
0.0
0.5
1.0
1.5
4
6
8
10
Log2(FABP4 Expression)

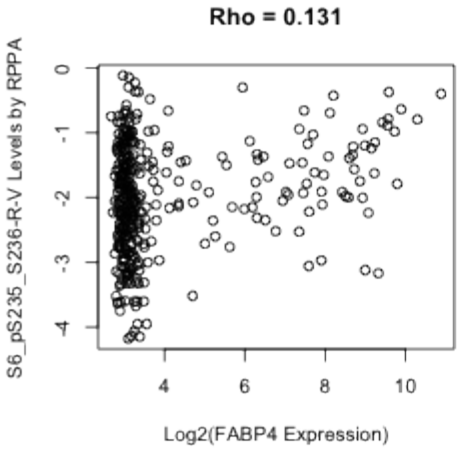

Rho = 0.247
VASP-R-C Levels by RPPA
-0.5
0.0
0.5
1.0
1.5
3
4
5
6
7
8
9
Log2(ADH1B Expression)

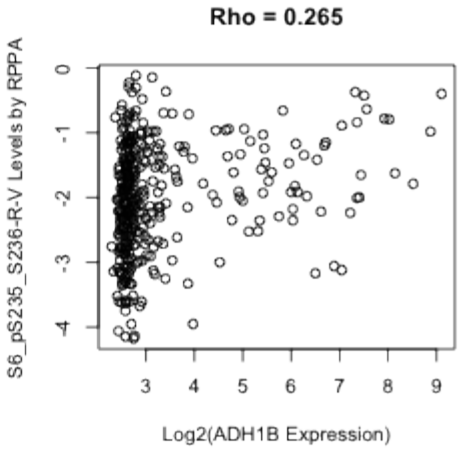

Rho = 0.104
Annexin_I-R-V Levels by RPPA
Log2(FABP4 Expression)

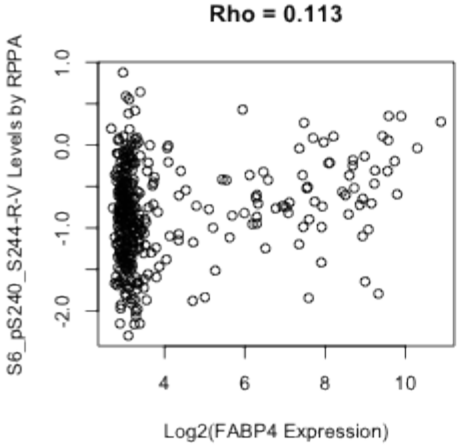

Rho = 0.198
Annexin_I-R-V Levels by RPPA
Log2(ADH1B Expression)

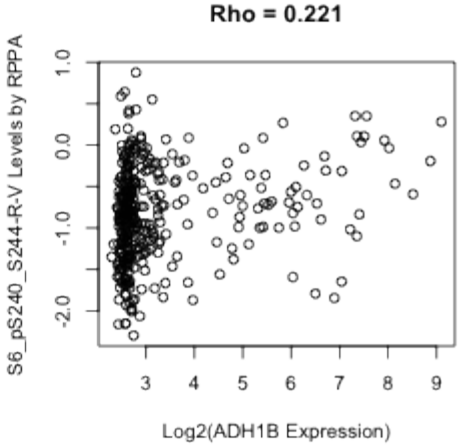

Rho = 0.136
PTEN-R-V Levels by RPPA
Log2(FABP4 Expression)

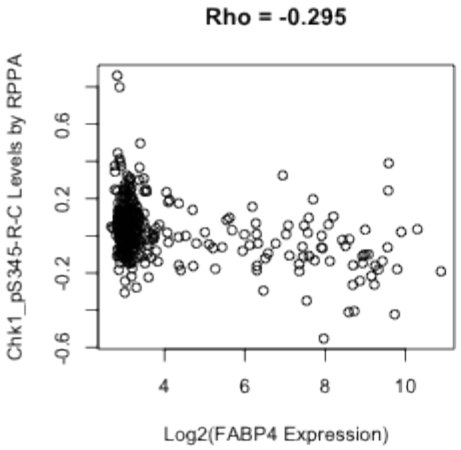

Rho = 0.21
PTEN-R-V Levels by RPPA
Log2(ADH1B Expression)

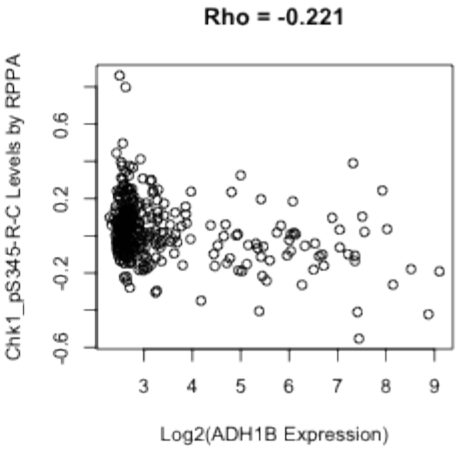

Rho = 0.131
S6_pS235_S236-R-V Levels by RPPA
0
-1
-2
-3
-4
4
6
8
10
Log2(FABP4 Expression)

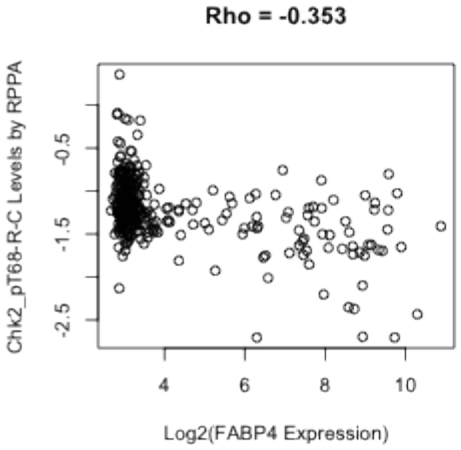

Rho = 0.265
S6_pS235_S236-R-V Levels by RPPA
0
-1
-2
-3
-4
3
4
5
6
7
8
9
Log2(ADH1B Expression)

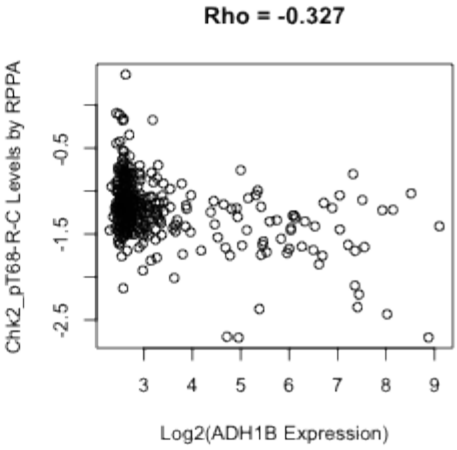

Rho = 0.113
S6_pS240_S244-R-V Levels by RPPA
1.0
0.0
-1.0
-2.0
4
6
8
10
Log2(FABP4 Expression)

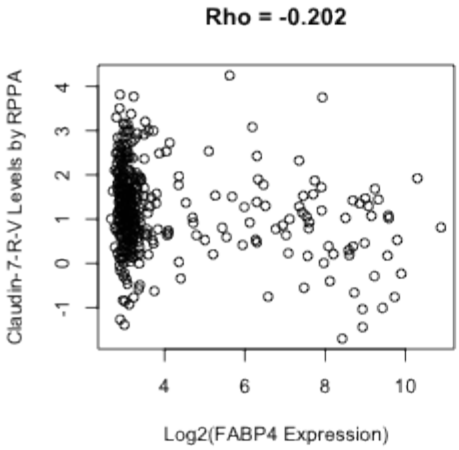

Rho = 0.221
S6_pS240_S244-R-V Levels by RPPA
1.0
0.0
-1.0
-2.0
3
4
5
6
7
8
9
Log2(ADH1B Expression)

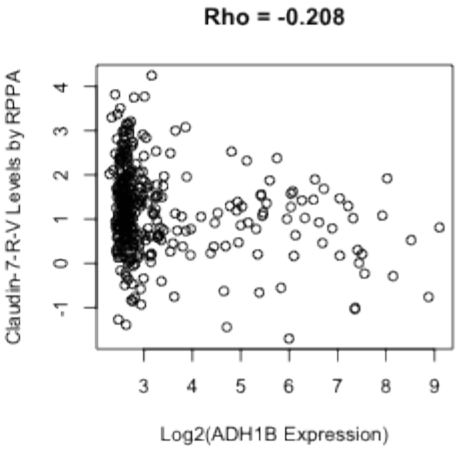

Rho = -0.295
Chk1_pS345-R-C Levels by RPPA
0.6
0.2
-0.2
-0.6
4
6
8
10
Log2(FABP4 Expression)

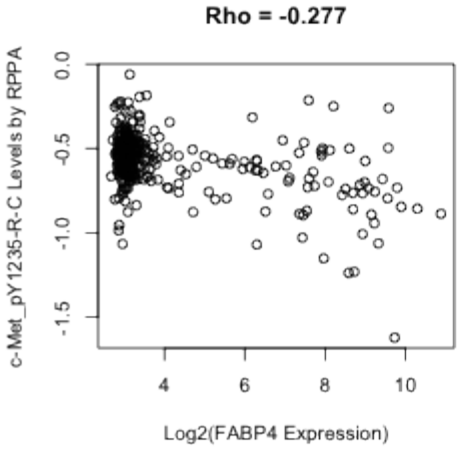

Rho = -0.221
Chk1_pS345-R-C Levels by RPPA
0.6
0.2
-0.2
-0.6
3
4
5
6
7
8
9
Log2(ADH1B Expression)

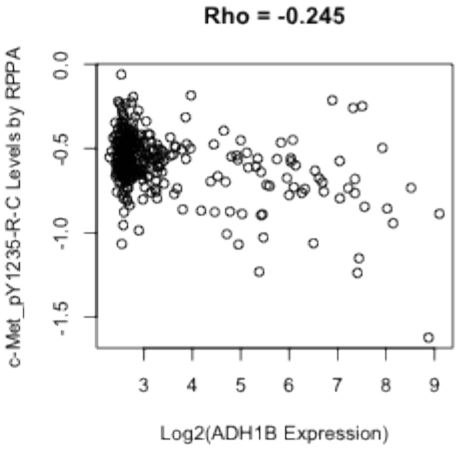

Rho = -0.353
Chk2_pT68-R-C Levels by RPPA
-0.5
-1.5
-2.5
4
6
8
10
Log2(FABP4 Expression)

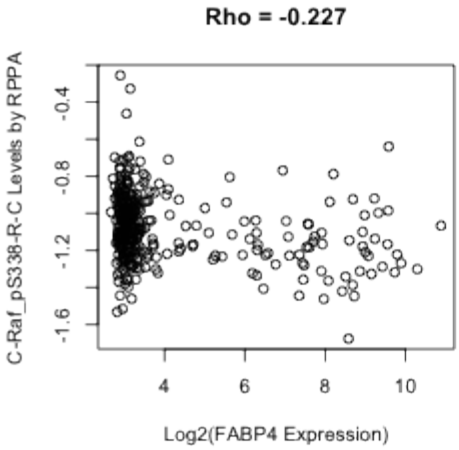

Rho = -0.327
Chk2_pT68-R-C Levels by RPPA
-0.5
-1.5
-2.5
3
4
5
6
7
8
9
Log2(ADH1B Expression)

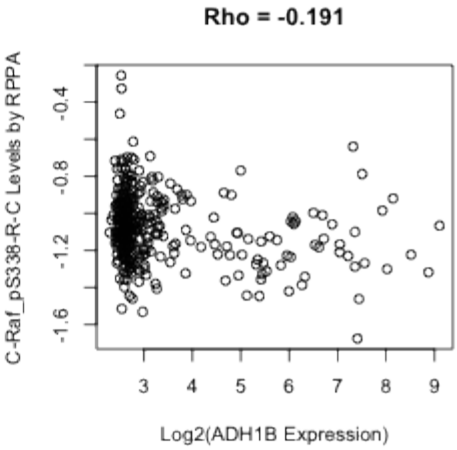

Rho = -0.202
Claudin-7-R-V Levels by RPPA
Log2(FABP4 Expression)

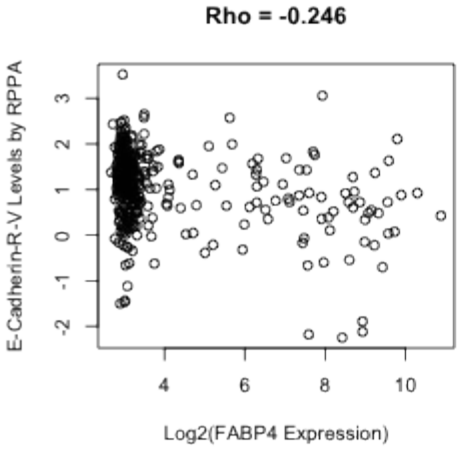

Rho = -0.208
Claudin-7-R-V Levels by RPPA
Log2(ADH1B Expression)

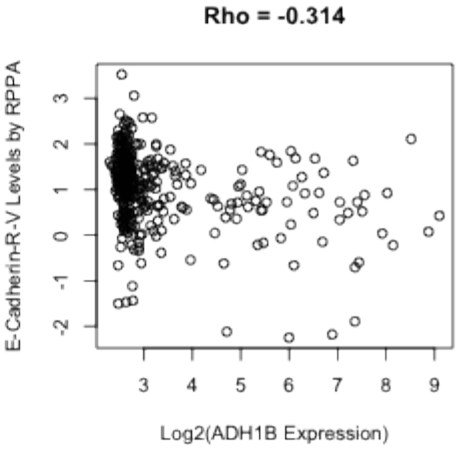

Rho = -0.277
c-Met_pY1235-R-C Levels by RPPA
Log2(FABP4 Expression)

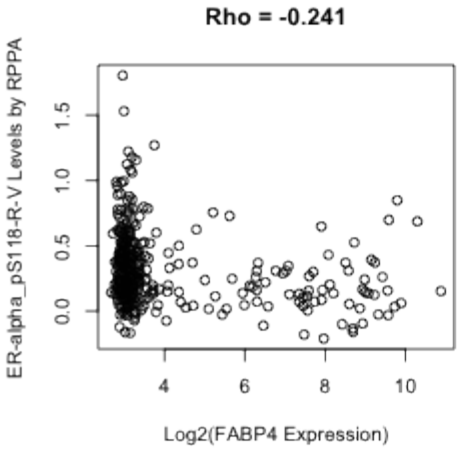

Rho = -0.245
c-Met_pY1235-R-C Levels by RPPA
Log2(ADH1B Expression)

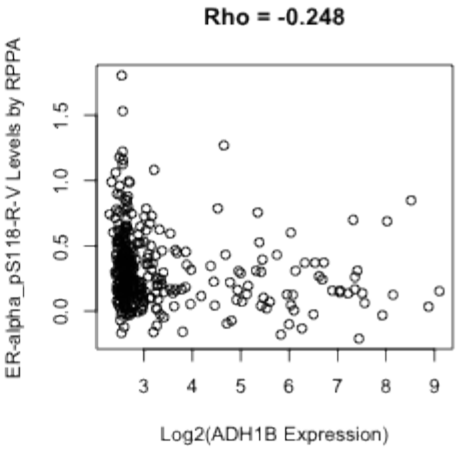

Rho = -0.227
C-Raf_pS338-R-C Levels by RPPA
-0.4
-0.8
-1.2
-1.6
4
6
8
10
Log2(FABP4 Expression)

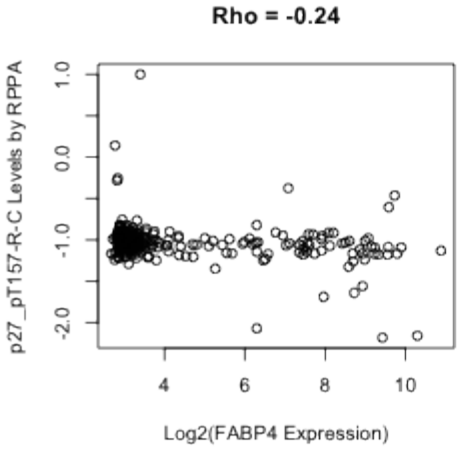

Rho = -0.191
C-Raf_pS338-R-C Levels by RPPA
-0.4
-0.8
-1.2
-1.6
3
4
5
6
7
8
9
Log2(ADH1B Expression)

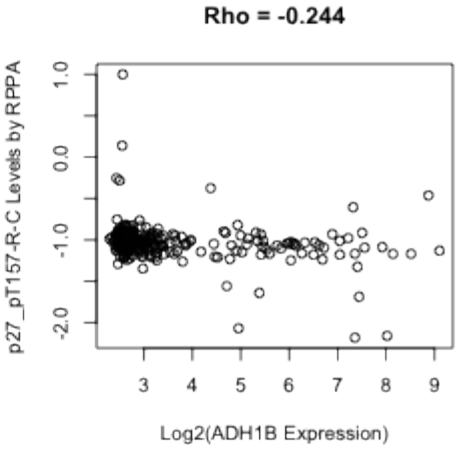

Rho = -0.246
E-Cadherin-R-V Levels by RPPA
3
2
1
0
-1
-2
4
6
8
10
Log2(FABP4 Expression)

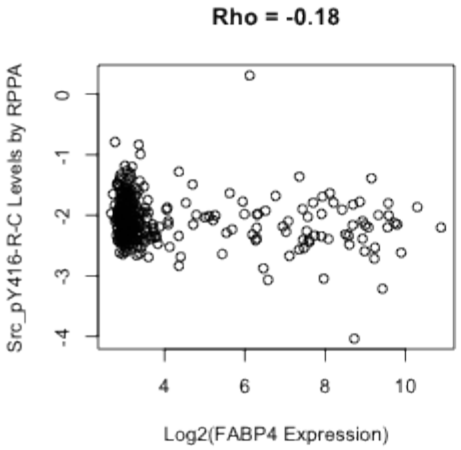

Rho = -0.314
E-Cadherin-R-V Levels by RPPA
3
2
1
0
-1
-2
3
4
5
6
7
8
9
Log2(ADH1B Expression)

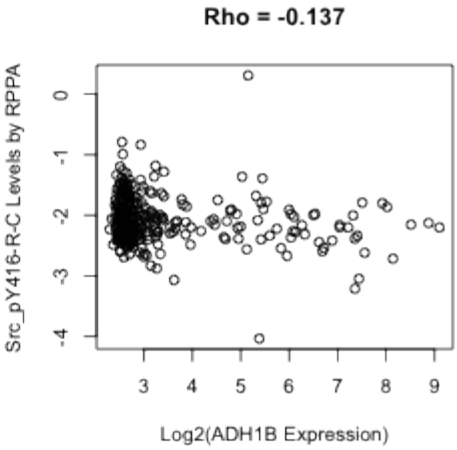

Rho = -0.241
ER-alpha_pS118-R-V Levels by RPPA
Log2(FABP4 Expression)

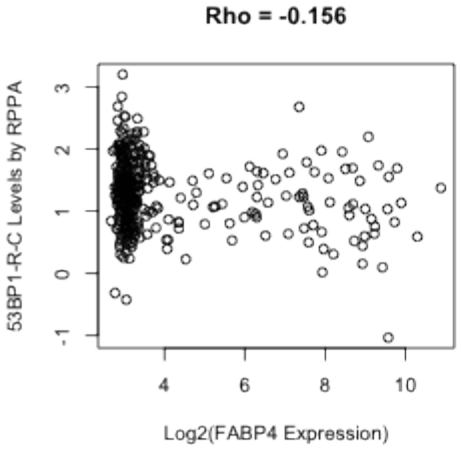

Rho = -0.248
ER-alpha_pS118-R-V Levels by RPPA
Log2(ADH1B Expression)

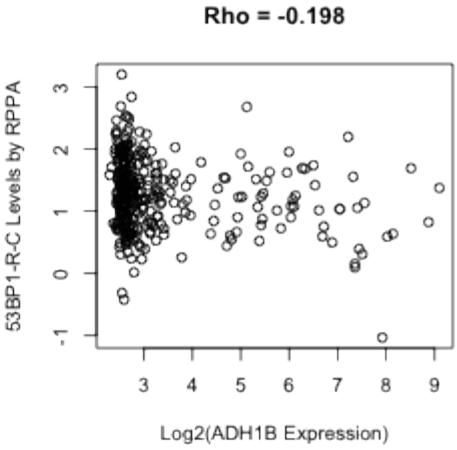

Rho = -0.24
p27_pT157-R-C Levels by RPPA
Log2(FABP4 Expression)

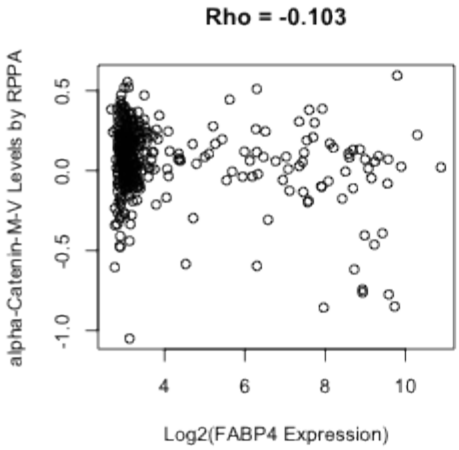

Rho = -0.244
p27_pT157-R-C Levels by RPPA
Log2(ADH1B Expression)

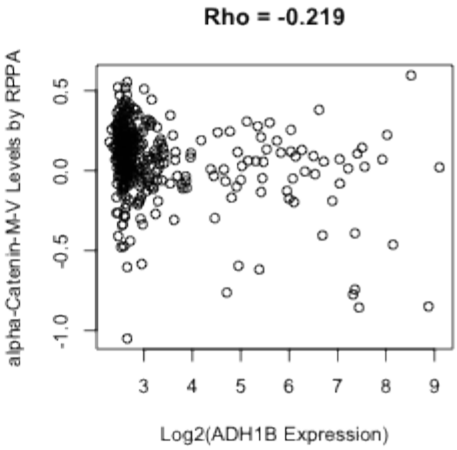

Rho = -0.18
Src_pY416-R-C Levels by RPPA
Log2(FABP4 Expression)

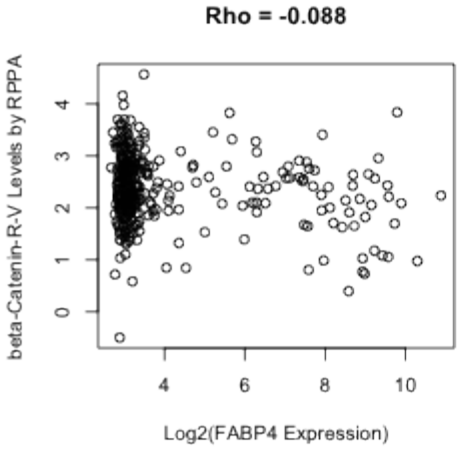

Rho = -0.137
Src_pY416-R-C Levels by RPPA
Log2(ADH1B Expression)

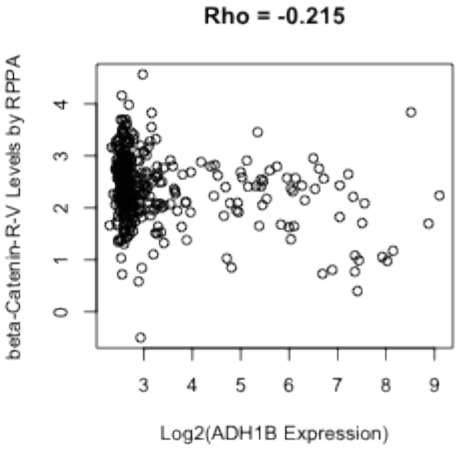

Rho = -0.156
53BP1-R-C Levels by RPPA
Log2(FABP4 Expression)

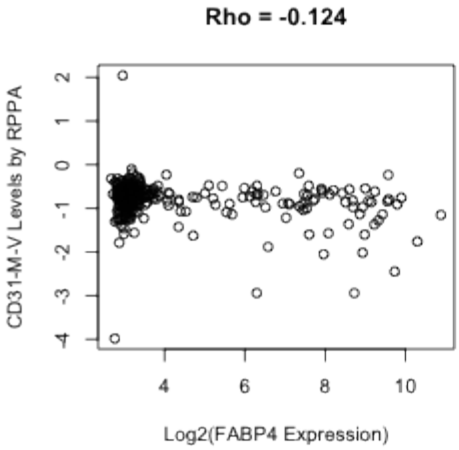

Rho = -0.198
53BP1-R-C Levels by RPPA
Log2(ADH1B Expression)

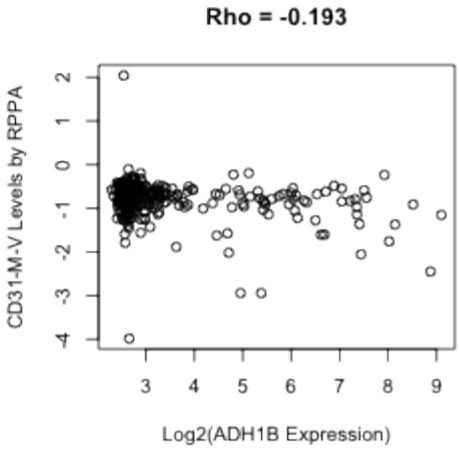

Rho = -0.103
alpha-Catenin-M-V Levels by RPPA
-1.0
-0.5
0.0
0.5
4
6
8
10
Log2(FABP4 Expression)

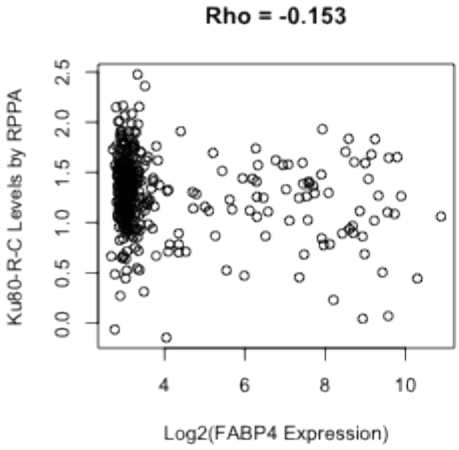

Rho = -0.219
alpha-Catenin-M-V Levels by RPPA
-1.0
-0.5
0.0
0.5
3
4
5
6
7
8
9
Log2(ADH1B Expression)

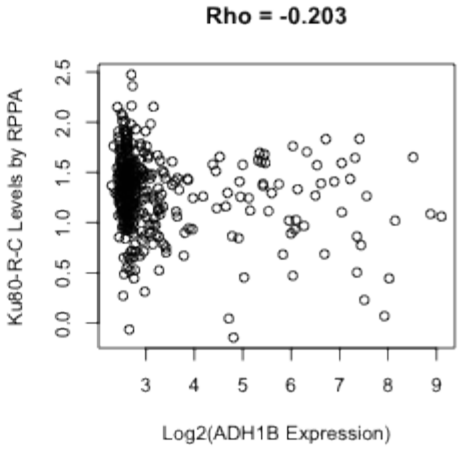

Rho = -0.088
beta-Catenin-R-V Levels by RPPA
0
1
2
3
4
4
6
8
10
Log2(FABP4 Expression)

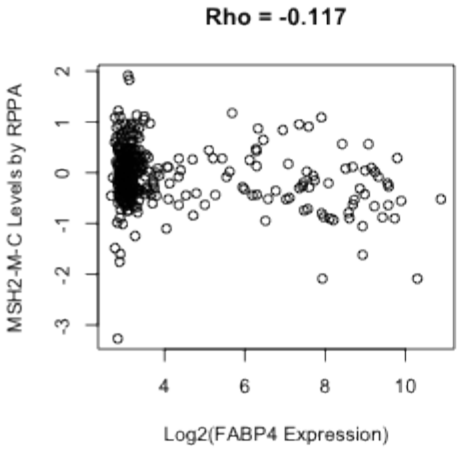

Rho = -0.215
beta-Catenin-R-V Levels by RPPA
0
1
2
3
4
3
4
5
6
7
8
9
Log2(ADH1B Expression)

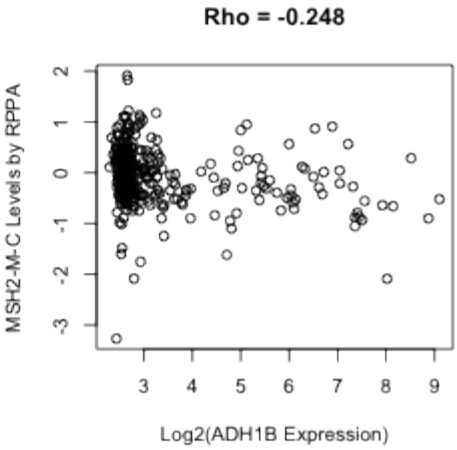

Rho = -0.124
CD31-M-V Levels by RPPA
Log2(FABP4 Expression)

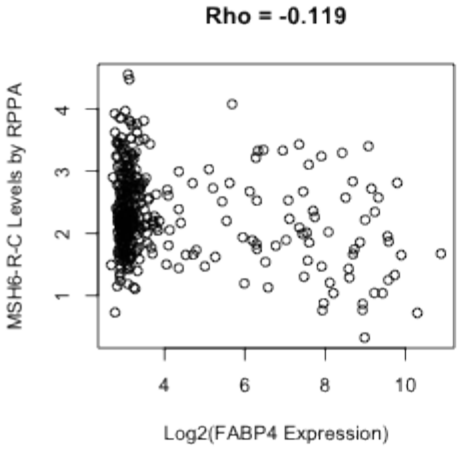

Rho = -0.193
CD31-M-V Levels by RPPA
Log2(ADH1B Expression)

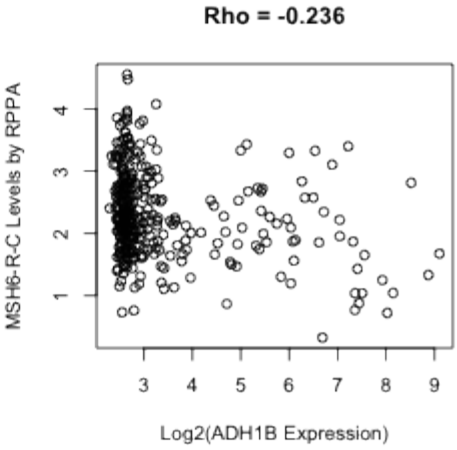

Rho = -0.153
Ku80-R-C Levels by RPPA
Log2(FABP4 Expression)

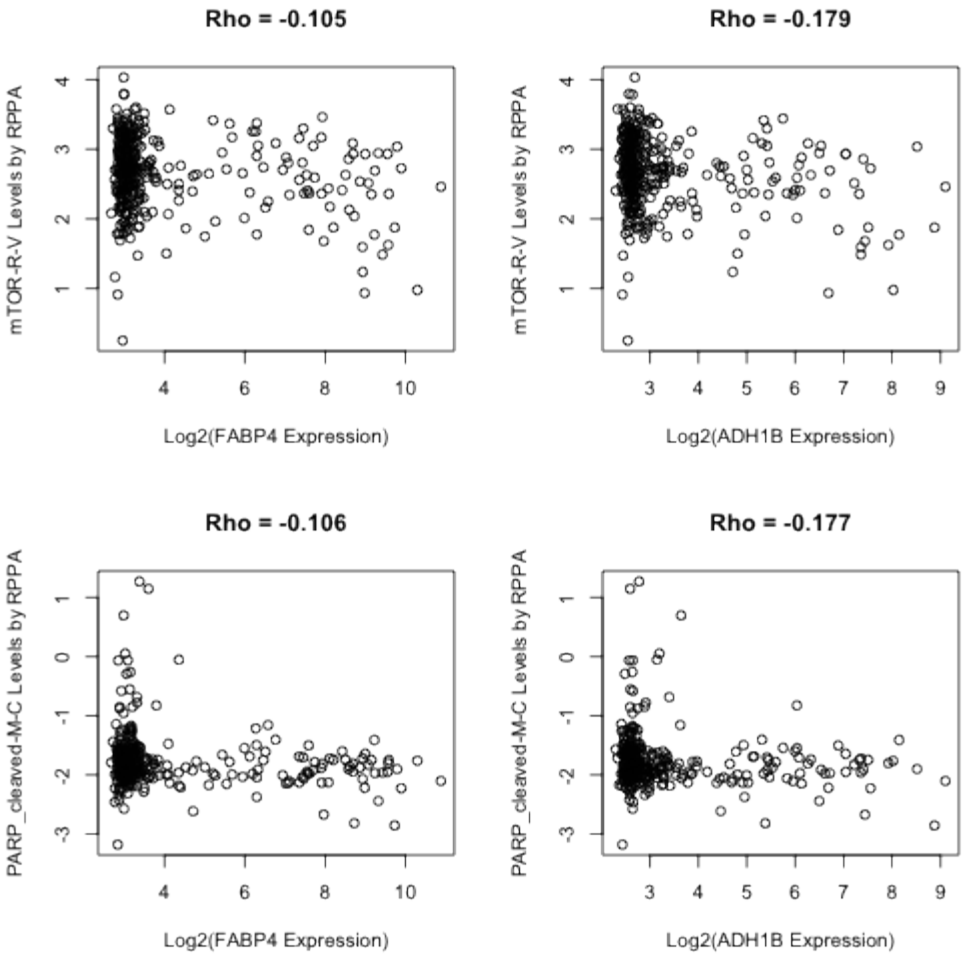

Rho = -0.203
Ku80-R-C Levels by RPPA
Log2(ADH1B Expression)


Rho = -0.117
MSH2-M-C Levels by RPPA
Log2(FABP4 Expression)


Rho = -0.248
MSH2-M-C Levels by RPPA
Log2(ADH1B Expression)


Rho = -0.119
MSH6-R-C Levels by RPPA
Log2(FABP4 Expression)


Rho = -0.236
MSH6-R-C Levels by RPPA
Log2(ADH1B Expression)



3.7 Output

Data from proteins of interest is exported for pathway analysis.

We prepare additional lists for export in addition to proteins meeting the P < 0.001 correlation criterion.

```
fabp4Proteins01 <- names(fabp4CorP)[fabp4CorP < 0.01]
adh1bProteins01 <- names(adh1bCorP)[adh1bCorP < 0.01]

rppaFABP4001 <- rppaUse[fabp4Proteins, ]
rppaADH1B001 <- rppaUse[adh1bProteins, ]

rppaFABP401 <- rppaUse[fabp4Proteins01, ]
rppaADH1B01 <- rppaUse[adh1bProteins01, ]

corFABP401 <- fabp4CorRho[fabp4Proteins01]
corADH1B01 <- adh1bCorRho[adh1bProteins01]

write.table(fabp4, file = "FABP4.csv", sep = ",")
write.table(adh1b, file = "ADH1B.csv", sep = ",")

write.table(rppaFABP4001, file = "rppaFABP4001.csv", sep = ",")
write.table(rppaADH1B001, file = "rppaADH1B001.csv", sep = ",")

write.table(rppaFABP401, file = "rppaFABP401.csv", sep = ",")
write.table(rppaADH1B01, file = "rppaADH1B01.csv", sep = ",")

write.table(corFABP401, file = "corFABP401.csv", sep = ",")
write.table(corADH1B01, file = "corADH1B01.csv", sep = ",")
```

4 Appendix

4.1 File Location

```
getwd()
```

```
## [1] "/Users/slt/SLT WORKSPACE/EXEMPT/OVARIAN/Ovarian residual disease study 2012/RD
manuscript/Web page for paper/Webpage"
```

4.2 SessionInfo

```
sessionInfo()
```

```
## R version 3.0.2 (2013-09-25)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] gplots_2.12.1 knitr_1.5
##
## loaded via a namespace (and not attached):
## [1] bitops_1.0-6       caTools_1.16       evaluate_0.5.1
## [4] formatR_0.10       gdata_2.13.2       gtools_3.2.1
## [7] KernSmooth_2.23-10 stringr_0.6.2      tools_3.0.2
```

5 References

[1] Hennessy BT, Lu Y, Gonzalez-Angulo AM, Carey MS, Myhre S, Ju Z, Davies MA, Liu W, Coombes K, Meric-Bernstam F, Bedrosian I, McGahren M, Agarwal R, Zhang F, Overgaard J, Alsner J, Neve RM, Kuo W-L, Gray JW, Borresen-Dale A-L, Mills GB. A Technical Assessment of the Utility of Reverse Phase Protein Arrays for the Study of the Functional Proteome in Non-microdissected Human Breast Cancers. *Clin Proteome*, **6**:129-51, 2010.