Correlation analysis between Expression (microarrays & RNA-seq) and methylation data in a set of cancer colon cell lines data. (0): Reading and preprocessing data

Alex Sánchez-Pla.
Statistics department. UB
& Statistics and Bioinformatics Unit (UEB). VHIR.

January 15, 2016

Contents

```
[1] "Package knitr already installed"
[1] "Package gdata already installed"
[1] "Package readxl already installed"
[1] "Package Biobase already installed"
[1] "Package annotate already installed"

Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
logical.return = TRUE, :  there is no package called 'pvca'


The downloaded source packages are in
```

```
/tmp/RtmpQC7UyY/downloaded_packages
[1] "Package VennDiagram already installed"
[1] "Package hgu133plus2.db already installed"
```

1 Introduction

The goal of this study is to check the correlation between expression and methylation values in a set of cell lines that have been analyzed to look for biomarkers for sensitivity to some drugs used in colon cancer treatment. Expression data have been obtained from microarrays and RNA-seq. Methylation has been measured on Illumina methylation arrays.

2 Data for the analysis

The data correspond to expression and methylation data from colon cancer cell lines characterized by their different sensitivities to drugs.

Data have been generated and preprocessed separately

- Expression microarrays have been normalized using the RMA algorithm. Probesets corresponding to duplicate identifiers have been removed (the probe with the highest variance is retained in each case). Data have been batch-centered to remove batch effect due to the place from where samples were processed.
- Methylation values have been normalized using standard approaches for this type of data.
- RNA-seq data have been preprocessed using standard approaches and turned into normalized counts using the RPKM algorithm. Only counts that could be assigned to genes have been retained.

A note on the forecoming analyses The analyses presented below have been performed first with a smaller set of data than presented here. Once the pipeline has been ready an effort to compile as much valid data as possible has been made so that the number of samples and genes available has been increased making this second version more comprehensive.

2.1 Microarray data

Microarray data have been provided by the researcher who took care of preprocessing them. Preprocessing consisted of RMA normalization followed by batch adjustment (for the ”core facility” batch effect) using the COMBAT method.

```
> readFromExcel <- function (usexlconnect=FALSE, usegdata=FALSE,
+                            dataFileName, dataFolder=".",
+                            rowNames=TRUE){
+   fileName<- file.path(dataFolder, dataFileName)
+   if (usegdata){
+   require(gdata)
+   if(rowNames){
+       dataSheet <- read.xls(fileName, sheet=1, header=TRUE, row.names=1)
+   }else{
+       dataSheet <- read.xls(fileName, sheet=1, header=TRUE)
+   }
+   }else{
+     if(usexlconnect){
+       require(XLConnect)
+       wb = loadWorkbook(fileName)
+       if(rowNames){
+           dataSheet <- readWorksheet(wb, sheet = 1, header = TRUE, row.names=1)
+       }else{
+           dataSheet <- read.xls(fileName, sheet=1, header=TRUE)
+       }
+     }else{
+       require(readxl)
+       dataSheet <-read_excel(fileName)      ### COMPTE: No els llegeix be!. No se perqu
+       if(rowNames){
+           if(length(dataSheet[,1])==length(unique(dataSheet[,1]))){
+               rownames(dataSheet)<-dataSheet[,1]
+               dataSheet<-dataSheet[,-1]
+           }else{
+               stop("Duplicate identifiers in row names are not allowed")
+           }
+       }
+     }
+   }
+   return(dataSheet)
+ }
```

```
> readExcel <- TRUE
>
> if (readExcel){
+   dataMarr0 <- readFromExcel (usexlconnect=FALSE, usegdata = TRUE,
+                               dataFileName ="microarrayValuesNEW.xls", dataFolder=combined
+   save(dataMarr0, file=file.path(combinedDir, "microarrayValuesNEW.Rda"))
+ }else{
+   load(file=file.path(combinedDir, "microarrayValuesNEW.Rda"))
+ }
>
> expres0 <- as.matrix(dataMarr0[, 10:51])
> annotsMarr0 <-dataMarr0[,1:9]
> colnames(annotsMarr0)

[1] "UGCluster"    "Name"         "Symbol"
[4] "Aliases"      "GeneID"       "UGRepAcc"
[7] "LLRepProtAcc" "Chromosome"   "Cytoband"

> TargetsFilename <-"microarrayTargetsNEW.xls"
> targetsMarr0 <-read.xls(file.path(combinedDir, TargetsFilename), sheet=1, header=TRUE,
+                            row.names=1)
> all(colnames(expres0)==rownames(targetsMarr0))

[1] TRUE

> require(Biobase)
> metadata <- data.frame(labelDescription=colnames(targetsMarr0),
+                         row.names=colnames(targetsMarr0))
> pData0 <- new("AnnotatedDataFrame", data=targetsMarr0, varMetadata=metadata)
> annot0 <- "hgu133plus2"
> eset0 <- ExpressionSet(assayData= expres0, phenoData=pData0, annotation = annot0)
> save(eset0, file= file.path(resultsDir, "esetMicroarraysNew.Rda"))
```

2.1.1 Filtering and aggregating microarray data

Microarray values are filtered (i) to remove genes without an Entrez identifier and (ii) to keep a single value per gene. Instead of averaging probes an "ad-hoc" function has been used to represent each gen by the probeset with highest variability.

```
> maxVarByID <- function (x, genes){
+   lincs<-cbind(P=rownames(x), G=genes)
+   geneNames <- unique(genes)
+   numGenes<-length(geneNames)
+   maxVars<-matrix(0, nrow=numGenes, ncol=ncol(x))
+   rownames(maxVars) <- geneNames
+   colnames(maxVars) <- colnames(x)
+   i<-0
+   for(g in geneNames){
+     i<-i+1
+     subsX <- x[lincs[,2]==g,]
+     if (!is.matrix(subsX)){
+          maxVars[i,]<- subsX
+  }else{
+    sds <- apply(as.matrix(subsX), 1, sd)
+    maxIdx <- which(sds==max(sds))
+    maxVars[i,] <-subsX[maxIdx,]
+  }
+ }
+ return(maxVars)
+ }
> # test
> # x<- matrix(c(1,2,2,2,1, 1,3,5,3,1, 1,1,2,1,1, 1,5,8,5,1, 1,1,2,2,1), nrow=5, byrow=TRUE
> # rownames(x)<-paste("p",1:5, sep="")
> # genes <- paste("g", c(1,2,2,3,3), sep="")
> # aggregate(x, by=list(genes), FUN=mean)
> # maxVarByID (x, genes)
```

First genes without Entrez Identifiers are removed

```
> expres1 <- expres0[!is.na(annotsMarr0$GeneID),]
> annotsMarr1 <- annotsMarr0[!is.na(annotsMarr0$GeneID),]
> # nrow(expres0); nrow(expres1)
```

Next the array is collapsed by gene identifiers keeping only one probeset by gene, the one with the highest variance.

```
> expres2 <- maxVarByID (expres1, annotsMarr1[,"GeneID"])
> colnames(expres2) <-colnames(expres1)
> genes2Symbols<- annotsMarr1[,c("GeneID", "Symbol")]
> genes2Symbols <- genes2Symbols[!duplicated(genes2Symbols[,1]),]
> # head(expres2[1:5,1:3]);
> # head(genes2Symbols)
> sum(rownames(expres2)!=genes2Symbols[,1])

[1] 0

> rownames(expres2) <-genes2Symbols[,2]
> # dim(expres2)
> save(expres2, file= file.path(resultsDir, "expresMicroarraysNewFiltered.Rda"))
> write.csv2(expres2, file=file.path(resultsDir, "expresMicroarraysNewFiltered.csv"), quote=
```

Filtered microarray data contains the expression values of 19991 unique genes and 42cell lines.

2.2 Methylation data

Preprocessesd methylation data have been provided by the core facility an excel file with two columns per cell line, one with “B-values” (that is the estimates of the methylation percentage") per each CpG (methylation sites) and one with quality scores for these values.

```
> require(gdata)
>
> readExcel <- TRUE
> if (readExcel){
+   dataMeth0 <- read.xls(file.path(combinedDir,"methylationValuesNEW.xls"), sheet=1, header
+   nrow(dataMeth0)
+   save(dataMeth0, file=file.path(combinedDir, "methylationValuesNEW.Rda"))
+ }else{
+   load(file=file.path(combinedDir, "methylationValuesNEW.Rda"))
+ }
>
> # colnames(dataMeth0)
> cols2remove <-grep("Detection", colnames(dataMeth0))
> dataMeth0<- dataMeth0[,-cols2remove]
> annotsMeth0<-dataMeth0[,1:2]
> dataMeth0<- dataMeth0[,-c(1,2)]
> # colnames(dataMeth0)
> cellLinesNames <- sapply(colnames(dataMeth0),
+                          function (s) substr(s, 1, nchar(s)-nchar(".AVG.Beta")))
> colnames(dataMeth0) <- cellLinesNames
> # head(dataMeth0)
> rownames(dataMeth0)<-annotsMeth0[,"TargetID"]
> colnames(dataMeth0)<-toupper(colnames(dataMeth0))
>
> ### Some final refinements
> dataMeth0 <- dataMeth0[,]
> # Remove column namd "JUNKER"
> noJURKMET<-which(colnames(dataMeth0)!="JURKMET")
> dataMeth0 <- dataMeth0[,noJURKMET]
> # Average Replicates for SW48
> SW48 <- (dataMeth0[,"SW48.REPLICATE.1"]+dataMeth0[,"SW48.REPLICATE.2"])/2
> dataMeth0[,"SW48.REPLICATE.1"]<- SW48
> colnames(dataMeth0)[which(colnames(dataMeth0)=="SW48.REPLICATE.1")]<-"SW48"
> noSW48.REPLICATE.2<-which(colnames(dataMeth0)!="SW48.REPLICATE.2")
> dataMeth0 <- dataMeth0[,noSW48.REPLICATE.2]
> dim(dataMeth0)

[1] 27578    46

> save(dataMeth0, annotsMeth0, file=file.path(resultsDir, "methylationValuesNew.Rda"))
```

2.2.1 Filtering and aggregating methylation data

The data have to be checked to detect any site without associated gene Symbol.

```
> (noSymbol <- sum(annotsMeth0[,"SYMBOL"]==""))

[1] 27

> sitesWithoutSymbol<- which(annotsMeth0[,"SYMBOL"]=="")
> dataMeth0 <- dataMeth0[-sitesWithoutSymbol,]
> annotsMeth0 <- annotsMeth0[-sitesWithoutSymbol,]
```

There are 27 sites without an associated gene symbol. They have been removed from the dataset.

Besides this these data may contain some missing values. However these have also been either removed or imputed.

```
> numDataMeth<- as.matrix(dataMeth0)
> means <- apply(numDataMeth, 1, mean)
> withMissings <- numDataMeth[which(is.na(means)),]
> length(withMissings)

[1] 0
```

There are no genes with, at least one missing value.

Methylation data are computed on methylation sites from which there may be several per gene. In order to match methylation and expression values it is needed to "unitize" the values assigning a unique methyation value to each gene. This can be done similarly to what was done with microarray values (keeping the site with highest variability) o just averaging the different values.

```
> numDataMeth1 <- aggregate(numDataMeth, by=list(annotsMeth0[,"SYMBOL"]), FUN=mean)
> rownames(numDataMeth1) <-numDataMeth1[,1]
> colnames(numDataMeth1)

 [1] "Group.1" "ALA"     "CACO2"   "CO115"   "COLO201"
 [6] "COLO205" "COLO320" "DLD1"    "GP5D"    "HCA7"
[11] "HCC2998" "HCT116"  "HCT15"   "HDC108"  "HDC9"
[16] "HT29"    "IS1"     "IS2"     "IS3"     "KM12"
[21] "LIM1215" "LIM2405" "LOVO"    "LS1034"  "LS174T"
[26] "LS513"   "RKO"     "RW2982"  "RW7213"  "SKCO1"
[31] "SW1116"  "SW403"   "SW48"    "SW620"   "SW837"
[36] "SW948"   "T84"     "TC71"    "V9P"     "VACO5"
[41] "FET"     "HDC111"  "HDC114"  "HDC75"   "HDC87"
[46] "HDC15"   "NOMET"
```

```
> numDataMeth1<-numDataMeth1[,-1]
> numDataMethByMean<-numDataMeth1
```

```
> numDataMeth2 <- maxVarByID (numDataMeth, as.character(annotsMeth0[,"SYMBOL"]))
> cpgs2Symbols<- annotsMeth0
> cpgs2Symbols <- cpgs2Symbols[!duplicated(cpgs2Symbols[,"SYMBOL"]),]
> # head(expres2[1:5,1:3]);
> # head(genes2Symbols)
> sum(rownames(numDataMeth2)!=cpgs2Symbols[,2])

[1] 0

> # dim(expres2)
> numDataMethByVar<-numDataMeth2
```

```
> write.csv2(numDataMeth1, file=file.path(resultsDir, "methylationDataNewAgregatedByMean.csv
> write.csv2(numDataMeth2, file=file.path(resultsDir, "methylationDataNewAgregatedByVar.csv"
> save(numDataMethByMean, numDataMethByVar, file= file.path(resultsDir, "methylationDataNewA
```

2.3 RNA-seq data

Normalized RNAseq data have been provided in a text file (”RNA seq - 60 colon cancer cell lines - MASTER FILE.xlsx”). This file contains expression values for genes and other types of transcripts such as microRNAs, SNORDs and others. All of these have no Gene (symbol) associated and they have been removed for this study

```
> require(gdata)
> readExcel <- TRUE
> if (readExcel){
+   dataRNAseq <- read.xls(file.path(combinedDir,"RNAseqValuesNEW.xls"), sheet=1, header=TRU
+   nrow(dataRNAseq)
+   save(dataRNAseq, file=file.path(combinedDir, "RNAseqValuesNEW.Rda"))
+ }else{
+   load(file=file.path(combinedDir, "RNAseqValuesNEW.Rda"))
+ }
>
> nrow(dataRNAseq)

[1] 22470

> colnames(dataRNAseq)
```

```
 [1] "C125.PM"       "C135"          "C70"
 [4] "CACO2"         "CCK81"         "COLO201"
 [7] "COLO205"       "COLO320"       "CX1"
[10] "DIFI"          "DLD1"          "GEO"
[13] "GP2D"          "GP5D"          "HCA7"
[16] "HCC2998"       "HCT116"        "HCT15"
[19] "HCT8"          "HDC54"         "HDC57"
[22] "HDC90"         "HRA19"         "HT115"
[25] "HT29"          "HT55"          "IS1"
[28] "IS2"           "IS3"           "KM12"
[31] "LIM1215"       "LIM1863"       "LIM1899"
[34] "LIM2099"       "LIM2405"       "LIM2537"
[37] "LIM2550"       "LIM2551"       "LS513"
[40] "NCIH747"       "RKO"           "RW2982"
[43] "RW7213"        "SKCO1"         "SNU175"
[46] "SNUC2B"        "SW1112"        "SW1116"
[49] "SW1222"        "SW403"         "SW480_APC"
[52] "SW480_CONTROL" "SW480"         "SW620"
[55] "SW837"         "SW948"         "T84"
[58] "V9P"           "VACO10"        "VACO4S"
[61] "VACO5"         "X"             "X.1"
[64] "X.2"           "X.3"           "X.4"
[67] "X.5"           "X.6"           "X.7"
[70] "X.8"           "X.9"           "X.10"
[73] "X.11"          "X.12"          "X.13"
[76] "X.14"          "X.15"          "X.16"
[79] "X.17"          "X.18"          "X.19"
[82] "X.20"          "X.21"
```

```
> RNAseqSymbols <- rownames(dataRNAseq)
>
> # Remove "ghost columns"
> if (ncol(dataRNAseq) > 62)
+   dataRNAseq <- dataRNAseq[,-c(62:84)]
> # Remove columns with SW480 controls
> noSW480 <-which(colnames(dataRNAseq)!="SW480_APC")
> dataRNAseq <- dataRNAseq[,noSW480]
> noSW480 <-which(colnames(dataRNAseq)!="SW480_CONTROL")
> dataRNAseq <- dataRNAseq[,noSW480]
> dim(dataRNAseq)
```

```
[1] 22470    59
```

```
> namesRNAseq <- colnames(dataRNAseq)
```

2.3.1 Detecting and removing genes with too many null counts from RNAseq data

RNA-seq data contains the expressions of 22470 genes computed on 59 cell lines. A known issue of these data type is the fact that it can contain zeroes which can affect computations.

```
> dataRNAseq[1:10,c(1:3)]

         C125.PM      C135      C70
CCDC124 80.80247 134.76315 49.71716
STK35   14.29656   9.36959 21.21671
DPYSL4   0.00000   0.06562  0.02035
GJC2     0.07915   0.32472  0.24403
FMNL1    1.09432   0.28528  0.22353
BICC1    1.26890   0.05720  2.72342
LIG3    12.20974   9.92753 16.22533
CA6      0.00000   0.00000  0.00000
BRPF1    9.75260   8.64180  8.68457
SRL      0.41741   0.01411  0.02626
```

In order to avoid including genes with too many zeroes –which means suggest genes that are not expressed in most samples– those with ***more than 66% of zeros*** will be removed from the data.

```
> zeros <-function(x){which (x==0)}
> discard <- function(x, where, howMany){
+   return(length(zeros(x[where])) > howMany)
+ }
>
> discardA <- function(x, percentage){
+     maxZeros <- ceiling (length(x)*percentage)
+     return (discard(x,1:length(x),maxZeros))
+ }
> # test
> # exData <- matrix (0, nrow=10, ncol=10)
> # for (i in 1:10){
> #     for(j in 1:i)
> #         exData[i,j]<-1
> # }
> # d1 <-apply(exData, 1, discardA, 0.66); length(d1); sum(d1);show(exData1 <- exData[!d1,]
> # d1 <-apply(exData, 1, discardA, 0.5); length(d1); sum(d1);show(exData1 <- exData[!d1,])
> # d1 <-apply(exData, 1, discardA, 0.33); length(d1); sum(d1);show(exData1 <- exData[!d1,]
> # d1 <-apply(exData, 1, discardA, 0.2); length(d1); sum(d1);show(exData1 <- exData[!d1,])
> discardedA <- apply(dataRNAseq, 1, discardA, 0.65)
> length(discardedA); sum(discardedA)

[1] 22470
[1] 5062

> dataRNAseqA <- dataRNAseq [!discardedA,]
> dim(dataRNAseqA)

[1] 17408    59

> RNAseqSymbolsA <-rownames(dataRNAseqA)
```

The number of genes left after removing genes with at least 66% zero values is 17408 that is there have been removed 5862 genes. Although these values might be relabelled as missing values, instead they will be kept as zeroes

```
> save(dataRNAseq, dataRNAseqA, file= file.path(resultsDir, "RNAseqDataNew.Rda"))
> write.csv2(dataRNAseq, file=file.path(resultsDir, "RNAseqDataNew.csv"))
> write.csv2(dataRNAseqA, file=file.path(resultsDir, "RNAseqDataNewLESSZeros.csv"), quote=FA
```

2.4 Genes in common between the three datasets

2.4.1 Genes in common between the three datasets

The dimensions of the three datasets availabe are not the same.

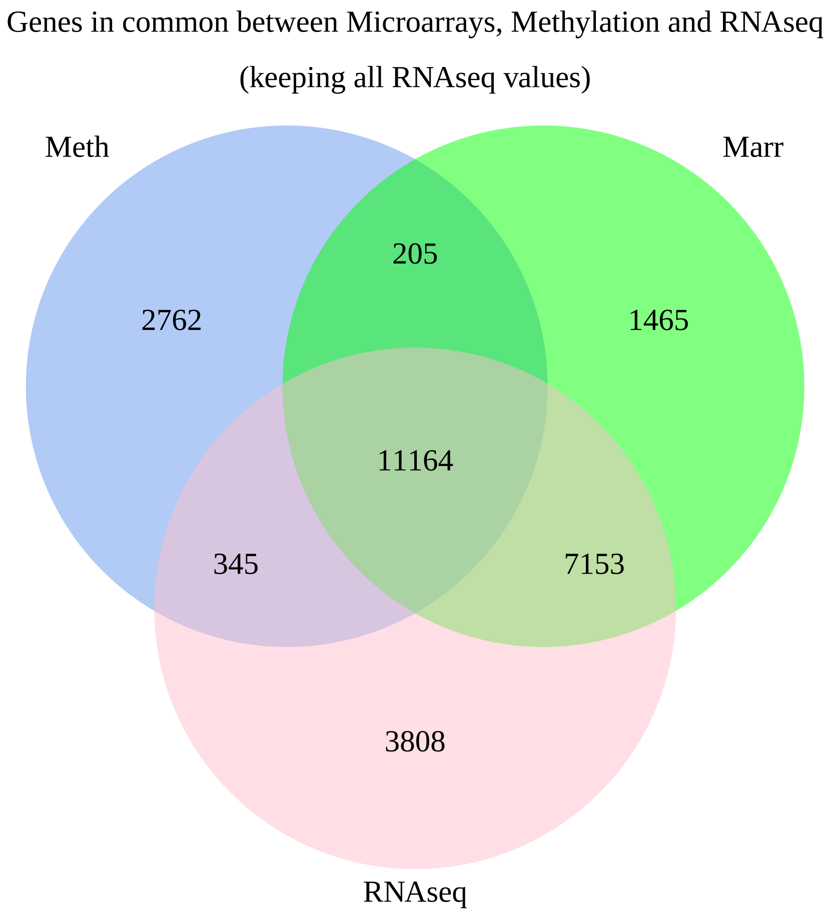
```
> dim(expres2)

[1] 19991    42

> dim(numDataMeth2)

[1] 14476    46

> dim(dataRNAseqA)

[1] 17408    59
```

This means that we may expect to have different gene names and different sample names between them.

```
> marrSymbols <- rownames(expres2)
> methSymbols <- rownames(numDataMeth2)
> RNAseqSymbols <- rownames(dataRNAseq)
> RNAseqSymbolsA <- rownames(dataRNAseqA)
```

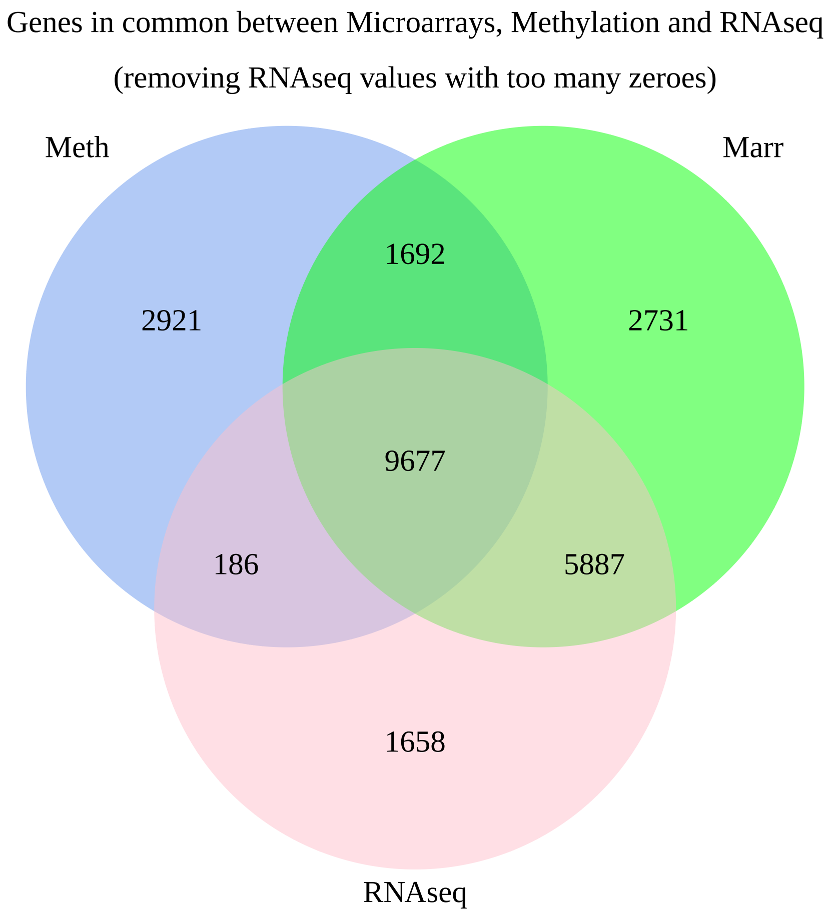
If we consider all genes available in the RNAse files we get:

```
> flog.threshold(ERROR)

NULL

> par(mfrow=c(2,1))
> require(VennDiagram)
> vd<- venn.diagram(list(Meth=methSymbols, Marr=marrSymbols, RNAseq=RNAseqSymbols),
+                    filename=NULL,
+                    col = "transparent", fill = c("cornflowerblue", "green", "pink"),
+                    main ="Genes in common between Microarrays, Methylation and RNAseq",
+                    sub="(keeping all RNAseq values)")
> grid.draw(vd)
```



Ignoring those with too many zeroes the result is:

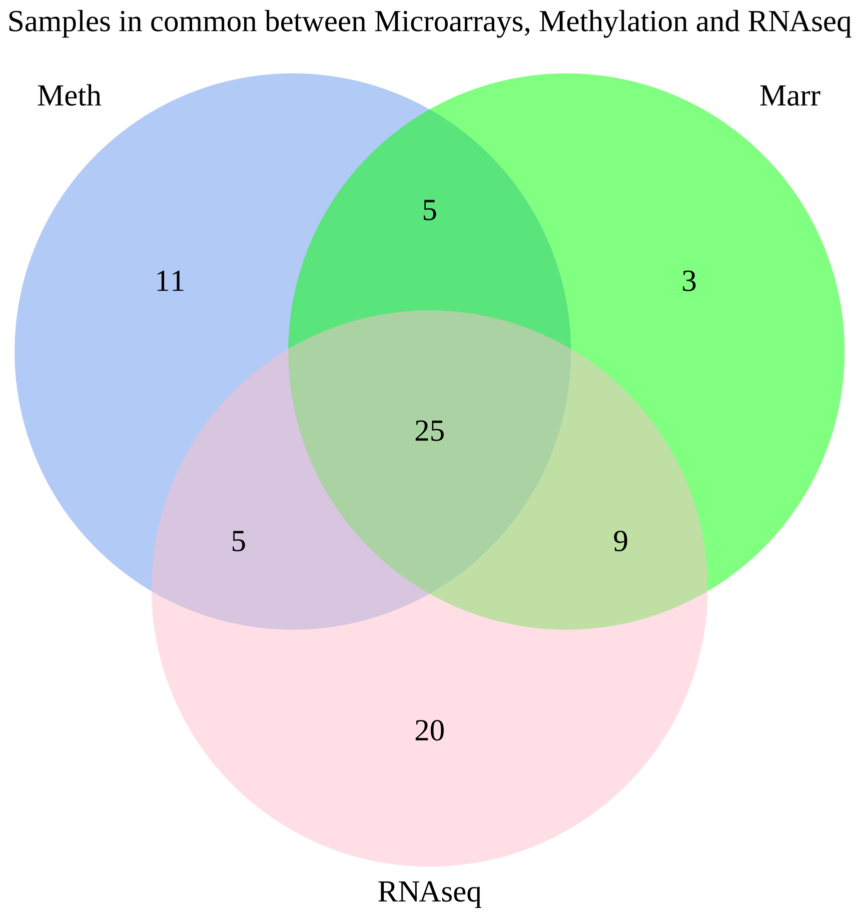
```
> #dev.new()
> vd1<- venn.diagram(list(Meth=methSymbols, Marr=marrSymbols, RNAseq=RNAseqSymbolsA),
+                    filename=NULL,
+                    col = "transparent", fill = c("cornflowerblue", "green", "pink"),
+                    main ="Genes in common between Microarrays, Methylation and RNAseq",
+                    sub="(removing RNAseq values with too many zeroes)")
> grid.draw(vd1)
```



2.4.2 Genes in common between the three datasets

Methylation and gene expression has been measured on many common cell lines, although some are not the same

```
> marrNames <- toupper(colnames(expres2))
> methNames <- toupper(colnames(numDataMeth2))
> RNAseqNames <- toupper(colnames(dataRNAseq))
> #dev.new()
> vd2<- venn.diagram(list(Meth=methNames, Marr=marrNames, RNAseq=RNAseqNames),
+                    filename=NULL,
+                    col = "transparent", fill = c("cornflowerblue", "green", "pink"),
+                    main ="Samples in common between Microarrays, Methylation and RNAseq")
> grid.draw(vd2)
```



Before we can investigate the existing correlation between each dataset some work has to be done to match the datasets on a “per-gene” and “per-sample” basis.

```
> sampleNames<-cbind(c(sort(marrNames), sort(methNames), sort(RNAseqNames)),
+                     c(rep("marr", length(marrNames)), rep("meth", length(methNames)),rep("R
> table(sampleNames[,1], sampleNames[,2])


          marr meth RNAseq
```

```
ALA        0    1      0
C125.PM    0    0      1
C135       0    0      1
C70        0    0      1
CACO2      1    1      1
CCK81      0    0      1
CO115      1    1      0
COLO201    1    1      1
COLO205    1    1      1
COLO320    1    1      1
CX1        0    0      1
DIFI       1    0      1
DLD1       1    1      1
FET        0    1      0
GEO        0    0      1
GP2D       0    0      1
GP5D       0    1      1
HCA7       0    1      1
HCC2998    1    1      1
HCT116     1    1      1
HCT15      1    1      1
HCT8       1    0      1
HDC108     0    1      0
HDC111     0    1      0
HDC114     0    1      0
HDC15      0    1      0
HDC54      0    0      1
HDC57      0    0      1
HDC75      0    1      0
HDC87      0    1      0
HDC9       0    1      0
HDC90      0    0      1
HRA19      0    0      1
HT115      0    0      1
HT29       1    1      1
HT55       0    0      1
IS1        1    1      1
IS2        1    1      1
IS3        0    1      1
KM12       1    1      1
LIM1215    1    1      1
LIM1863    1    0      1
LIM1899    1    0      1
LIM2099    0    0      1
LIM2405    1    1      1
```

```
  LIM2537    1    0      1
  LIM2550    1    0      1
  LIM2551    1    0      1
  LOVO       1    1      0
  LS1034     0    1      0
  LS174T     1    1      0
  LS411      1    0      0
  LS513      0    1      1
  NCIH747    0    0      1
  NOMET      0    1      0
  RKO        1    1      1
  RW2982     1    1      1
  RW7213     1    1      1
  SKCO1      1    1      1
  SNU175     0    0      1
  SNUC2B     0    0      1
  SW1112     0    0      1
  SW1116     1    1      1
  SW1222     1    0      1
  SW1417     1    0      0
  SW403      1    1      1
  SW48       1    1      0
  SW480      1    0      1
  SW620      1    1      1
  SW837      1    1      1
  SW948      1    1      1
  T84        1    1      1
  TC71       1    1      0
  V9P        0    1      1
  VACO10     0    0      1
  VACO432    1    0      0
  VACO4S     0    0      1
  VACO5      1    1      1
```