STA 525: FINAL PROJECT

MAY 10, 2016

STATE UNIVERSITY OF NEWYORK AT BUFFALO

1 INTRODUCTION

Multiple changes in cancer cells like chromosomal instability, activation of oncogenes, silencing of tumor suppressor genes, inactivation of DNA repair systems are caused not only by genetic but also by epigenetic abnormalities. During carcinogenesis, the genome simultaneously undergoes genome-wide hypomethylation & regional hypermethylation of CpG islands133, which may be of selective advantage for the incipient tumor cell. Both hypo- and hypermethylation of the genome can have epigenetic and genetic consequences for the cell.

Regarding the cancer dataset we focused on, Pancreatic ductal adenocarcinoma (PDAC) is the most common malignancy of the pancreas. PDAC is an aggressive and difficult malignancy to treat. Pancreatic ductal adenocarcinoma (PDAC) is the fourth leading cause of cancer-related deaths in the United States, with dismal overall 5-year survival rates of 6%. PDAC most often presents at an advanced stage and with metastatic disease.

Bisulfite is the Gold standard technology for enabling single base resolution. Uracil is replaced by Thymine by DNA Polymerization during amplification. Both 5mC & 5hmC are substantially resistant to mutagenesis conditioned by bisulfite ion. "C" interpreted as either 5mC or 5hmC in pre-treated DNA. Destroys as much as 95% of sample via acid hydrolysis.

Whole-Genome Bisulfite Sequencing is an efficient nanogram-scale library preparation, sequencing, and analysis of DNA methylation. In order to generate an average read depth of 14-15x per strand for two different samples which equates to approximately 90 Gb pairs of data per sample. Nearly 400 total lanes of Illumina GA2 sequencing were required. With today's sequencing technology this equates to approximately 3 lanes of HiSeq 2000 run or a HiSeq2500 run in high output mode using 2x100 reads. So, while sequencing throughput has increased dramatically, obviously it is still not a practical solution for most research needs. The EpiGnome workflow results in di-tagged DNA that is amplified by PCR, resulting in directional libraries with the appropriate adapters for Illumina sequencing.

Further research into epigenetic changes could lead into the research about the cause or trigger for the differentially marking of the genome which are still unknown. There are still questions on the variety of stimuli that can bring about epigenetic changes, ranging from developmental progression & aging to viral infection & diet. More in-depth analysis will help us have a better understanding & interference into the ways in which genome learns from its experience. Whole-genome bisulfite sequencing (WGBS) allows genome-wide DNA methylation profiling, but the current issues regarding associated high sequencing costs continue to limit its widespread application.

In recent years, it has been described the relationship between DNA methylation and gene expression and the study of this relationship is often difficult to accomplish. This case study will show the steps to investigate the relationship between the two types of data.

We have used DNA methylation, RNA-Seq, & Clinical data for our analysis. The TCGA portal allows us to download Bisulfite sequencing data as well as array base data. We retrieved data of type, Methylation- Bisulfite Sequencing in Level 3 which had .bed format files of whole genome methylation which calls for each CpG site, per sample. Clinical data included compiled patient clinical information for each cancer study.

2 MATERIALS AND METHODS

2.1 DATA DESCRIPTION

We focused on Pancreatic Adenocarcinoma [1], as our endpoint. We downloaded DNA methylation data for HumanMethylation450k platforms, RNA expression data for IlluminaGA_RNAHiSeqV2, and clinical patient data. These three types of data have been obtained via TCGA BioLinks [2]. i.e. the Pancreatic Adenocarcinoma clinical data with 185 patients with five covariates of our interests being age($\leq$65, $>$65), smoking history (Yes or No), pathological stage(IA ,IB ,IIA, IIB, and IV, we focus on total I , total II and later stage), gender (males or female), histological type (Pancreas-Adenocarcinoma Ductal Type, and benign type), and the DNA Methylation data (TCGA level 3 data) with 40 samples (38 valid samples and 2 replicated samples) of total 485512 features (135476 type I probes and 350036 type II probes, we only focus on type I probe), RNA sequencing data (normalized) with 20531 genes of 183 patients (20502 genes of 178 patients, after matching for clinical data).

For partial analysis (using just one batch of samples), we used data.frame as our data format; for full analysis (using all batches on TCGA data portal), we used SummarizedExperiment object as our data format.

2.2 PILOT ANALYSIS

The three types of data have been prepared via TCGA BioLink. For DNA methylation data we only focus on type I probes, which yields 135476 features. For RNA sequencing data, we need to remove non-expression features by selecting barcodes. For clinical data, we have matched to the methylation data to identify reduced features, which the results is shown as Table 1:

Table 1 Summary of the cut-off adopted in clinical covariates

Covariates of Our Interest	Initial Diagnosis (Matched to Methylation Data)		Initial Diagnosis (Unmatched data)	
age	$\leq$65	$>$65	$\leq$65	$>$65
	21	17	96	89
Smoking history	Smokers	Non-smokers	Smokers	Non-smokers
	16	17	69	80
Histological type	Pancreatic Ductal	Pancreatic Other Subtype	Pancreatic Ductal	Pancreatic Other Subtype
	31	6	154	25
Pathological Stage	Stage I	Stage II+	Stage I	Stage II+
	3	35	21	162
Gender	Male	Female	Male	Female
	25	13	83	102

Specifying cut-off of each covariate. For age, we have used media of the sample age distribution as a cut-off, i.e. 65 yrs.; for smoking history, we simply categorized “yes” or “no” to reduce our predictor; for histological type, we focused on those with Pancreatic Ductal Adenocarcinoma as against of Other subtype; for Pathological stage, stage IA and IB are categorized jointly as stage I and stage IIA, IIB, stage III and IV are categorized jointly as stage II+; for gender, simply consider male and female.

Supervised statistical analysis has been performed to identify gene features or RNA transcripts which are differentially expressed between covariates of our interests and that might serve as classifiers or yield statistical significant results. P-values have been obtained by using the Kruskal-Wallis test [3] for comparisons across all covariates among the methylation and RNA sequencing data. We expected to select top 100 significant results.

Adjustment for multiple testing has been applied using the Benjamini-Hochberg False Discovery Rate (FDR) [4].

2.3 FOLLOW-UP ANALYSIS

Based on the pilot analysis, we found that we had some interesting results for RNA-seq data set but the results for methylation data did not agree. It may be cause of the limited amount of samples in the methylation data. Then we downloaded the full batches of sample (5.99 GB) for methylation array and performed a follow-up analysis. Note that in follow-up analysis we use non-normalized RNA-seq data.

2.3.1 DNA methylation

A DMR(differentially methylated region) [6] analysis was conducted, which gave the difference of DNA methylation for the probes of the groups and their significance value. We aim to find differentially methylated CpG sites, which are regarded as possible functional regions involved in gene transcriptional regulation. The output can be seen in a volcano plot.

In order to find these regions we use the beta-values (methylation values ranging from 0.0 to 1.0) to compare different subgroups.

1. Calculates the difference between the mean methylation of each group for each probes.
2. Calculates the p-value using the Wilcoxon test [5] and adopts the Benjamini-Hochberg adjustment (FDR) method. Set a minimum absolute beta values delta of 0.2 and a false discovery rate (FDR)-adjusted Wilcoxon rank-sum P-value of < 0.05 for the difference.
3. Create a volcano plot to identify the differentially methylated CpG sites.

2.3.2 Expression analysis

A DEA (differential expression analysis) was conducted, which will give the fold change of gene expression and their significance value by using edgeR package.

1. Use Array Array Intensity correlation (AAIC) to define outlier.
2. Perform normalization for RNA-Seq data by adjusting for GC-content effect.
3. Filter mRNA transcripts, selecting a threshold, we set quantile cut = 0.25.
4. The exact test proposed by Robinson and Smyth (2008) [7] is adopted for a difference in mean between two groups of negative binomial random variables.
5. Create a volcano plot to identify the differentially methylated CpG sites.

2.4 INTEGRATION THE TWO DATA TYPES

We use both previous analysis and do a starburst plot to select the genes that are Candidate Biologically Significant. Create Starburst plot for comparison of DNA methylation and gene expression. The log10 (FDR-corrected P value) is plotted for beta value for DNA methylation and gene expression for each gene.

3 RESULTS

3.1 PILOT ANALYSIS

Pilot analysis is conducted for identify the differential expressed gene. Figure 1 shows the analysis results by Kruskal-Wallis p-values. For all the four plots in Figure 1, the points appear in the lower left corner means that the corresponding genes have shown to be significantly expressed from both RNA-seq data and methylation data at level = 0.05. We can see that for different age groups there is no significantly differential expression neither at RNA-seq nor DNA methylation. For other three covariates, some gene differentially expressed for RNA-seq data, especially for different histological types and pathologic stages. It is interesting that none of these four covariates show any significantly differential expressions which conflict with the results we see in RNA-seq.

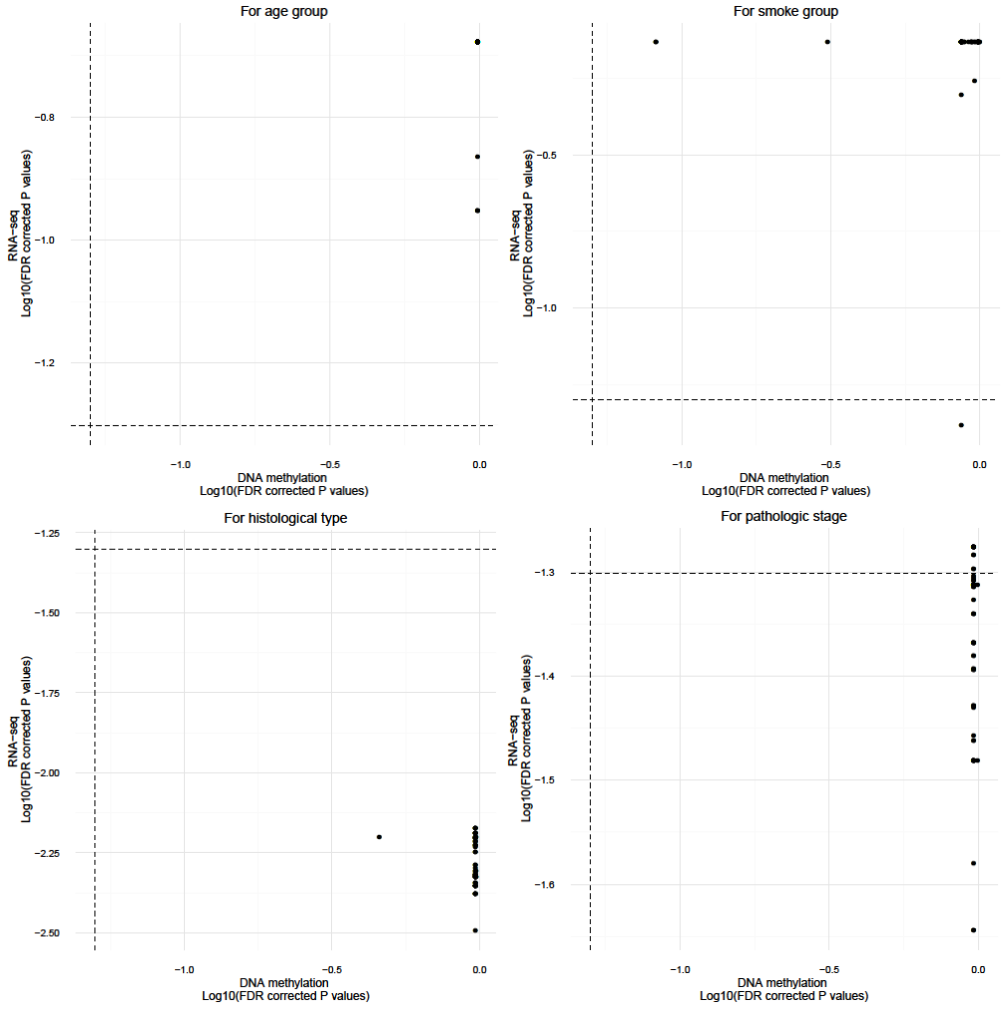

Figure 1: Comparison plot for p-values of RNA-seq and methylation analysis for four covariates. The x-axes correspond to the log10 (FDR-corrected P value) for beta value for DNA methylation (match from

the top 100 p-values for expression) and the y-axes correspond to the log10 (FDR-corrected P value) for gene expression (the top 100). The black dashed line shows the FDR-adjusted P value of 0.05.

3.2 FOLLOW-UP ANALYSIS

3.2.1 RNA-seq data

For follow-up analysis, we further evaluate the gene expression fold change by Volcano plot by TCGA biolink package. Depending on the fold change (x-axis) which is an indication of gene expression levels, we can set a fold change (FC) threshold to indicate that up or down regulated genes. In Figure 2, we set the Log2 threshold as 0.3. By volcano plots, we can easily find which genes differentially expressed for each covariate (subgroups). For example, we can identify 16 up-regulated differential expressed genes (log fold change > 3.0 and FDR < 0.05), which were represented in Figure 2 (upper left) regarding to age group.

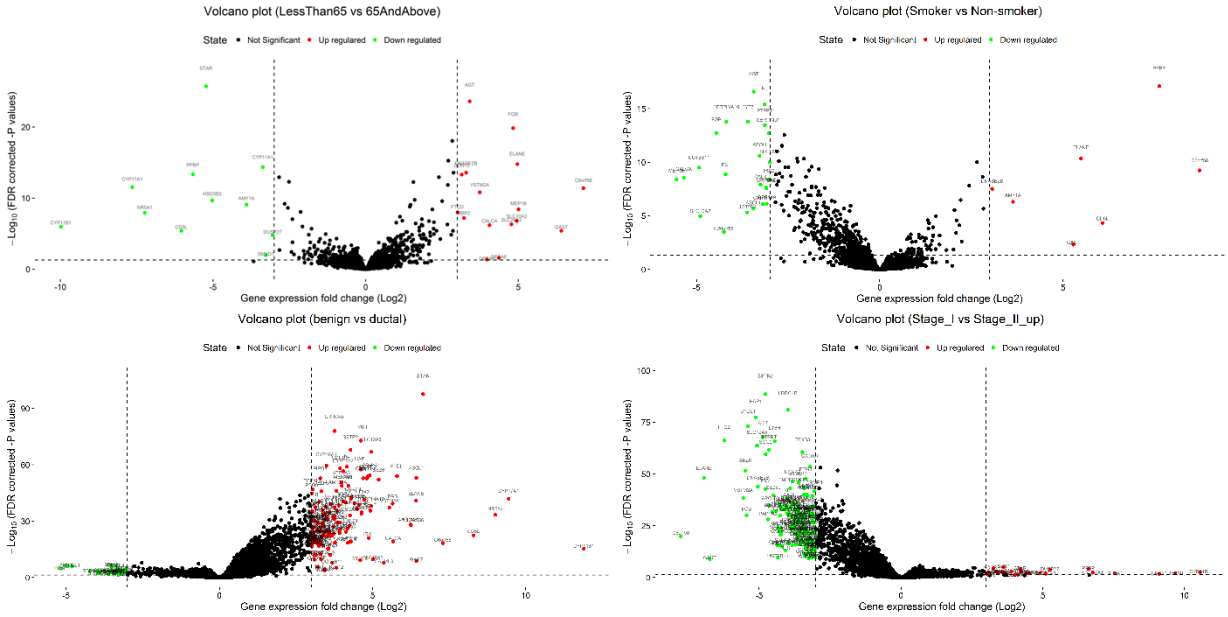

Figure 2: Volcano plots of differential expression genes for four covariates. The x-axes correspond to the fold change (FC) of expression data (Log2), and the y-axes correspond to significance of gene expression (Log10 FDR corrected p-value). The horizontal dashed lines show the FDR-adjusted P value of 0.05 and the vertical dashed lines show the fold change Log2 threshold of 0.3. The color code correspond to different types of regulated genes.

3.2.2 Methylation data

For methylation analysis, we first calculate the mean DNA methylation per group for all four covariates, and create a mean DNA methylation boxplot using the function TCGAvisualize_meanMethylation. Form Figure 3, only histological type and pathologic stage show some different between-group patterns. Gender are evenly distributed per group for all covariates.

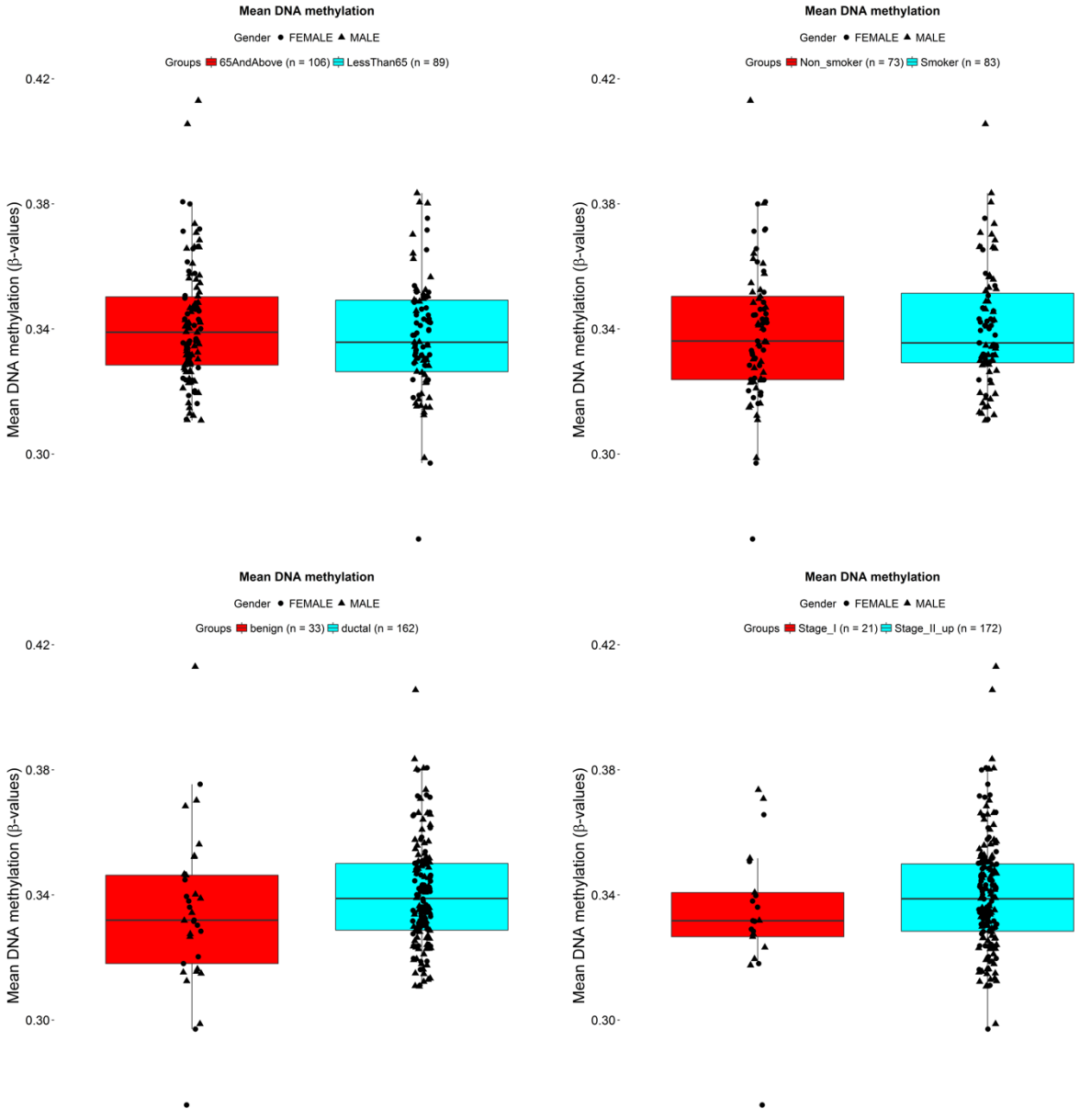

Figure 3: Boxplot for grouped mean DNA methylation. The y-axes correspond to the beta values for methylation.

Next, we do a DMR (differentially methylated region) analysis, which will give the difference of DNA methylation for the probes of the groups and their significance value. Distribution of significant probes based on their relationship to CpG Islands. From Figure 4, there is no significant probability for the difference in both age group and smoking group, which consistent with our pilot analysis. While for some we can identify some probes (hypermethylated as in red and hypomethylated as in green) significantly differential methylated for histological type and pathologic stage.

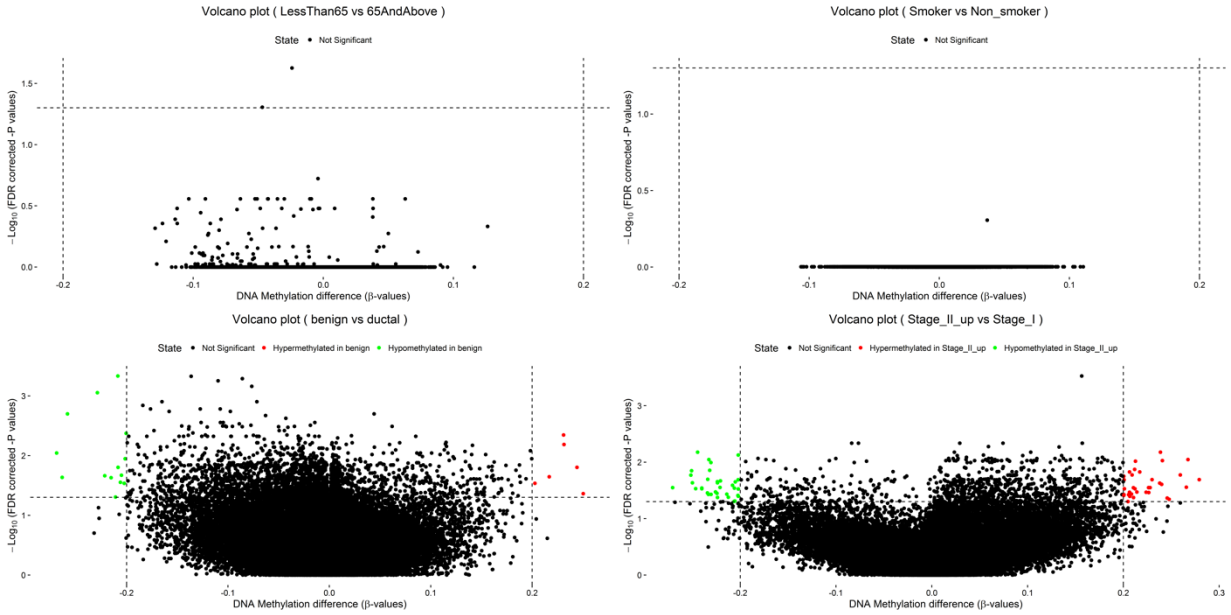

Figure 4: Volcano plots of DNA methylation for four covariates. The x-axes correspond to DNA methylation (beta value), and the y-axes correspond to significance of methylation (Log10 FDR corrected p-value). The horizontal dashed lines show the FDR-adjusted P value of 0.05 and the vertical dashed lines show the beta value threshold as 0.2. The color code correspond to different types of methylation genes.

Combining both RNA-seq and methylation analysis, we can select the genes that are Candidate Biologically Significant by starburst plot. The starburst plot was introduced to illustrate the results of integrating DNA methylation and gene expression data. From Figure 5, for different age group and smoking group, we hardly see any significant genes associated with discriminant subgroups in terms of methylation analysis. We only find two hypo-methylated genes which are not significantly expressed yet for age group.
For histological type, at top left corner, there are a lot of genes showed a difference in DNA methylation greater than 0.25 beta-value and a log2 FC greater than 3.0 (up-regulated and hypo methylated) between benign and ductal cancer subtypes. For different researchers, they can look at the four corners of the starburst plot to find their interested genes. In contrast to histological type, for pathologic stage, there are many down-regulated and hyper methylated genes differential methylated between stage I and later stage.

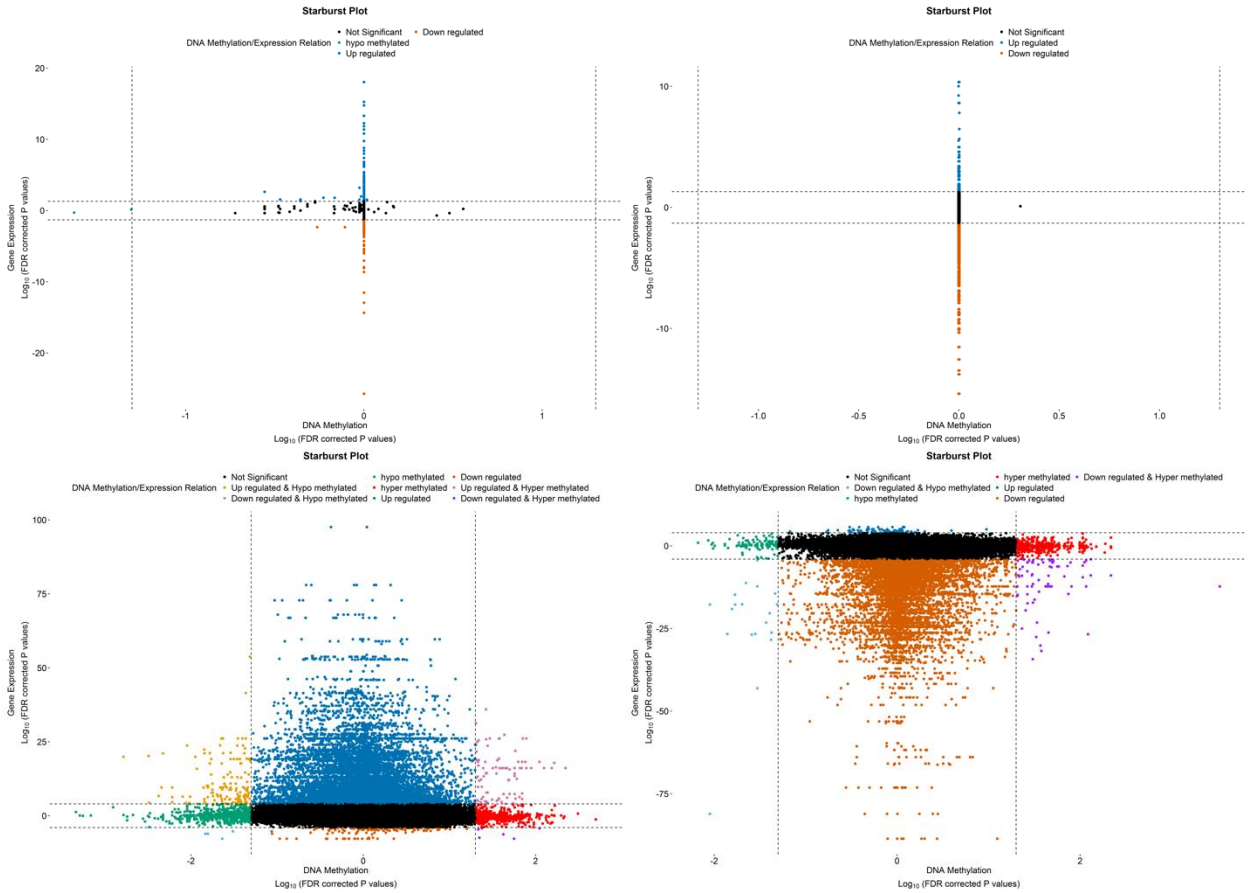

Figure 5: Starburst plot for DNA methylation significance versus gene expression significance. The x-axes correspond to the log10 of the correct p-value for DNA methylation and the y-axis is the log10 of the correct p-value for the expression data. The starburst plot highlights nine distinct quadrants. The color code for each quadrants correspond to different combinations of types of regulation and methylation.

4 DISCUSSION

This study can be considered as two-step analysis. Due to the huge amount of the methylation data, at first we only use 38 samples for methylation analysis, by the contrast, we use 183 samples to conduct expression analysis. Also, we use the results (Kruskal-Wallis p-values) of expression analysis to filter some significant genes then map to the gene symbols of methylation data. However, this approach seems not work when we compare the results between the RNA-seq and methylation data. None of covariates (subgroups) are significant for methylation analysis (even the difference between two subgroups is significant from RNA-Seq data), and the p-values are quite discrete which also concentered on large amount of point mass. We speculate that it might be because we have the limited samples in methylation data which lead to the poor statistical power.

For further identify the genes associate with certain subtype, we expand our methylation sample size and match them to the RNA-seq data. Then we have reasonable results for methylation data which are consistent to the expression data. It partially confirms our previous guess.

When looking at Figure 1 and Figure 2, there are some discrepancies regarding the p-values for expression and methylation analyses. For example, in pilot analysis, we cannot find any of significantly differential expressed genes for age group. However we actually found some of significant genes in follow-up study at the same level. That might be because we use different test methods (Kruskal-Wallis vs. Wilcoxon test) for both analyses. Moreover, it is possible that the pre-processing procedures are different (e.g. different normalization methods or if the filter applied).

For future work, there are a few possible extensions. In our follow-up study, we use full data set which contains multiple batches. We can further adjust the batch effect for integration analysis which is an interesting area for bioinformatics studies. Furthermore, we only use one single cancer (PAAD) in our case, so we can extend the integration analysis multiple cancer and tissue types.

5 REFERENCES

[1]. Pancreatic Adenocarcinoma Case Counts; The Cancer Genome Atlas; website https://tcga-data.nci.nih.gov/tcga/tcgaCancerDetails.jsp?diseaseType=PAAD&diseaseName=Pancreatic%20adenocarcinoma
[2]. Bioconductor: TCGA BioLinks website http://bioconductor.org/packages/release/bioc/html/TCGAbiolinks.html
[3]. The Kruskal–Wallis test; https://en.wikipedia.org/wiki/Kruskal%E2%80%93Wallis_one-way_analysis_of_variance
[4]. Benjamini, Yoav; Hochberg, Yosef (1995) "Controlling the false discovery rate: a practical and powerful approach to multiple testing"Journal of the Royal Statistical Society, Series B 57 (1): 289–300.
[5]. Wilcoxon signed-rank test Wikipedia; https://en.wikipedia.org/wiki/Wilcoxon_signed-rank_test.
[6]. Differentially methylated regions; Wikipedia; https://en.wikipedia.org/wiki/Differentially_methylated_regions
[7]. Robinson, MD; Smyth,GK "Small-sample estimation of negative binomial dispersion, with applications to SAGE data". Biostatistics, 9, 321-332.

APPENDIX

```
##########################################
# Download and prepare the data
#
# Data description:
#   PAAD.met - PAAD methylation data - 485577 by 28 (25 samples)
#   PAAD.exp - PAAD RNA-seq data - 20531 by 183 (full samples)
#   clinical_paad_data - clinical data - 24 by 78 (24 samples or full samples)
##########################################

##########################################
# NOTE:
#   We only need to match the samples by the first 3 digits
#   of barcode (eg. TCGA-F2-7276)
##########################################

rm(list=ls())
# source("http://bioconductor.org/biocLite.R")
# biocLite(c("TCGAbiolinks","SummarizedExperiment"))

library(TCGAbiolinks)
library(SummarizedExperiment)

#-------------------------------------------------------
# 1. Download DNA methylation data with TCGAbiolinks
#-------------------------------------------------------
DownloadFlag=F  # flip this flag to T to download data from TCGA

if(DownloadFlag) {
  path <- "paad"
  batchsample <- "TCGA-2J-AAB1-01"

  query <- TCGAquery(tumor = "PAAD", level = 3, platform = "HumanMethylation450", sample =
batchsample)

  # How many sample we have?
  length(unlist(strsplit(query$barcode,",")))

  # Download the TCGA data
  TCGAdownload(query, path =path)

  df=T
  if(df){
    PAAD.met <- TCGAprepare(query, dir = path,
                            save = TRUE,
                            filename = "metPAAD.rda",summarizedExperiment = F,
                            add.subtype = TRUE)
  } else {
    # summarizedExperiment version (default)
    PAAD.met <- TCGAprepare(query, dir = path,
                            save = TRUE,
                            filename = "metPAAD.rda",summarizedExperiment = T,
                            add.subtype = TRUE)
  }

  #-------------------------------------------------------
  # 2. download RNA expression data
  #-------------------------------------------------------
  query.rna <- TCGAquery(tumor="PAAD",level=3, platform="IlluminaHiSeq_RNASeqV2", sample =
batchsample)

  # How many sample we have?
  length(unlist(strsplit(query.rna$barcode,",")))

  TCGAdownload(query.rna,path=path,type = "rsem.genes.normalized_results")

  df=T
  if(df) {
    PAAD.exp <- TCGAprepare(query.rna, dir=path, save = TRUE,
```

```
                            type = "rsem.genes.normalized_results",
                            filename = "expPAAD.rda", summarizedExperiment = F,
                            add.subtype = TRUE)
  } else {
    PAAD.exp <- TCGAprepare(query.rna, dir=path, save = TRUE,
                            type = "rsem.genes.normalized_results",
                            filename = "expPAAD.rda", summarizedExperiment = T,
                            add.subtype = TRUE)
  }

  #
  # Add a CrossReact column to the the data frame
  # also add a SNP column

  # load nonSpecific
  nonSpecificTable=read.table("Final project/RData/nonspecific probes
Illumina450k.table",sep="\t",header = T)

  # load SNP
  load("Final project/RData/SNPinfo.RData")

  # Identify which records to take from SNPinfo
  SNPflag=(rowSums(!is.na(SNPinfo[,3:6]))!=0)
  badSNPs=rownames(SNPinfo)[SNPflag]
  matchDX=match(badSNPs,rownames(PAAD.met))

  PAAD.met$SNPflag=F
  PAAD.met$SNPflag[matchDX]=T

  if(F){ # double check our work
    CheckBad=rownames(PAAD.met)[which(PAAD.met$SNPflag)]
    CheckMatch=match(CheckBad,rownames(SNPinfo))
    SNPinfo[CheckMatch,]
  }

  # now, create a CrossReact flag
  matchDX=match(nonSpecificTable[,1],rownames(PAAD.met))
  PAAD.met$CrossReactflag=F
  PAAD.met$CrossReactflag[matchDX]=T

 # type I/II probe designation
  require(IlluminaHumanMethylation450kanno.ilmn12.hg19)
  data(IlluminaHumanMethylation450kanno.ilmn12.hg19)

  class(IlluminaHumanMethylation450kanno.ilmn12.hg19)

  MyAnn=getAnnotation(IlluminaHumanMethylation450kanno.ilmn12.hg19)
  matchDX=match(MyAnn$Name,rownames(PAAD.met))

  PAAD.met$Type=NA
  PAAD.met$Type[matchDX]=MyAnn$Type

  #-----------------------------
  # We can obtain clinical data
  # by TCGAquery_clinic
  #-----------------------------

  clinical_paad_data_all <- TCGAquery_clinic("paad","clinical_patient")
  clinical_paad_data <-
TCGAquery_clinic("paad","clinical_patient",sample=unlist(strsplit(query$barcode,",")))

  save.image(file="Final project/RData/LoadData.RData")

}

if(!DownloadFlag) load("Final project/RData/LoadData.RData")
```

```
################################
### Pre-Processing Data
################################

library(TCGAbiolinks)
library(SummarizedExperiment)
library(ggplot2)
rm(list=ls())

load("Final project/RData/LoadData.RData")

#################################################
# Look at our data set first, and manipulation
# these can be used in our data decription
#-------------------------------------------------
# ## Note: we have 3 dataset:
#         1. Methylation data
#         2. RNA-seq data
#         3. Clinical data
#################################################

#--------------
# Methylation
#--------------
# See how many type I we have in met data
table(PAAD.met$Type)

# select Type I only (met)
paad.met <- subset(PAAD.met,Type=="I")

#--------------
# RNA-seq
#--------------
# remove unknown gene (exp)
J=dim(PAAD.exp)[1]
genenames=rep("",J)
for(j in 1:J) genenames[j]=unlist(strsplit(rownames(PAAD.exp)[j],"\\|"))[1]
rows.to.keep <- which(genenames!="?")
paad.exp <- PAAD.exp[rows.to.keep,]

#--------------
# Clinical
#--------------

# look at the data
# matched to met data
attach(clinical_paad_data)
table(age_at_initial_pathologic_diagnosis)
table(tobacco_smoking_history)
table(pathologic_stage)
table(histological_type)
table(gender)

median(as.numeric(age_at_initial_pathologic_diagnosis))

detach(clinical_paad_data)

# unmatched (all samples)
attach(clinical_paad_data_all)
table(age_at_initial_pathologic_diagnosis)
table(tobacco_smoking_history)
table(pathologic_stage)
table(histological_type)
table(gender)

median(as.numeric(age_at_initial_pathologic_diagnosis))

detach(clinical_paad_data_all)

# Recode the covariates by making cut-off
# Age group
```

```
clinical_paad_data$agegrp <- ifelse(clinical_paad_data$age_at_initial_pathologic_diagnosis > 65,
1,0)
clinical_paad_data_all$agegrp <-
ifelse(clinical_paad_data_all$age_at_initial_pathologic_diagnosis > 65, 1,0)

# Smoking group
clinical_paad_data$smoking <- ifelse(as.numeric(clinical_paad_data$tobacco_smoking_history) <= 1,
0, 1)
clinical_paad_data_all$smoking <-
ifelse(as.numeric(clinical_paad_data_all$tobacco_smoking_history) <= 1, 0, 1)

# pathologic_stage
clinical_paad_data$path_stage[clinical_paad_data$pathologic_stage=="Stage I"|
                              clinical_paad_data$pathologic_stage=="Stage IA"|
                              clinical_paad_data$pathologic_stage=="Stage IB"] <- "Stage I"

clinical_paad_data$path_stage[clinical_paad_data$pathologic_stage=="Stage II"|
                              clinical_paad_data$pathologic_stage=="Stage IIA"|
                              clinical_paad_data$pathologic_stage=="Stage IIB"] <- "Stage II"

clinical_paad_data$path_stage[clinical_paad_data$pathologic_stage=="Stage III"|
                              clinical_paad_data$pathologic_stage=="Stage IV"] <- "Stage III &
IV"

clinical_paad_data_all$path_stage[clinical_paad_data_all$pathologic_stage=="Stage I"|
                                  clinical_paad_data_all$pathologic_stage=="Stage IA"|
                                  clinical_paad_data_all$pathologic_stage=="Stage IB"] <-
"Stage I"

clinical_paad_data_all$path_stage[clinical_paad_data_all$pathologic_stage=="Stage II"|
                                  clinical_paad_data_all$pathologic_stage=="Stage IIA"|
                                  clinical_paad_data_all$pathologic_stage=="Stage IIB"] <-
"Stage II"

clinical_paad_data_all$path_stage[clinical_paad_data_all$pathologic_stage=="Stage III"|
                                  clinical_paad_data_all$pathologic_stage=="Stage IV"] <-
"Stage III & IV"

# histological_type
clinical_paad_data$hist_type <- ifelse(clinical_paad_data$histological_type =="Pancreas-
Adenocarcinoma Ductal Type", 1,0)
clinical_paad_data_all$hist_type <- ifelse(clinical_paad_data_all$histological_type =="Pancreas-
Adenocarcinoma Ductal Type", 1,0)

## Note: we need recode other covariates..

#-----------------------------------
# Manipulate 3 data set for analysis
#-----------------------------------
# match clinical sample to exp data (183/185)
sample.exp <-  substr(colnames(paad.exp),1,12)

sample.cli <- clinical_paad_data_all$bcr_patient_barcode

length(sample.exp)  # 183
length(unique(sample.exp)) # 178
length(sample.cli) # 185
length(unique(sample.cli)) # 185

## note: there are 183-178=5 replicated patients (maybe multiple samples per patient)
## we need to remove them

sample.select <- unique(intersect(sample.exp,sample.cli))

paad.cli <- clinical_paad_data_all[which(clinical_paad_data_all$bcr_patient_barcode %in%
sample.select),]
paad.rna <- paad.exp[,-which(duplicated(sample.exp))]

# then sort by barcode
paad.cli <- paad.cli[order(paad.cli$bcr_patient_barcode),]
```

```
colnames(paad.rna) <- substr(colnames(paad.rna),1,12)

paad.rna <- paad.rna[, order(names(paad.rna))]

# Note: so far we have 3 dataset, include info in our data description:
#         1. Methylation data (paad.met) 38/40
#         2. RNA-seq data (paad.rna) 178
#         3. Clinical data (paad.cli) 178

# save.image("Final project/RData/ReadyToAnalysis.RData")

load("Final project/RData/ReadyToAnalysis.RData")

#-----------------------------------------------
# Methylation
#   - Filter by previous RNA-seq results
#   - Build a linear model for all 5 covariates
#-----------------------------------------------

#----------------------------------
# Load p-values and rank them
#----------------------------------

load("Final project/RData/pval_lm_rna.RData")
load("Final project/RData/pval_rna.RData")

# adjust by FDR first
myPvals_Lm <- p.adjust(myPvals_Lm, method="fdr", n=length(myPvals_Lm))

# rank p-values
bestK=order(myPvals_Lm)[1:100]

# find the most significant gene
bestK.gene=rep("",100)
for(j in 1:100) bestK.gene[j]=unlist(strsplit(rownames(paad.rna[bestK,])[j],"\\|"))[1]

# map to methylation data
paad.met.best <- subset(paad.met, Gene_Symbol %in% bestK.gene)

#-----------------------------------------------
# use the "best" set of methylation data to test
#-----------------------------------------------
nrow(na.omit(paad.met.best)) # 849
paad.met.best <- na.omit(paad.met.best)

# not run
paad.met.best.b <- na.omit(paad.met.best)[,-c(1:3,44:46)]

# match clinical sample to exp data (38/40)
sample.met <-  substr(colnames(paad.met.best.b),1,12)
sample.flag <- unique(intersect(sample.met,sample.cli))
paad.cli.match <- paad.cli[which(paad.cli$bcr_patient_barcode %in% sample.flag),]

# remove duplicate patitent
paad.met.best.b<- paad.met.best.b[,-which(duplicated(sample.met))]
paad.met.best.match <- paad.met.best.b[,which(substr(colnames(paad.met.best.b),1,12) %in%
sample.flag)]

# then sort by barcode
paad.cli.match <- paad.cli.match[order(paad.cli.match$bcr_patient_barcode),]
colnames(paad.met.best.match) <- substr(colnames(paad.met.best.match),1,12)
paad.met.best.match <- paad.met.best.match[, order(names(paad.met.best.match))]

# save and load 2nd manipulated data
# save(paad.met.best,paad.met.best.match,paad.cli.match,file="Final
project/RData/ReadyToAnalysis_lm.RData")

dim(paad.met.best.match)
dim(paad.cli.match)

load("Final project/RData/ReadyToAnalysis_lm.RData")
```

```
# adopt previous code
age.grp <- as.factor(paad.cli.match$agegrp)
smoking <- as.factor(paad.cli.match$smoking)
path.stage <- as.factor(paad.cli.match$path_stage)
hist.type <- as.factor(paad.cli.match$hist_type)
gender <- as.factor(paad.cli.match$gender)
#-----------------------------------------------
# Do a lm scan
#-----------------------------------------------
X=as.matrix(paad.met.best.match)
myLmp=function(i){
  cat(i,"...",fill=F)
  lm <- lm(X[i,]~age.grp + smoking + path.stage + hist.type)
  f <- summary(lm)$fstatistic
  p <- pf(f[1],f[2],f[3],lower.tail=F)
  attributes(p) <- NULL
  return(p)
}
myPvals_Lm_met=mapply(myLmp,1:(dim(X)[1]))
myPvals_Lm_met <- p.adjust(myPvals_Lm_met, method="fdr", n=length(myPvals_Lm_met))
DMR <- data.frame(Gene_Symbol=paad.met.best$Gene_Symbol, pval=myPvals_Lm_met)
GeneList.met <- unique(DMR$Gene_Symbol)

Gpval.met <- c()
for (i in 1:length(GeneList.met)) {
  idx = which(DMR$Gene_Symbol %in% GeneList.met[i])
  Gpval.met[i] <- min(DMR[idx,]$pval)
}

DMR.g <- data.frame(Gene_Symbol=GeneList.met, pval=Gpval.met)

#-----------------------------------------------
# Match and compare met vs. rna
#-----------------------------------------------
match.idx <- match(GeneList.met,bestK.gene)
bestK.gene[match.idx]
Gpval.rna <- myPvals_Lm[bestK][match.idx]
df <- data.frame(Gene_Symbol=DMR.g$Gene_Symbol, pval.met=DMR.g$pval, pval.rna=Gpval.rna)
p.lm <- ggplot(data = df, aes(x = log10(df$pval.met), y = log10(df$pval.rna))) +
  geom_point() +
  geom_hline(aes(yintercept = log10(0.05)), linetype = "dashed") +
  geom_vline(aes(xintercept = log10(0.05)), linetype = "dashed") +
  xlab("DNA methylation \nLog10(FDR corrected P values)") +
  ylab("RNA-seq \nLog10(FDR corrected P values)") +
  theme_minimal()
x11()
print(p.lm)

pdf("p_lm.pdf")
print(p.lm)
dev.off()

#-----------------------------------------------
# RNA-seq
# Do a quick kruskal-wallis scan
# Marginal test for each covariates
#-----------------------------------------------
load("Final project/RData/ReadyToAnalysis.RData")
X=as.matrix(paad.rna)
#----
age.grp <- as.factor(paad.cli$agegrp)
myKrusk=function(i){
  cat(i,"...",fill=F)
  kruskal.test(x=X[i,],g=age.grp)$p.value
}
myPvals_age=mapply(myKrusk,1:(dim(X)[1]))

#----
smoking <- as.factor(paad.cli$smoking)
myKrusk=function(i){
```

```
  cat(i,"...",fill=F)
  kruskal.test(x=X[i,],g=smoking)$p.value
}

myPvals_smoke=mapply(myKrusk,1:(dim(X)[1]))

#----
path.stage <- as.factor(paad.cli$path_stage)
myKrusk=function(i){
  cat(i,"...",fill=F)
  kruskal.test(x=X[i,],g=path.stage)$p.value
}

myPvals_path=mapply(myKrusk,1:(dim(X)[1]))

#----
hist.type <- as.factor(paad.cli$hist_type)
myKrusk=function(i){
  cat(i,"...",fill=F)
  kruskal.test(x=X[i,],g=hist.type)$p.value
}
myPvals_hist=mapply(myKrusk,1:(dim(X)[1]))

#----
gender <- as.factor(paad.cli$gender)
myKrusk=function(i){
  cat(i,"...",fill=F)
  kruskal.test(x=X[i,],g=gender)$p.value
}

myPvals_gender=mapply(myKrusk,1:(dim(X)[1]))

# save(myPvals_age,myPvals_smoke,myPvals_path,myPvals_hist,myPvals_gender,file="Final
project/RData/pval_rna.RData")

#-----------------------------------------------
# Build a linear model for all 5 covariates
#-----------------------------------------------
myLmp=function(i){
  cat(i,"...",fill=F)
  lm <- lm(X[i,]~age.grp + smoking + path.stage + hist.type)
  f <- summary(lm)$fstatistic
  p <- pf(f[1],f[2],f[3],lower.tail=F)
  attributes(p) <- NULL
  return(p)
}
myPvals_Lm=mapply(myLmp,1:(dim(X)[1]))

# save(myPvals_Lm, file="Final project/RData/pval_lm_rna.RData")

#-----------------------------------------------
# Methylation
#   - Filter by previous RNA-seq results
#   - Do a quick kruskal-wallis scan
#     Marginal test for each covariates
#-----------------------------------------------
load("Final project/RData/pval_lm_rna.RData")
load("Final project/RData/pval_rna.RData")

##################
# age
##################
# adjust by FDR first
myPvals_age <- p.adjust(myPvals_age, method="fdr", n=length(myPvals_age))

# rank p-values
bestK=order(myPvals_age)[1:100]

# find the most significant gene
bestK.gene=rep("",100)
for(j in 1:100) bestK.gene[j]=unlist(strsplit(rownames(paad.rna[bestK,])[j],"\\|"))[1]
```

```
# map to methylation data
paad.met.best <- subset(paad.met, Gene_Symbol %in% bestK.gene)
#-----------------------------------------------
# use the "best" set of methylation data to test
#-----------------------------------------------
nrow(na.omit(paad.met.best))
paad.met.best <- na.omit(paad.met.best)
# not run
paad.met.best.b <- na.omit(paad.met.best)[,-c(1:3,44:46)]

# match clinical sample to exp data (38/40)
sample.met <-  substr(colnames(paad.met.best.b),1,12)
sample.flag <- unique(intersect(sample.met,sample.cli))
paad.cli.match <- paad.cli[which(paad.cli$bcr_patient_barcode %in% sample.flag),]

# remove duplicate patitent
paad.met.best.b<- paad.met.best.b[,-which(duplicated(sample.met))]
paad.met.best.match <- paad.met.best.b[,which(substr(colnames(paad.met.best.b),1,12) %in%
sample.flag)]

# then sort by barcode
paad.cli.match <- paad.cli.match[order(paad.cli.match$bcr_patient_barcode),]

colnames(paad.met.best.match) <- substr(colnames(paad.met.best.match),1,12)
paad.met.best.match <- paad.met.best.match[, order(names(paad.met.best.match))]

# save and load 2nd manipulated data
# save(paad.met.best,paad.met.best.match,paad.cli.match,file="Final
project/RData/ReadyToAnalysis_age.RData")
dim(paad.met.best.match)
dim(paad.cli.match)

load("Final project/RData/ReadyToAnalysis_age.RData")

# adopt previous code
age.grp <- as.factor(paad.cli.match$agegrp)
smoking <- as.factor(paad.cli.match$smoking)
path.stage <- as.factor(paad.cli.match$path_stage)
hist.type <- as.factor(paad.cli.match$hist_type)
gender <- as.factor(paad.cli.match$gender)
#-----------------------------------------------
# Do a age scan
#-----------------------------------------------
X=as.matrix(paad.met.best.match)
myKrusk=function(i){
  cat(i,"...",fill=F)
  kruskal.test(x=X[i,],g=age.grp)$p.value
}
myPvals_age_met=mapply(myKrusk,1:(dim(X)[1]))
myPvals_age_met <- p.adjust(myPvals_age_met, method="fdr", n=length(myPvals_age_met))
DMR <- data.frame(Gene_Symbol=paad.met.best$Gene_Symbol, pval=myPvals_age_met)
GeneList.met <- unique(DMR$Gene_Symbol)
Gpval.met <- c()
for (i in 1:length(GeneList.met)) {
  idx = which(DMR$Gene_Symbol %in% GeneList.met[i])
  Gpval.met[i] <- min(DMR[idx,]$pval)
}
DMR.g <- data.frame(Gene_Symbol=GeneList.met, pval=Gpval.met)
#-----------------------------------------------
# Match and compare met vs. rna
#-----------------------------------------------
match.idx <- match(GeneList.met,bestK.gene)
bestK.gene[match.idx]
Gpval.rna <- myPvals_age[bestK][match.idx]
df <- data.frame(Gene_Symbol=DMR.g$Gene_Symbol, pval.met=DMR.g$pval, pval.rna=Gpval.rna)
p.age <- ggplot(data = df, aes(x = log10(df$pval.met), y = log10(df$pval.rna))) +
  geom_point() +
  geom_hline(aes(yintercept = log10(0.05)), linetype = "dashed") +
  geom_vline(aes(xintercept = log10(0.05)), linetype = "dashed") +
  xlab("DNA methylation \nLog10(FDR corrected P values)") +
```

```
  ylab("RNA-seq \nLog10(FDR corrected P values)") +
  ggtitle("For age group") +
  theme_minimal()
x11()
print(p.age)

pdf("p_age.pdf")
print(p.age)
dev.off()

load("Final project/RData/pval_lm_rna.RData")
load("Final project/RData/pval_rna.RData")

#--- omit another covariates---#


##########################################
# For full data set
# use TCGAbiolink for analysis
##########################################
rm(list=ls())
load("RData/LoadData.RData")
load("RData/LoadData_cli.RData")
load("RData/LoadData_met_df.RData")

# attach clinical infomation to both data set
length(colData(paad.exp)$patient)
length(colData(paad.met)$patient)
length(paad.cli$bcr_patient_barcode)

names(paad.cli)[79:82]

matchDX= match(colData(paad.exp)$patient, paad.cli$bcr_patient_barcode)
matchDX_met= match(colData(paad.met)$patient, paad.cli$bcr_patient_barcode)

colData(paad.exp)$agegrp <- paad.cli$agegrp[matchDX]
colData(paad.exp)$smoking <- paad.cli$smoking[matchDX]
colData(paad.exp)$path_stage <- paad.cli$path_stage[matchDX]
colData(paad.exp)$hist_type <- paad.cli$hist_type[matchDX]
colData(paad.exp)$gender <- paad.cli$gender[matchDX]

colData(paad.met)$agegrp <- paad.cli$agegrp[matchDX_met]
colData(paad.met)$smoking <- paad.cli$smoking[matchDX_met]
colData(paad.met)$path_stage <- paad.cli$path_stage[matchDX_met]
colData(paad.met)$hist_type <- paad.cli$hist_type[matchDX_met]
colData(paad.met)$gender <- paad.cli$gender[matchDX_met]

#--------------------------------
# load nonSpecific
nonSpecificTable=read.table("RData/nonspecific probes Illumina450k.table",sep="\t",header = T)

# load SNP
load("RData/SNPinfo.RData")

# Identify which records to take from SNPinfo
SNPflag=(rowSums(!is.na(SNPinfo[,3:6]))!=0)
badSNPs=rownames(SNPinfo)[SNPflag]
matchDX=match(badSNPs,rownames(PAAD.met))

PAAD.met$SNPflag=F
PAAD.met$SNPflag[matchDX]=T

if(F){ # double check our work
  CheckBad=rownames(PAAD.met)[which(PAAD.met$SNPflag)]
  CheckMatch=match(CheckBad,rownames(SNPinfo))
  SNPinfo[CheckMatch,]
}

# now, create a CrossReact flag
matchDX=match(nonSpecificTable[,1],rownames(PAAD.met))
PAAD.met$CrossReactflag=F
```

```
PAAD.met$CrossReactflag[matchDX]=T

# type I/II probe designation
require(IlluminaHumanMethylation450kanno.ilmn12.hg19)
data(IlluminaHumanMethylation450kanno.ilmn12.hg19)

class(IlluminaHumanMethylation450kanno.ilmn12.hg19)

MyAnn=getAnnotation(IlluminaHumanMethylation450kanno.ilmn12.hg19)
matchDX=match(MyAnn$Name,rownames(PAAD.met))

PAAD.met$Type=NA
PAAD.met$Type[matchDX]=MyAnn$Type

# select Type I only (met)
# PAAD.met <- subset(PAAD.met,Type=="I")

# filter Type I
TypeI.DX <- which(PAAD.met$Type=="I")
PAAD.met <- subset(PAAD.met,Type=="I")
paad.met <- paad.met[TypeI.DX,]

# filter SNPflag
SNP.DX <- which(PAAD.met$SNPflag==F)
paad.met <- paad.met[SNP.DX,]
save(paad.cli,paad.met,paad.exp,query.met,paad.query.exp,file="RData/ManipulatedData.RData")

#-----------------------------
# Age group
#-----------------------------
rm(list=ls())
load("RData/ManipulatedData.RData")

############################################################
# removing the samples without subtype classification
paad.aux <- subset(paad.met,select = colData(paad.met)$agegrp %in% c("65AndAbove","LessThan65"))

############################################################
TCGAvisualize_meanMethylation(paad.aux,
                              groupCol = "agegrp",
                              subgroupCol = "gender",
                              group.legend  = "Groups",
                              subgroup.legend = "Gender",
                              filename = "paad_mean_agegrp.png")

# na.omit
paad.aux <- subset(paad.aux,subset = (rowSums(is.na(assay(paad.aux))) == 0))

# Volcano plot
paad.aux <- TCGAanalyze_DMR(paad.aux, groupCol = "agegrp",
                            group1 = "65AndAbove",
                            group2="LessThan65",
                            p.cut = 0.05,
                            diffmean.cut = 0.2,
                            legend = "State",
                            plot.filename = "paad_agegrp_met_volcano.png")

#----------------------
paad.exp.aux <- subset(paad.exp, select = colData(paad.exp)$agegrp %in%
c("65AndAbove","LessThan65"))

idx <- colData(paad.exp.aux)$agegrp %in% c("65AndAbove")
idx2 <- colData(paad.exp.aux)$agegrp %in% c("LessThan65")

dataPrep <- TCGAanalyze_Preprocessing(object = paad.exp.aux,
                                      cor.cut = 0.0)

dataNorm <- TCGAanalyze_Normalization(tabDF = dataPrep,
                                      geneInfo = geneInfo,
                                      method = "gcContent")
```

```
dataFilt <- TCGAanalyze_Filtering(tabDF = dataNorm,
                                  qnt.cut = 0.25,
                                  method='quantile')

dataDEGs <- TCGAanalyze_DEA(mat1 = dataFilt[,idx],
                            mat2 = dataFilt[,idx2],
                            Cond1type = "65AndAbove",
                            Cond2type = "LessThan65",
                            method = "glmLRT")

TCGAVisualize_volcano(dataDEGs$logFC,dataDEGs$FDR,
                      filename = "paad_agegrp_exp_volcano.png",
                      x.cut = 3,
                      y.cut = 0.05,
                      names = rownames(dataDEGs),
                      xlab = " Gene expression fold change (Log2)",
                      legend = "State",
                      title = "Volcano plot (LessThan65 vs 65AndAbove)")

#----------------------
starburst_agegrp <- TCGAvisualize_starburst(paad.aux, dataDEGs,"65AndAbove","LessThan65",
                                            filename = "paad_agegrp_starburst.png",
                                            met.p.cut = 0.05,
                                            exp.p.cut = 0.05,
                                            diffmean.cut = 0.25,
                                            logFC.cut = 3,
                                            names = TRUE)
```