Towards an Investigation of Lung Cancer Genes Using Multi omcis Approaches

Jihwan Lim and Inkyun Park

Supervisor: Prof. dr. Tim De Meyer, Prof. dr. Jo Vandesompele & Prof. dr. Wim Trypsteen

Counsellors: Louis Coussement & Menno Van Damme

Abstract. Modern omics-related study provides us enormous data from various techniques for better understanding into biological properties. Especially, combination of these different techniques, called multi-omics, leads deeper insights to specific biological study. This multi-omics approach was used to investigate lung cancer study in this report using four publicly available datasets: Microarray data, Infinium data, RNA-seq data, and ChIP-seq data.

Diverse insight toward molecular process of lung cancer could be explored with these four types of omics datasets. Statistically significant genes, overlapping results from different sources, were investigated, and visualized in this report as well

Keywords: Lung cancer, microarray, RNA sequencing, methylation profiling, ChIP sequencing, EGFR

1. Introduction

Top cancer responsible for the most deaths in United States is lung cancer, which is expected to produce 236,470 new patients and kill 130,180 patients in 2022. Also, patients with lung cancer more than 5 years show the lower survival rates (18.6 %) than other commonly occurring cancers, such as colorectal (64.5 %), breast (89.6 %), and prostate (98.2 %). On average, patients do not withstand more than one year after diagnosis of lung cancer (Nierengarten, 2022). Although there are lots of efforts in studying lung cancer due to these risks on clinical aspects, lung cancer is not fully understood yet. This is because of, for example, heterogeneity of tumor tissues; unpredictable mutation and inconsistent behavior of each cancer cell; or tendency to have resistant to chemotherapy agents eventually (Kim, 2016). With rise of sequencing technology and easiness of building -omics database, many methods using the genomic database are emerged to understand and treat lung cancer properly. In this paper, using open Gene Expression Omnibus (GEO) database, four different -omics technologies was combined to find out consistent results of lung cancer.

The transcriptomic aspect of lung carcinoma tissues is analyzed using microarray and RNA-seq data finding differentially expressed genes which are also statistically significant. For epigenomic aspect of lung carcinoma tissues was analyzed using methylation array and ChIP-seq data. For this point, epidermal growth factor receptor (EGFR) mutated lung cancer was analyzed since EGFR mutation, about 31.6% of non-small cell lung cancer (NSCLC), is one of the mostly tested for targeted therapy (Kumari et al., 2019). EGFR is a trans-membrane glycoprotein regulates signaling pathway of cell proliferation (Kumari et al., 2019), which leads to tumor tissue with its mutation. Three of datasets including Microarray data, Infinium data, and RNA-seq data were integrated to further understanding of considerable genes and corresponding biological pathway. The impact of treatment toward EGFR positive lung cancer, could also be analyzed integration of three different omics approach.

1.1 Data

First dataset was used for differential analysis is microarray data of invasive lung cancer tissues and its adjacent normal tissues from 6 patients. RNA was extracted from dissected tissues, and it was profiled using Illuminia Technologies Human Genome U133 Plus 2.0 Array. Raw data was acquired from Gene Expression Omnibus database with accession number of GSE118370 (Xu et al., 2018).

Next dataset is RNA sequencing data from GSE40419. In this database, 87 cancer tissues and 77 adjacent normal tissues with various tumor stages, gender, and smoking status were deposited. From these, 3 samples with stage 3 tumor and its adjacent normal tissues were selected for differential expression analysis. Sequencing was performed using Illumina HiSeq 2000 and results in paired-end 101-bp-long reads (Seo et al., 2012).

Methylation profiling data contains methylome of both healthy and tumor tissue from Norway patients. Since there are too many factors to be considered, only 3 mutated EGFR tumor tissue with stage 3 and wild type of both KRAS and TP53 genes and 3 normal tissues were selected which have same sentrix position with 3 EGFR mutated tumor tissues for analysis. These samples were profiled using Illumina HumanMethylation450 BeadChip (Bjaanaes et al., 2015).

Finally, ChIP-seq data contains 4 EGFR mutated lung cancer cell line PC9. 2 cancer samples are untreated. Other 2 samples are treated with erlotinib for 11 days. Even though, the study design of this data was looking into impact of treatment resistance tissue, we only looked into impact of treatment toward methylation peak. These samples were profiled using ChIP-seq with using H3K4me3 antibodies.

2. Methods

2.1 Microarray

Microarray data was first downloaded from the GEO database. All 6 patients with invasive lung cancer adenocarcinoma are selected for this analysis. Using *rma* function from *affy* package, raw data was preprocessed with three steps: background correction, quantile normalization and summarization. The effect of preprocessing was examined by *arrayQualityMetrics* packages for raw, log-transformed and *rma* preprocessed data. Differential expression analysis is performed by *limma* package afterwards. Lastly, differential expression results were annotated with biological pathway in GO terms by *gonna* function.

2.2 RNA sequencing

3 patients with stage 3 lung adenocarcinoma were selected and data was acquired from GEO database too. To prepare raw data into compatible form for differential analysis, trimming and mapping of raw FASTQ files are performed in the HPC server, due to high computational and memory burden of preprocessing of big FASTQ files. Since sequencing quality score is usually bad at the end of reads, *Trimmomatic* cuts the end of reads based on quality score to improve overall quality of reads. Then, Illumina adapter sequences are also removed. Next, trimmed reads are mapped into human reference transcriptome by *Kallisto*, a pseudoalignment tool with rapid and accurate mapping, but without the need for alignment. Results of preprocessing are assessed by *FastQC* and summarized by *MultiQC*. After all steps, abundance files, results from *Kallisto*, are imported to R using *tximport*.

Differential gene expression analysis was implemented by *edgeR* packages. First, genes with low counts are filtered based on count-per-million of each reads. Next step was usually normalization; however, differential expression analysis was conducted thereafter as normalization is already done in *Kallisto*. Then, the data is fitted to the likelihood ratio tests from *edgeR* to find out differentially expressed genes. After differential analysis, significant genes are annotated with biological pathway in GO terms by *gonna* function as described above.

2.3 Methylation array

Methylation profiling data was generated by genome tilling array, Illumina HumanMethylation450 BeadChip. Methylation profiles of 164 lung tumor samples and 19 matched normal samples are determined with Illumina Infinium 450K array. The *wateRmelon* and *lumi* packages are used for analysis of methylation data such as data preprocessing, quality control, and normalization. *limma* was used to identify differentially methylated positions. Gene set analysis was performed differentially methylated to get insight with related pathway using *goana* function.

2.4 ChIP sequencing

ChIP-seq data was generated by using Illumina HiSeq 2000 platform and H3K4me3 antibody. The non-small cell lung carcinoma (NSCLC) PC9 cell line was established for investigating the impact of H3K4 demethylase KDM5A protein which is related to resistance to cancer treatment. For better understanding of this clinical purpose, EGFR mutated PC9 cell lines were treated with Erlotinib for 11 days and compared with untreated samples. There are 2 biological replicates for both treated and untreated PC9 cell lines. These datasets are firstly aligned with using *bowtie2.* 2 control sample *bam* files and 2 treated sample *bam* files were combined into control *broadpeak* file and treated *broadpeak* file respectively using *macs2* after sorting and merging. These series of process for preprocessing data was done on HPC server and QC was preformed with *multiqc* before further analysis.

The *GenomicRanges* package was used to build input file of UCSC Genome Browser for further analysis using *broadpeak* file. *DiffBind* package was used to compare and analyze the regions that were differentially bind for each of treatment and untreated (control) samples. *DESeq2* and *edgeR* methods were used for getting the result of differential enrichment analysis.

3. Results

3.1 Microarray

Microarray data was preprocessed with *rma* function in *affy* packages. Background correction, quantile normalization and summarization were all conducted in one function. As shown in Figure 1, the effect of preprocessing is observed by seeing boxplot of the data. More quality control reports, and relevant codes can be found in appendix.

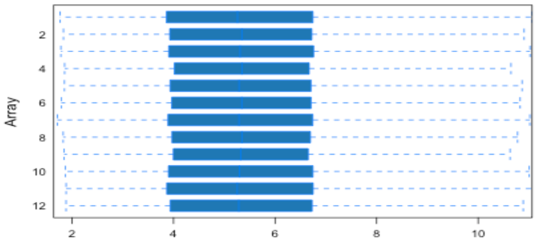


Figure 1. Boxplot of rma preprocessed microarray data, showing similar trends for all 12 samples.

Using *limma*, genes of tumor and adjacent normal samples were compared, and 4457 genes were found out to be differentially expressed. Table 1 shows list of top differentially expressed genes with fold change and p-value. Positive fold changes indicates that gene is highly expressed in the tumor tissue. MA plot in figure 2 also shows the results of *limma* differential expression analysis. Afterwards, differentially expressed genes were annotated with biological pathways in GO terms using gene set analysis function *gonna*. Top biological pathways with significant results are shown in table 2.

Table 1. Differentially expressed genes between tumor and adjacent normal tissues from microarray.

Gene Symbol	logFC	adj.P.Val
SPAAR	2.648723	6.10e-06
SLC6A4	6.115768	1.54e-05
RTKN2	4.058742	1.54e-05
TEK	3.012743	1.54e-05
SEMA6A	3.532686	1.54e-05

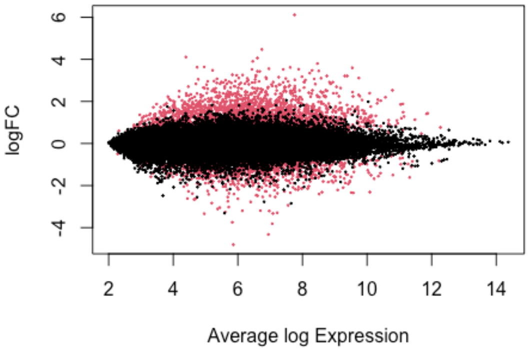


Figure 2. MA plot of differentially expressed genes between tumor and adjacent normal tissues. Genes with less than 0.05 adjusted p-values are depicted in red dots.

Table 2. Top 5 biological pathways from gene set analysis on microarray expression data.

GO Terms	Biological Pathway
GO:0007155	cell adhesion
GO:0009653	anatomical structure morphogenesis
GO:0016477	cell migration
GO:0048856	anatomical structure development
GO:0007275	multicellular organism development

3.2 RNA sequencing

Raw RNA-seq data was trimmed and mapped by *Trimmomatic* and *Kallisto* respectively. Higher mean quality scores and are observed after trimming. About 80% of sequences are aligned after pseudo alignment mapping by *Kallisto*. Then, differential expression analysis using *edgeR* found out 1296 genes and 1129 genes were down and up regulated respectively. A p-value and FDR distribution with peak around 0 show the analysis was performed in right manner. Table 3 shows top differentially expressed genes with high statistical meaning. Also, this can be found out in MA plot in figure 3 which shows down and up-regulated genes in red dots. More detailed results and codes for *edgeR* processing can be found in the appendix. Gene set analysis on differentially expressed gene was conducted and annotated with GO terms as shown in table 4.

Table 3. Differentially expressed genes between tumor and adjacent normal tissues from RNA-seq.

Gene Symbol	logFC	adj.P.Val
NR4A2	4.64032684	5.58E-22
TOP2A	-4.2623642	1.14E-19
SPP1	-7.1573626	3.14E-18
EGR3	3.89102764	1.00E-17
MMP12	-7.6385967	6.05E-15

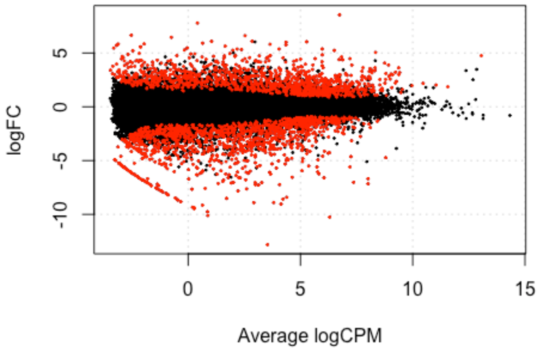


Figure 3. MA plot of differentially expressed genes between tumor and adjacent normal tissues. Genes with less than 0.05 adjusted p-values are depicted in red dots.

Table 4. Top 5 biological pathways from gene set analysis on RNA sequencing data.

GO Terms	Biological Pathway
GO:0000278	mitotic cell cycle
GO:1903047	mitotic cell cycle process
GO:0007049	cell cycle
GO:0022402	cell cycle process
GO:0048856	anatomical structure development

3.3 Methylation

Whole genome methylation profiles of 3 lung cancer and 3 normal tissues were analyzed using *wateRmelon*, *lumi*, and *limma.* Firstly, NA values in M-value were removed. Then, data that had beta values with insufficient detect p-value which is over

0.05 was removed as well. After preprocess of datasets, t-test was performed to compare average methylation percentage (beta-value) between normal lung samples and tumor samples. The p-value from this t-test was about 0.7174 which means that differences between two groups are not significant. For further analysis and QC, beta values were normalized to remove external effects like technical variations then both normalized dataset and unnormalized dataset were transformed as *Mehtylumi* form. Linear model was statistically designed with contrast between cancer and normal tissue to compare both tissues with the threshold defined by Benjamini–Hochberg procedure. This linear model was fit using empirical Bayes method to investigate which genes were differentially expressed between normal and tumor tissue samples.

Total 10011 CpG probes were detected as significantly (FDR < 0.05) differentially methylated between two groups with 7754 down regulated methylation and 2237 upregulated methylation. This differential methylation of CpG site was tested using *limma*. After several process such as annotation of genes and choosing CpG probes in genic regions, 7435 CpG probes were remained as significantly (FDR < 0.05) differentially methylated probes. Genes related to the regions of transcription start sites (TSS) and 1st exons were selected for analysis since these regions were linked to transcriptional silencing in cancer by methylation (Brenet et al., 2011). Finally, gene set analysis with CpG regions were performed to get the information of related biological pathway. Top hits for GO Biological Process terms include anatomical structure development, multicellular organism development, system development, and anatomical structure morphogenesis as shown in table 6.

Table 5. Differentially expressed genes between tumor and adjacent normal tissues from Infinium array data.

Gene	logFC	adj.P.Val
DPYS	-3.545936	0.0151440
KIAA0319	3.258170	0.0151440
GRM6	-3.428868	0.0182476
HOXD9	-3.559862	0.0200073
LOC100287834	3.652897	0.0200073

Table 6. Top 5 biological pathways from gene set analysis on Infinium array data.

GO Terms	Biological Pathway
GO:0048856	anatomical structure development
GO:0007275	multicellular organism development
GO:0048731	system development
GO:0032502	developmental process
GO:0007399	nervous system development

3.4 ChIP sequencing

2 untreated and 2 treated EGFR-mutated PC9 cell lines were analyzed using *GenomicRagnges* and *DiffBind*. Control *proadpeak* file and treated *broadpeak* file were respectively preprocessed and submitted to UCSC Genome Browser with right format for visualization. To perform differential enrichment analysis, *DiffBind* package was used.

After loading the file using *dba* function, count information of each (control and treated samples) could be identified using *dba.count.* With PCA plot and heatmap as shown in figure 4 and figure 5, close correlation between biological replicates could be identified. With this clustering of replicates, treatment is used as a factor and replicate is considered as block effect design contrast model. Statistically significantly differentially bound sites were analyzed using both *DESeq2* and *edgeR* methods. After this step, both replicates showed closer clustering. Venn diagram showed that *edgeR* method identified 2 times more peaks than *DESeq2* method. Untreated samples showed broader distribution of reads over all differentially bound site. Identified peaks from these analysis for each untreated and treated samples are generated as *bed* format respectively and submitted to UCSC Genome Browser to visualize significant peaks. The result of submission to UCSC Genome Browser of untreated sample *bed* file showed statistically significant peaks on CAECAM6, PECAM1, and DUSP6 gene regions as shown in figure 6. Whereas treated sample *bed*_file showed statistically significant peaks on NUPR1, OLR1, and NKAIN4 gene regions as shown in figure 7.

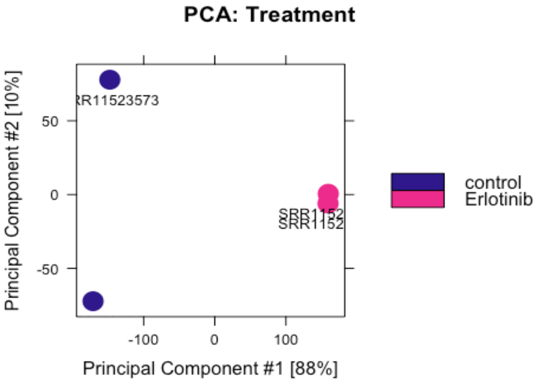


Figure 4. PCA plot between treated and untreated PC9 cell lines

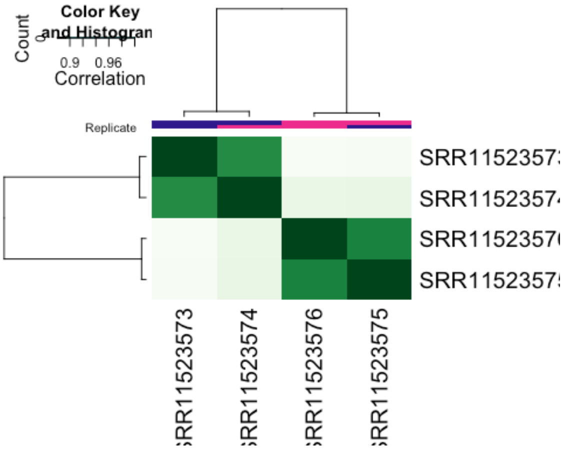


Figure 5. Heatmap between treated and untreated PC9 cell lines

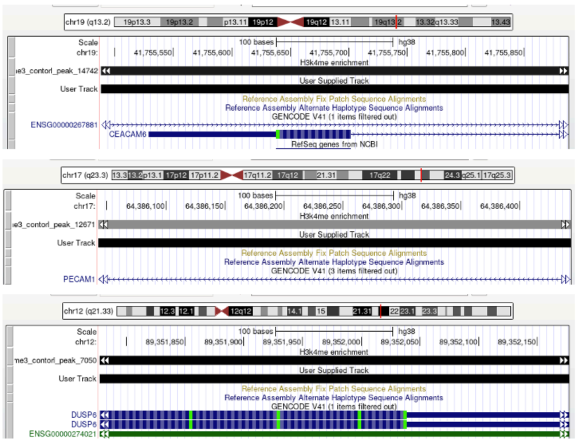


Figure 6. Specific methylation peak for untreat PC9 cell lines

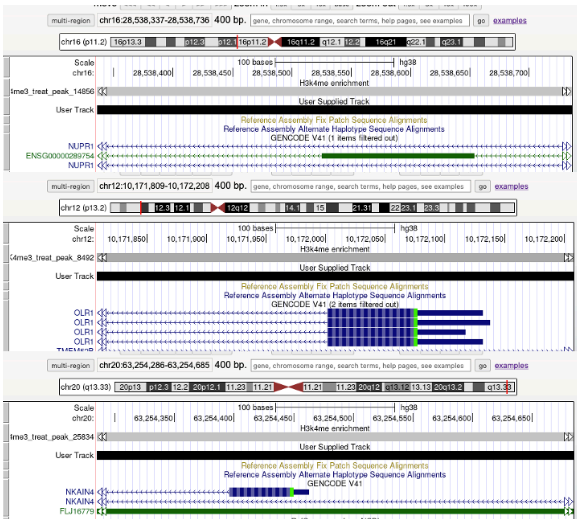

chr16:28,538,337-28,538,736 400 bp.
NUPR1
ENSG00000289754
NUPR1
chr12:10,171,809-10,172,208 400 bp.
OLR1
OLR1
OLR1
OLR1
chr20:63,254,286-63,254,685 400 bp.
NKAIN4
NKAIN4
FLJ16779

Figure 7. Specific methylation peak for treated PC9 cell lines

3.5 Data integration

The Results from microarray expression, RNA sequencing and methylation profiling were compared to see commonly expressed genes. Firstly, microarray and RNA sequencing data was combined and plotted on scatter plot of fold change of significant genes in both analyses as shown in figure 8. Dots on the plot can be largely divided into four sections according to amount of fold change. Most of genes are same trend of fold change in both analyses, but very few of them show reverse trend of fold change. Total 420 genes were found to be common in both analysis and depicted on the plots. Next, all three analyses were compared together and showed that 112 genes were found to be significant in these three analyses. This is shown in figure 9.

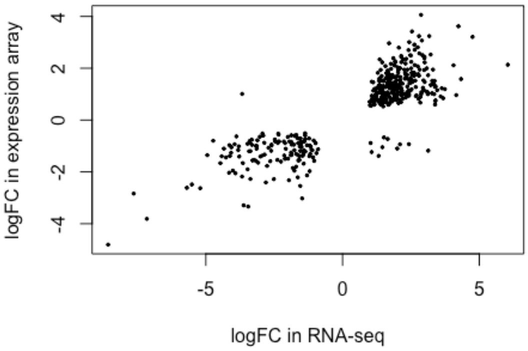


Figure 8. Scatter plot of log fold change of microarray and RNA sequencing results. Genes that differentially expressed in both analyses are plotted. Data is priorly filtered with FDR > 0.05.

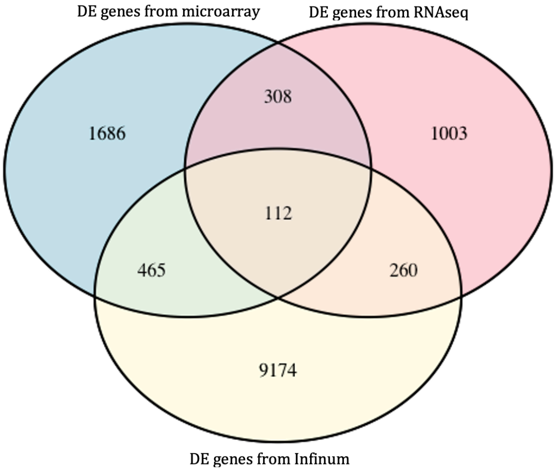


Figure 9. Venn diagram of overlapping differentially expressed (DE) and significant genes in three analyses.

4. Discussion

4.1 Data Integration

Differential expressed genes from microarray and RNA sequencing analyses were compared each other showing very similar trend of log fold change in both analyses. There are different aspects in both analyses, such as RNA extraction method, way of detecting RNA from array or flow cell, as well as different stages of tumor cells. Microarray data was earned from patients with invasive tumor tissue, showing biological pathways related to metastasis as shown in table 2. Differentially expressed genes in RNA sequencing are represented with biological pathways related to cell cycle as shown in table 4. This is because tumor samples were collected from patients with stage 3 lung cancer; Stage 3 cancer does not spread to other area yet. But stage 4 does (Patel, 2021). However, figure 8 shows very similar trend of log fold change despite these differences. There is a need for further investigation on genes that do not follow the trends. But they are expected to be genes involved in cell migration and metastasis.

Significant genes were compared also in three different analyses, microarray expression, RNA sequencing and methylation profiling. Through comparison, 112 genes are found out to be significant in these analyses. Top 4 significant genes are TGFBR3, LEPR, PHACTR1 and SOX17. All these genes have evidence of lung cancer related genes from the literature (Stelzer et al., 2016).

4.2 Methylation Data

Differential methylation analysis was performed for individual CpG sites for Infinium data. Among 10011 statistically significant (FDR < 0.05) CpG probes, there were three times downregulated methylated probes more than upregulated regions. This result means that methylation, leading to transcriptional silencing, decreased in tumor tissues. About 2998 statistically significant (FDR < 0.05) genes among 7455 annotated genes (only statistically significant) were related to TSS or 1st exon which are closely related to methylation. TSC22D4, PFKP, and DPYS were found as top 3 significant gene at promoter regions. We could find gene related to cancer, which is PKFP gene has role in metabolic reprogramming cancer cell such as lung or kidney. We could also know that methylation of this region is downregulated leading activation of this gene in tumor tissue with negative fold change value. DPYS gene has role in making protein which is important for pyrimidine metabolism, and it is highly expressed in lung or kidney. TSC22D4 is just a transcriptional regulator.

Gene set analysis was also performed to get insight about the biological pathway for specific CpG probes. Interestingly, lots of genes or pathway with methylation analysis are related to cell adhesion, and cell development or proliferation which are all closely related to cancer. Pathway related to cell proliferation or development pathway could be found with this analysis. Since tumor tissues mostly have problem with their continuous proliferation or development process, we could get more detailed insight with this analysis. However, since only individual CpG probes were analyzed for this report, analyzing regional level of methylation would provide much better insight if we combined regional CpG study with individual probe study.

ChIP-seq data was used to analyze and visualize the methylated regions. We could compare what binding regions are present only in untreated sample or treated sample. We investigated some genes that are present in top rated regions of methylation. For untreated cells, we could find CEACAM6, PECAM1, and DUSP6 genes in methylated regions as significant result. For treated cells, NUPR1, OLR1, and NKAIN4 genes in methylated regions were found as significant result. Even though there are some significant methylation peaks on promoter regions like TNFRSF21 gene containing region related to cell death, we couldn't identify our targeted region of study. In other word, there was few peaks with genes from overlapping results of three other omics study.

5. Conclusion

In conclusion, lung cancer related genes were studied using multiple omics methodology in this paper. Biologically and statistically meaningful lung cancer genes were detected from microarray, RNA sequencing and methylation profiling in common. These findings are linked to ChIP sequencing, but there were not enough significant methylation peaks consistent to the findings. If there is ChIP sequencing data with consistent study design, comparing tumor tissues and its adjacent normal tissues, more clear and consistent results and discussions points can be made further.

6. Contribution

Microarray expression and RNA sequencing analysis were done by Inkyun Park. Methylation profiling and ChIP sequencing were done Jihwan Lim. Corresponding parts in the paper were written by Inkyun Park and Jihwan Lim respectively.

References

All gene information is from www.genecards.org

Bjaanaes, M. M., Fleischer, T., Halvorsen, A. R., Daunay, A., Busato, F., Solberg, S., Jørgensen, L., Kure, E., Edvardsen, H., Børresen-Dale, A.-L., Brustugun, O. T., Tost, J., Kristensen, V., & Helland, Å. (2015). Genome-wide DNA methylation analyses in lung adenocarcinomas: Association with EGFR, Kras and TP53 mutation status, gene expression and prognosis. Molecular

Oncology, 10(2), 330–343. https://doi.org/10.1016/j.molonc.2015.10.021

Brenet, F., Moh, M., Funk, P., Feierstein, E., Viale, A. J., Socci, N. D., & Scandura, J. M. (2011). DNA methylation of the first exon is tightly linked to transcriptional silencing. PLoS ONE, 6(1). https://doi.org/10.1371/journal.pone.0014524

Eric S. Kim (2016). [Advances in Experimental Medicine and Biology] Lung Cancer and Personalized Medicine Volume 893 || Chemotherapy Resistance in Lung Cancer., 10.1007/978-3-319-24223-1(Chapter 10), 189–209. doi:10.1007/978-3-319-24223-1_10

Kumari, N., Singh, S., Haloi, D., Mishra, S. K., Krishnani, N., Nath, A., & Neyaz, Z. (2019). Epidermal growth factor receptor mutation frequency in squamous cell carcinoma and its diagnostic performance in cytological samples: A molecular and Immunohistochemical Study. World Journal of Oncology, 10(3), 142–150. https://doi.org/10.14740/wjon1204

Nierengarten, M. B. (2022). Annual report to the nation on the status of cancer. *Cancer*, *129*(1), 8–8. https://doi.org/10.1002/cncr.34586

Patel, J. D. (2021, November 11). *Lung cancer - non-small cell - stages*. Cancer.Net. Retrieved December 23, 2022, from https://www.cancer.net/cancer-types/lung-cancer-non-small-cell/stages

Seo, J.-S., Ju, Y. S., Lee, W.-C., Shin, J.-Y., Lee, J. K., Bleazard, T., Lee, J., Jung, Y. J., Kim, J.-O., Shin, J.-Y., Yu, S.-B., Kim, J., Lee, E.-R., Kang, C.-H., Park, I.-K., Rhee, H., Lee, S.-H., Kim, J.-I., Kang, J.-H., & Kim, Y. T. (2012). The transcriptional landscape and mutational profile of lung adenocarcinoma. *Genome Research*, *22*(11), 2109–2119. https://doi.org/10.1101/gr.145144.112

Stelzer G, Rosen R, Plaschkes I, Zimmerman S, Twik M, Fishilevich S, Iny Stein T, Nudel R, Lieder I, Mazor Y, Kaplan S, Dahary, D, Warshawsky D, Guan - Golan Y, Kohn A, Rappaport N, Safran M, and Lancet D.The GeneCards Suite: From Gene Data Mining to Disease Genome Sequence Analysis , Current Protocols in Bioinformatics(2016), 54:1.30.1 - 1.30.33.doi: 10.1002 / cpbi.5. [PDF]

Xu, L., Lu, C., Huang, Y., Zhou, J., Wang, X., Liu, C., Chen, J., & Le, H. (2018). SPINK1 promotes cell growth and metastasis of lung adenocarcinoma and acts as a novel prognostic biomarker. *BMB Reports*, *51*(12), 648–653. https://doi.org/10.5483/bmbrep.2018.51.12.205

Xu, L., Lu, C., Huang, Y., Zhou, J., Wang, X., Liu, C., Chen, J., & Le, H. (2018). SPINK1 promotes cell growth and metastasis of lung adenocarcinoma and acts as a novel prognostic biomarker. BMB Reports, 51(12), 648–653. https://doi.org/10.5483/bmbrep.2018.51.12.205

APPENDIX A

EXPRESSION ARRAY ANALYSIS

Expression array analysis R markdown codes are shown in appendix A

Data Analysis of Microarray Data of Tumour and Normal Lung Adenocarcinoma Tissues

Jihwan Lim & Inkyun Park

2022-12-23

Microarray data of lung cancer cells and adjacent normal cells from 6 patients are collected from GSE118370 database.

```
library(GEOquery)
library(affy)
library(arrayQualityMetrics)
library(limma)
library(biomaRt)
library(org.Hs.eg.db)
library(knitr)
```

1. Data Preparation

Using `getGEO`, we can download phenotype data of the microarray dataset.

```
# Get phenotype data from GSE118370
GSE118370 <- getGEO('GSE118370',GSEMatrix=TRUE)

## Found 1 file(s)

## GSE118370_series_matrix.txt.gz

lung_exp <- GSE118370[[1]]

# Check the downloaded data
head(lung_exp@phenoData@data[["title"]])

## [1] "Invasive lung adenocarcinoma tissue of patient No.1"
## [2] "paired normal lung tissue of of patient No.1"
## [3] "paired normal lung tissue of of patient No.2"
## [4] "Invasive lung adenocarcinoma tissue of patient No.2"
## [5] "Invasive lung adenocarcinoma tissue of patient No.3"
## [6] "paired normal lung tissue of of patient No.3"

# Read filenames from local disk
filenames <- list.files("./data/", pattern="*.CEL")
filenames <- paste0("./data/", filenames)

# Call AffyBatch obejct from CEL files and phenotype data
lung_affybatch <- ReadAffy(filenames = filenames, phenoData=pData(lung_exp)
)
```

```
kable(head(pData(lung_exp)[,-c(3,4,5,7,10:20,22:31)]))
```

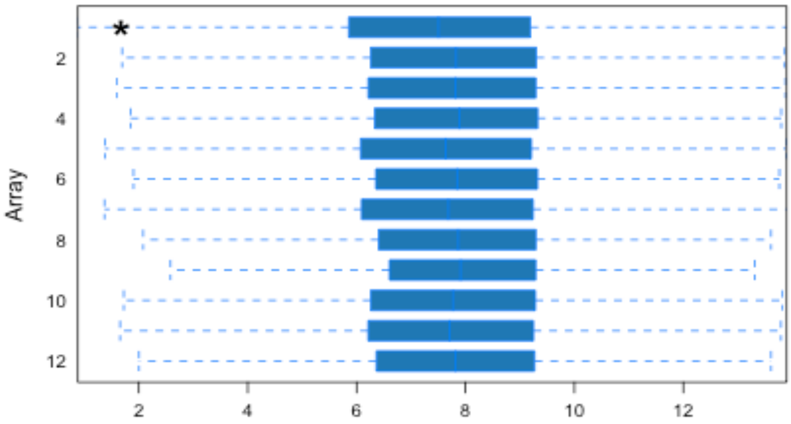
	title	geo_accession	type	source_name_ch1	organism_ch1	platform_id
GSM3325818	Invasive lung adenocarcinoma tissue of patient No.1	GSM3325818	RNA	Invasive lung adenocarcinoma tissue	Homo sapiens	GPL570
GSM3325819	paired normal lung tissue of of patient No.1	GSM3325819	RNA	normal lung tissue	Homo sapiens	GPL570
GSM3325820	paired normal lung tissue of of patient No.2	GSM3325820	RNA	normal lung tissue	Homo sapiens	GPL570
GSM3325821	Invasive lung adenocarcinoma tissue of patient No.2	GSM3325821	RNA	Invasive lung adenocarcinoma tissue	Homo sapiens	GPL570
GSM3325822	Invasive lung adenocarcinoma tissue of patient No.3	GSM3325822	RNA	Invasive lung adenocarcinoma tissue	Homo sapiens	GPL570
GSM3325823	paired normal lung tissue of of patient No.3	GSM3325823	RNA	normal lung tissue	Homo sapiens	GPL570

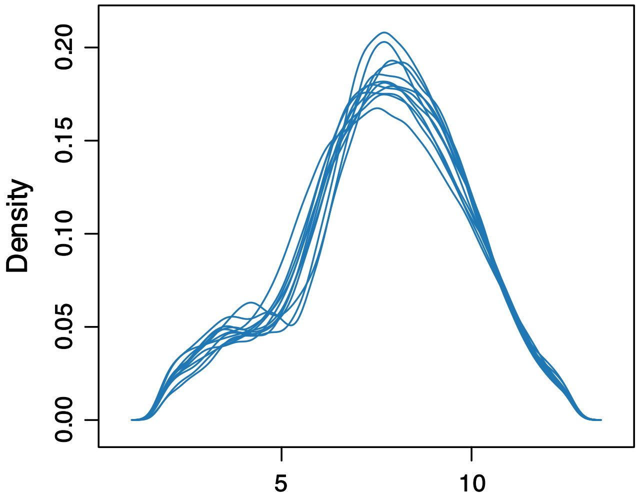
2. Preprocessing

2.1 Before preprocessing

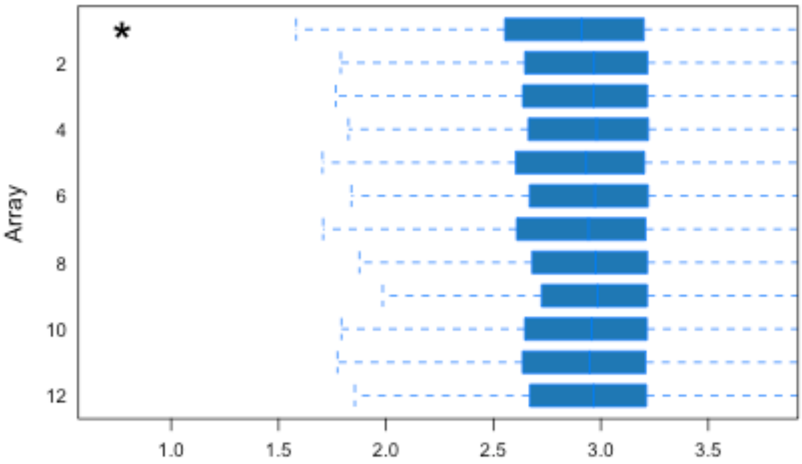
Do quality evaluation of raw data and log transformed data. Reports will be downloaded at local computer. We will show boxplotss and density plots to show the effect of preprocessing.

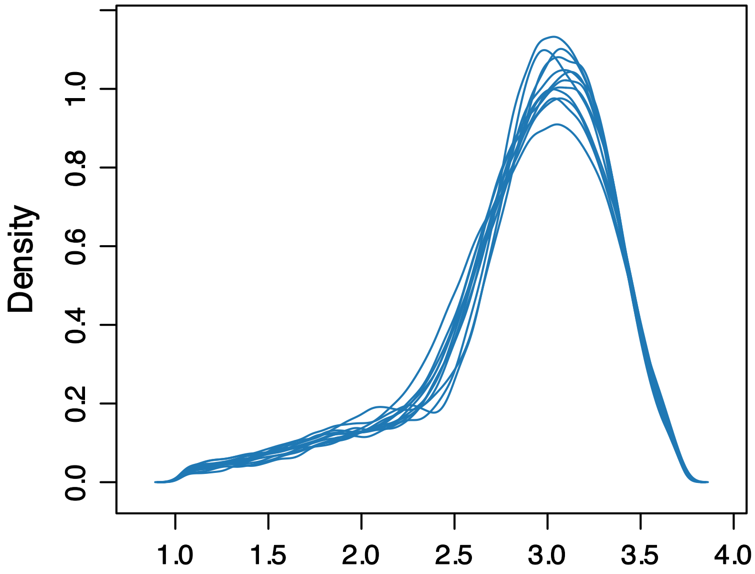
```
#assessing quality of raw dataset
arrayQualityMetrics(lung_exp,
                    outdir = "report_raw",
                    force = TRUE,
                    do.logtransform = FALSE)

## The report will be written into directory 'report_raw'.

## (loaded the KernSmooth namespace)
```





```
#assessing quality of log transformed dataset
arrayQualityMetrics(lung_exp,
                    outdir = "report_log_transformed",
                    force = TRUE,
                    do.logtransform = TRUE)

## The report will be written into directory 'report_log_transformed'.
```





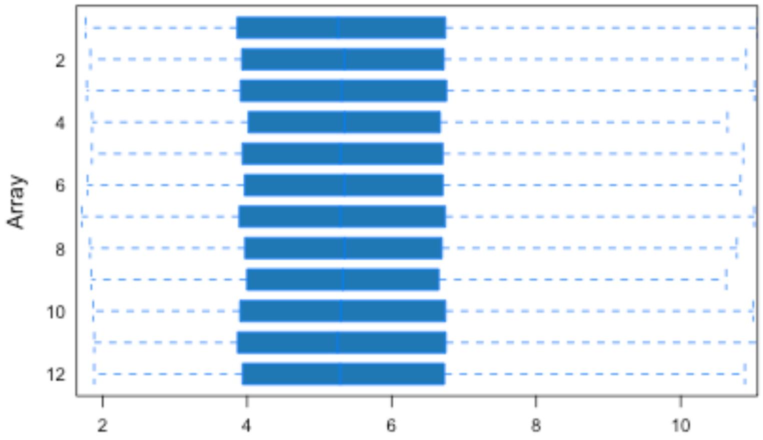
2.2 Preprocessing

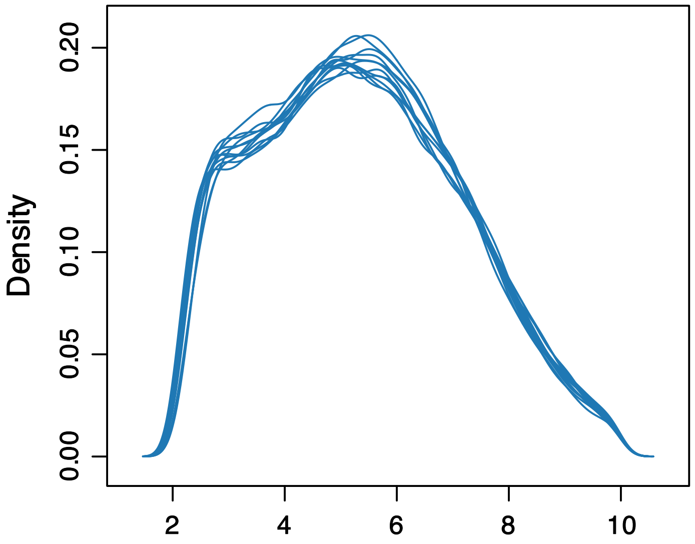
Do preprocessing using rma function, as well as background correction and quantile normalization.

```
lung_RMA <- affy::rma(lung_affybatch, background=TRUE, normalize=TRUE)

## Background correcting
## Normalizing
## Calculating Expression
```

Then, do quality evaluation on rma preprocessed data.

```
arrayQualityMetrics(expressionset = lung_RMA,
                    outdir = "report_rma", force = TRUE)
```





3. Differential Expression Analysis with RMA preprocessed data

Now, RMA preprocessed data will be used to analyze differential expression between two conditions.

First, we will look up data.

```
annot <- factor(substr(pData(lung_RMA)[,31], 0, nchar(pData(lung_RMA)[,31]
)-7))
```

3.1 Differential Expression by LIMMA

Using `limma`, differential expressed genes can be spotted.

```
design <- model.matrix(~ 0 + annot)
colnames(design) <- c("T", "N") #change colnames of design

# Fit genes on linear model
fit <- lmFit(lung_RMA, design)
cont.matrix <- makeContrasts(T-N, levels=design)

# Get estimated coefficients and standard error from fit
fit2 <- contrasts.fit(fit, cont.matrix)
# To estimate moderated variances
fit2 <- eBayes(fit2)
```

3.1.1 Differential Expression Analysis Results

```
# Extract DE genes
LIMMAout <- topTable(fit2,adjust="BH",number=nrow(exprs(lung_RMA)))
LIMMAout_sig <- LIMMAout[LIMMAout$adj.P.Val < 0.05, ]
LIMMAout_sig <- LIMMAout_sig[order(LIMMAout_sig$adj.P.Val),]
kable(head(LIMMAout_sig))

dim(LIMMAout_sig)
```

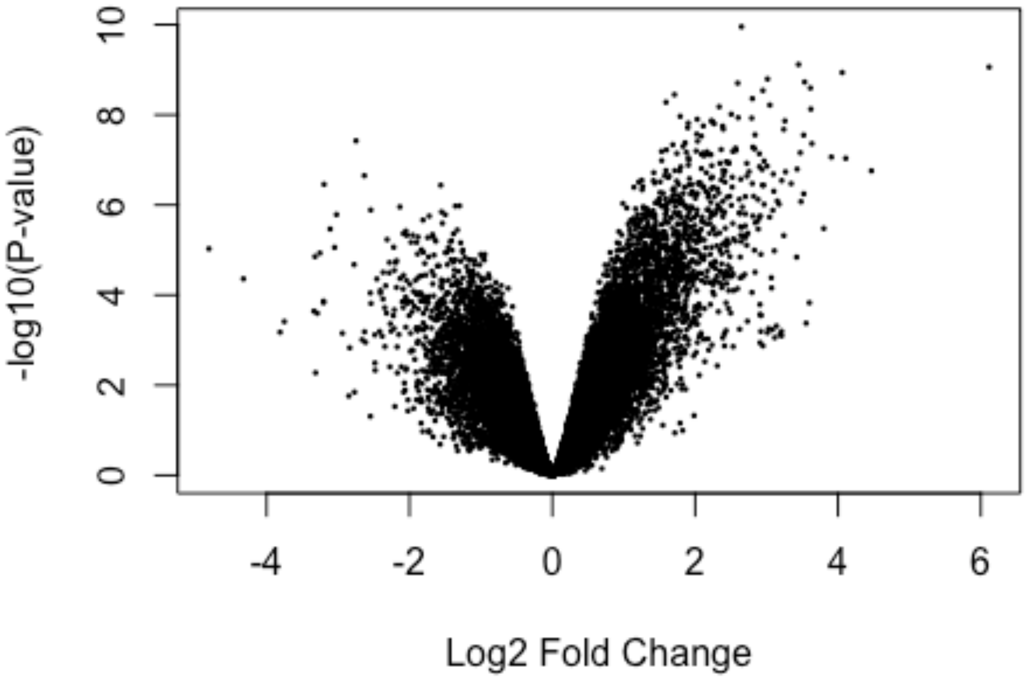
	logFC	AveExpr	t	P.Value	adj.P.Val	B
1557371_a_at	2.648723	6.013027	16.57061	0	6.10e-06	14.06004
1569608_x_at	3.446956	8.505440	14.35398	0	1.54e-05	12.49478
242009_at	6.115768	7.757118	14.21228	0	1.54e-05	12.38401
230469_at	4.058742	6.607292	13.91617	0	1.54e-05	12.14798
206702_at	3.012743	5.616195	13.57615	0	1.54e-05	11.86911
225660_at	3.532686	7.882041	13.41607	0	1.54e-05	11.73483

```
## [1] 4457    6
```

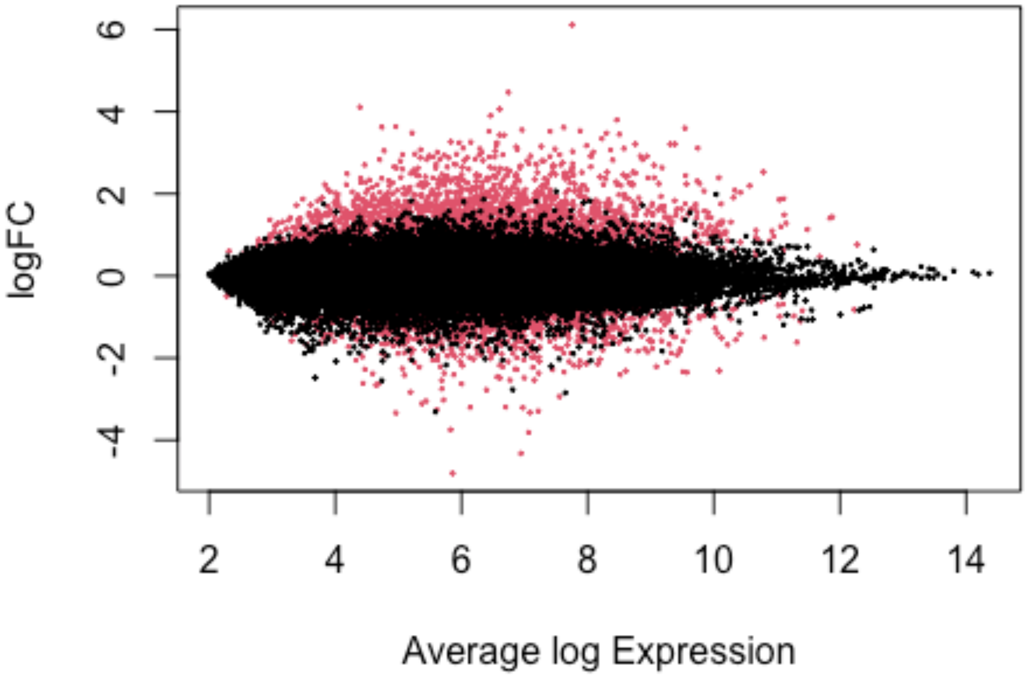
3.1.2 Plots

There is two ways to check data is making sense. In volcano plot, we can look for high or down regulated genes with statistically significant meaning. In MA plot, we are expecting horizontal distribution of points. Differentially expressed genes will be located top or botoom of the plot.

```
#volcano
volcanoplot(fit2)
```



```
#MA plot
plot(LIMMAout$AveExpr, LIMMAout$logFC,
     col=as.factor(LIMMAout$adj.P.Val < 0.05), pch=20, cex=0.25,
     xlab="Average log Expression", ylab="logFC")
```



4. Annotation

To annotate genes with high fold change, we need annotation file of the microarray platform. There, we can find annotations for probe IDs.

```
# Call annotation file
annotation_MA <- read.table("GPL570-55999.txt", sep="\t", fill=TRUE,quote=
"",head=TRUE)

# Extract probe IDs
probe_ids <- rownames(LIMMAout_sig)
LIMMAout_sig$entrez_id <- NA

# Annotate probe IDs into entrez gene ID
for (i in probe_ids) {
  probe_id <- paste(c(rbind("^", i, "$")), collapse='')
  entrez_id <- annotation_MA[annotation_MA$ID == i,]$ENTREZ_GENE_ID
  LIMMAout_sig[i, ]$entrez_id <- entrez_id
}

LIMMAout_sig$entrez_id <- gsub("\\ .*","",LIMMAout_sig$entrez_id)
kable(head(LIMMAout_sig))
```

	logFC	AveExpr	t	P.Value	adj.P.Val	B	entrez_id
1557371_a_at	2.648723	6.013027	16.57061	0	6.10e-06	14.06004	158376
1569608_x_at	3.446956	8.505440	14.35398	0	1.54e-05	12.49478	
242009_at	6.115768	7.757118	14.21228	0	1.54e-05	12.38401	6532
230469_at	4.058742	6.607292	13.91617	0	1.54e-05	12.14798	219790
206702_at	3.012743	5.616195	13.57615	0	1.54e-05	11.86911	7010
225660_at	3.532686	7.882041	13.41607	0	1.54e-05	11.73483	57556

Perform gene set analysis on differentially expressed genes. As `goana` only takes entrez gene id for the analysis, all genes IDs or symbols should be converted to entrez ID beforehand.

```
entrez_ids <- LIMMAout_sig$entrez_id

#subset for non duplicated and mapped genes
entrez_ids <- entrez_ids[!(duplicated(entrez_ids) | is.na(entrez_ids))]

goana_out <- goana(de=entrez_ids, species="Hs", trend=T)

goana_out <- goana_out[order(goana_out$P.DE, decreasing=FALSE),]
goana_out$FDR.DE <- p.adjust(goana_out$P.DE, method="BH")
topGOcpg <- topGO(goana_out, ontology="BP", number=Inf)
kable(head(topGOcpg, 10))
```

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0007155	cell adhesion	BP	1510	360	0	0
GO:0009653	anatomical structure morphogenesis	BP	2746	546	0	0
GO:0016477	cell migration	BP	1556	354	0	0
GO:0048856	anatomical structure development	BP	5785	932	0	0
GO:0007275	multicellular organism development	BP	4804	802	0	0
GO:0040011	locomotion	BP	1925	404	0	0
GO:0032502	developmental process	BP	6355	996	0	0
GO:0032879	regulation of localization	BP	2808	530	0	0
GO:0048731	system development	BP	4345	737	0	0
GO:0048870	cell motility	BP	1750	373	0	0

5. Save Results

```
write.csv(LIMMAout_sig, "DEgenes_microarray.csv")
write.csv(topGOcpg, "GSA_microarray.csv")
```

APPENDIX B

RNA SEQUENCING ANALYSIS

RNA sequencing analysis R markdown codes are shown in appendix B

Data Analysis of RNA-seq Data of Tumour and Normal Lung Adenocarcinoma Tissues

Jihwan Lim & Inkyun Park

2022-12-23

RNA sequencing data of lung cancer cells and adjacent normal cells from 3 patients with stage 3 lung cancer are collected from GSE40419 database. Quality control and mapping to reference files using KALLISTO of raw FASTQ files are already done in the HPC.

```
# Load necessary packages
library(biomaRt)
library(tximport)
library(edgeR)
library(limma)
library(org.Hs.eg.db)
library(DESeq2)
library(knitr)
```

1. Data Preparation

Get gene ID from reference files to annotate gene ID on the sample data from KALLISTO

```
# Get annotation data
human_mart <- useEnsembl("ensembl","hsapiens_gene_ensembl")

# What are the available attributes
atr <- listAttributes(human_mart)

data <- getBM(attributes = c('ensembl_gene_id', 'ensembl_transcript_id',
                           'external_gene_name'),
             mart = human_mart)

tx2geneGtf <- dplyr::select(data, ensembl_transcript_id, ensembl_gene_id)
tx2geneGtf <- dplyr::rename(tx2geneGtf, TXNAME = ensembl_transcript_id)
tx2geneGtf <- dplyr::rename(tx2geneGtf, GENEID = ensembl_gene_id)

kable(head(tx2geneGtf, 3))
```

TXNAME	GENEID
ENST00000387314	ENSG00000210049
ENST00000389680	ENSG00000211459
ENST00000387342	ENSG00000210077

1.1 Load data

Load in sample data which is already mapped to genome.

```
## Get file locations
files <- list.files("kallisto_quant/")
files <- files[grep("abundance.tsv",files)]
samples <- unlist(strsplit(files,"_"))[c(1:length(files))*2-1]
files <- paste(rep("kallisto_quant/",length(files)),files,sep="")
names(files) <- samples

## Load RNAseq data
txi <- tximport(files, type = "kallisto", tx2gene = tx2geneGtf)

## Note: importing `abundance.h5` is typically faster than `abundance.tsv`

## reading in files with read_tsv

## 1 2 3 4 5 6
## summarizing abundance
## summarizing counts
## summarizing length

## Have a look at the data
kable(head(txi$counts))
```

	ERR164515	ERR164522	ERR164526	ERR164600	ERR164607	ERR164611
ENSG00000000003	500.8357	634.8906	899.17647	4743.917	6287.0130	2029.8513
ENSG00000000005	0.0000	1.0000	26.00001	1.000	6.0000	0.0000
ENSG00000000419	585.9435	719.6850	771.92814	1954.772	3419.8227	1888.0177
ENSG00000000457	493.9299	733.8527	647.83964	1931.198	1827.7032	1016.4454
ENSG00000000460	100.8412	139.0246	125.79970	980.007	1193.2568	627.2276
ENSG00000000938	1137.0002	1308.0006	1191.99880	2559.999	957.0004	1565.9961

```
dim(txi$counts)

## [1] 62703     6
```

2. Statistical analysis

First, we check duplicated row of the data and make annotation for design.

```
## Check for duplicate rows
sum(duplicated(rownames(txi$counts)))

## [1] 0

dim(txi$abundance)

## [1] 62703     6
## Make annotation for design later on
tissue <- factor(c("Tumor","Tumor","Tumor","Normal","Normal","Normal"))
```

2.1 EdgeR

edgeR package is differential expression analysis with statistical models for RNA-seq data.

2.1.1 Preprocessing

As normalization factors are already calculated with tximport, we can next do filtering by cpm (counts-per-million). In filtering, we want to choose genes with certain expression at different 3 samples.

```
## Make tpm values compatible with edgeR
cts <- txi$counts
normMat <- txi$length

# Obtaining per-observation scaling factors for length, adjusted to avoid c
hanging the magnitude of the counts.
normMat <- normMat/exp(rowMeans(log(normMat)))
normCts <- cts/normMat

# Computing effective library sizes from scaled counts, to account for comp
osition biases between samples.
eff.lib <- calcNormFactors(normCts) * colSums(normCts)

# Combining effective library sizes with the length factors, and calculatin
g offsets for a log-link GLM.
normMat <- sweep(normMat, 2, eff.lib, "*")
normMat <- log(normMat)

kable(eff.lib)
```

	x
ERR164515	22664941
ERR164522	27694706
ERR164526	26642912
ERR164600	70649076
ERR164607	51993236
ERR164611	49290962

```
# Creating a DGEList object for use in edgeR.
y <- DGEList(cts)
y <- scaleOffset(y, normMat)

# Estimate cpm threshold value and filter genes with low counts by cpm.
cutoff <- 3/(mean(y$samples$lib.size)/1000000)
keep <- rowSums(cpm(y)>cutoff) >= 3
y <- y[keep, ,keep.lib.sizes=FALSE]
summary(keep)
```

```
##    Mode   FALSE    TRUE
## logical   30749   31954
```

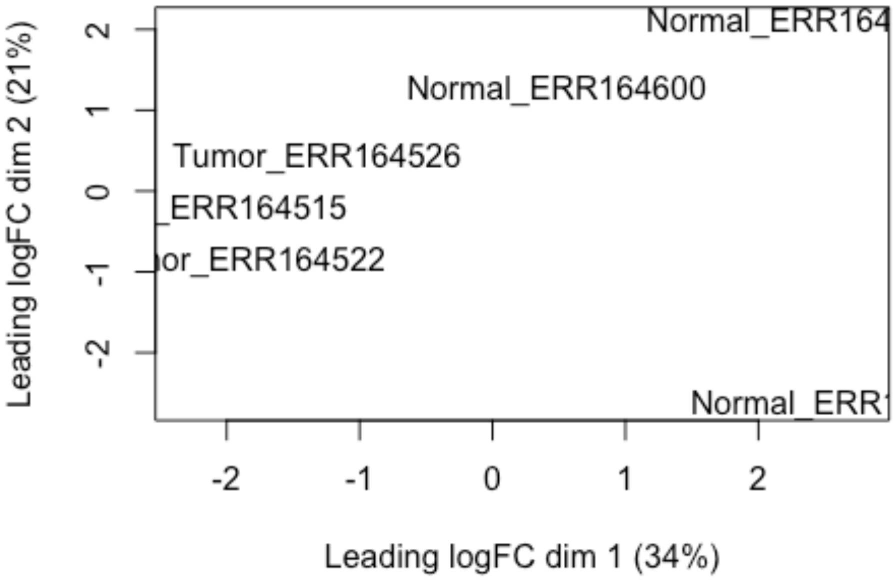
Define design matrix based on our experimental design: find differentially expressed genes between tumor and adjacent normal tissues.

```
design <- model.matrix(~tissue)
rownames(design) <- colnames(y)
kable(design)
```

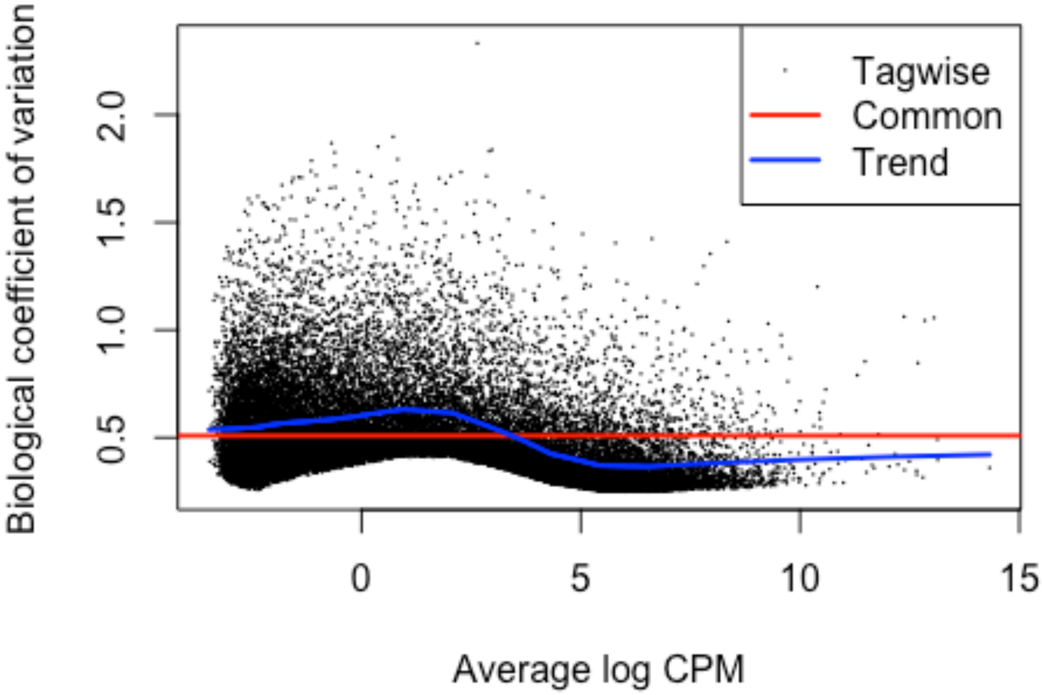
	(Intercept)	tissueTumor
ERR164515	1	1
ERR164522	1	1
ERR164526	1	1
ERR164600	1	0
ERR164607	1	0
ERR164611	1	0

Plot Multi-Dimensional Scaling plot (MDS) and Biological Coefficient of Variation (BCV).

```
label <- paste0(tissue, "_", colnames(y))
limma::plotMDS(y, labels = label)
```



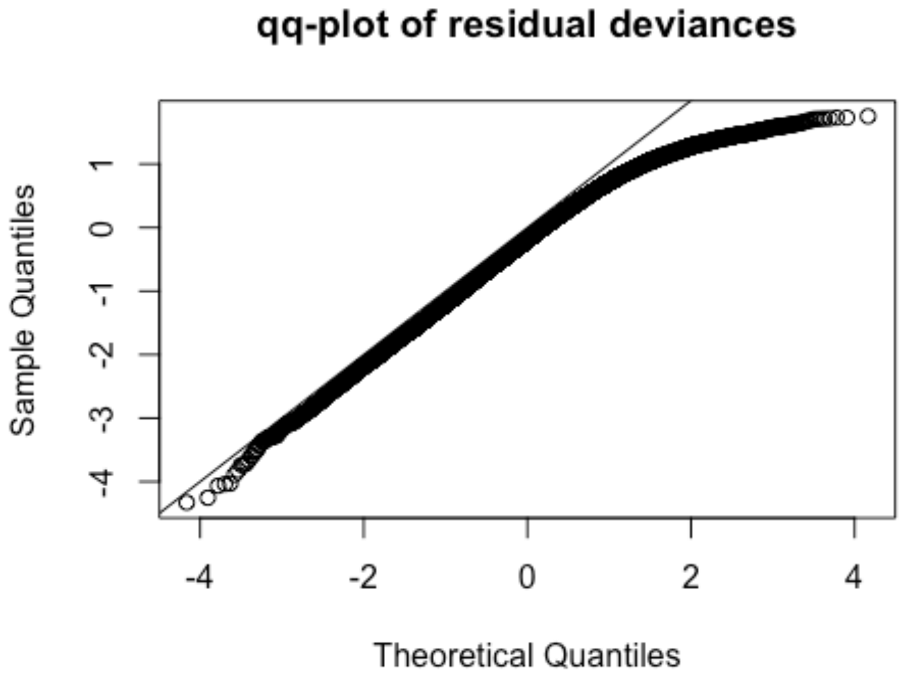
```
y <- estimateDisp(y, design, robust=TRUE)
plotBCV(y)
```



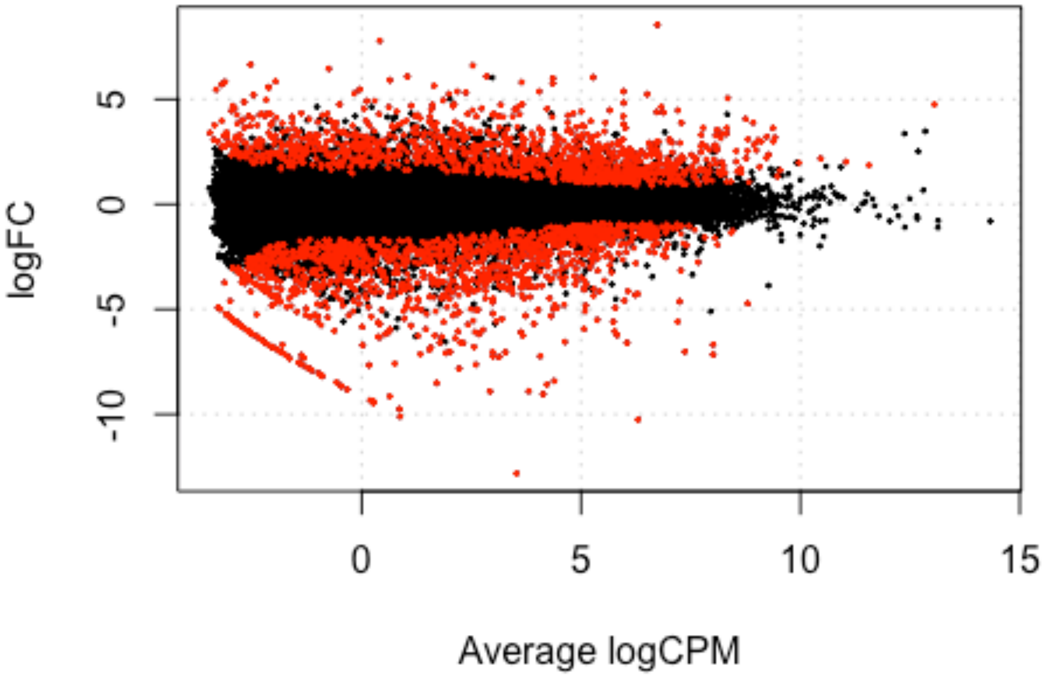
2.1.2 Differential Expression Analysis using edgeR

Using edgeR packages, now we can find differentially expressed genes.

```
# Perform likelihood ratio tests:
fit <- glmFit(y, design)

# See goodness of the fit.
gof(fit, plot=TRUE)
```



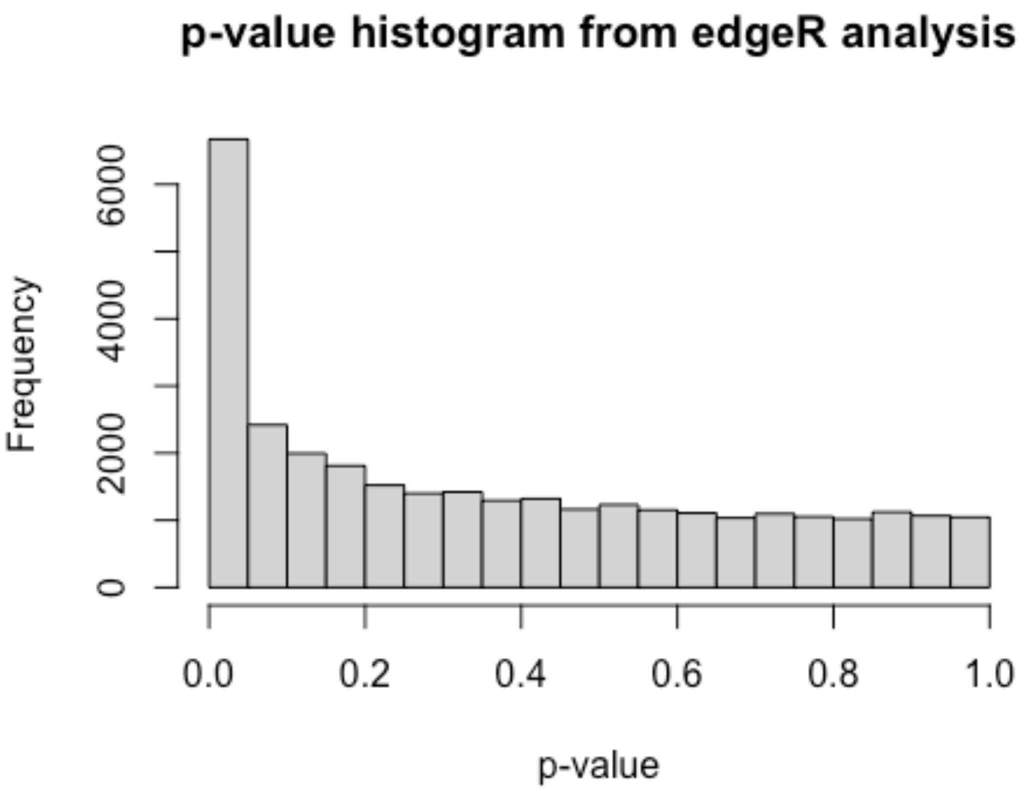
```
lrt <- glmLRT(fit)
dt <- decideTestsDGE(lrt)
plotSmear(lrt, de.tags=rownames(y)[as.logical(dt)])
```



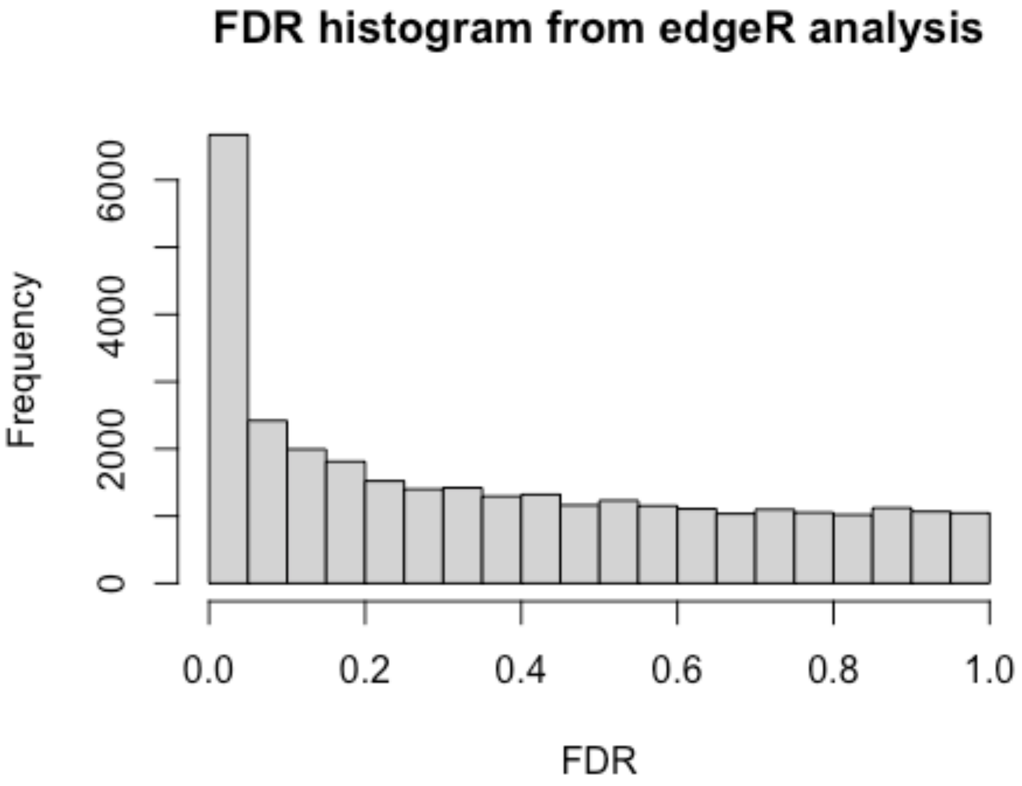
```
# Summary of up or down regulated genes.
summary(dt)

##        tissueTumor
## Down          1296
## NotSig       29529
## Up            1129
```

We found out that 2425 genes are differentially expressed.

```
res_edger <- topTags(lrt, n="Inf", sort.by="logFC")

# p-value histogram
hist(res_edger$table$PValue,
     main="p-value histogram from edgeR analysis",
     xlab = "p-value")
```



```
# FDR histogram
hist(res_edger$table$PValue,
     main="FDR histogram from edgeR analysis",
     xlab = "FDR")
```



```r
# Select significantly expressed genes
res_edger_sig <- res_edger[res_edger$table$FDR < 0.05,]$table
res_edger_sig <- res_edger_sig[order(res_edger_sig$FDR), ]
kable(head(res_edger_sig))
```

	logFC	logCPM	LR	PValue	FDR
ENSG00000153234	4.640327	6.833942	113.41945	0	0
ENSG00000131747	-4.262364	6.299861	101.50038	0	0
ENSG00000118785	-7.157363	8.013378	94.13553	0	0
ENSG00000179388	3.891028	5.553918	91.26761	0	0
ENSG00000262406	-7.638597	2.599256	78.16732	0	0
ENSG00000007908	5.754552	4.354936	76.99101	0	0

```r
dim(res_edger_sig)
```

```
## [1] 2425    5
```

2.1.3 Gene Set Analysis

Perform gene set analysis on differentially expressed genes.

```
# Change ensembl gene ID into entrez ID to be compatible with goana functio
n.
entrez_ids <- mapIds(org.Hs.eg.db,
                     keys=rownames(res_edger_sig),
                     column="ENTREZID",
                     keytype="ENSEMBL")
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
# Add ensemble gene ID on results from edgeR
#df1$vector1<-vector1[match(df1$ID,names(vector1))]
res_edger_sig$entrezIDs <- entrez_ids[match(rownames(res_edger_sig), names
(entrez_ids))]


#subset for non duplicated and mapped genes
entrez_ids <- entrez_ids[!(duplicated(entrez_ids) | is.na(entrez_ids))]

goana_out <- goana(de=entrez_ids, species="Hs", trend=T)

goana_out <- goana_out[order(goana_out$P.DE, decreasing=FALSE),]
goana_out$FDR.DE <- p.adjust(goana_out$P.DE, method="BH")
topGOcpg <- topGO(goana_out, ontology="BP", number=Inf)
kable(head(topGOcpg, 10))
```

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0000278	mitotic cell cycle	BP	898	170	0	0
GO:1903047	mitotic cell cycle process	BP	744	149	0	0
GO:0007049	cell cycle	BP	1760	265	0	0
GO:0022402	cell cycle process	BP	1205	195	0	0
GO:0048856	anatomical structure development	BP	5785	631	0	0
GO:0032502	developmental process	BP	6355	676	0	0
GO:0007275	multicellular organism development	BP	4804	538	0	0
GO:0050896	response to stimulus	BP	9030	887	0	0
GO:0051301	cell division	BP	622	116	0	0
GO:0000280	nuclear division	BP	446	92	0	0

```
dim(topGOcpg)
```

```
## [1] 15947     6
```

3. Save Results

```
write.csv(res_edger_sig, "DEgenes_edger_RNAseq.csv")
write.csv(topGOcpg, "GSA_edger_RNAseq.csv")
```

APPENDIX C

INFINIUM ARRAY ANALYSIS

Infinium array analysis R markdown codes are shown in appendix C

Methylation Array Analysis

Jihwan Lim & Inkyun Park

2022-12-23

1. Methylation Array Analysis

A methylation array data set was analysed to assess methylation changes in tumor tissue versus normal lung tissue. The data was collected from lung cancer patients and normal people in Norway. DNA from patients and people were analysed with a Illumina Infinium HumanMethylation450 BeadChip.(GSE40419)

1.1 Load in necessary packages

```
library(tidyverse)
library(lumi)
library(wateRmelon)
library(ChAMPdata)
library(IlluminaHumanMethylation450kanno.ilmn12.hg19)
library(org.Hs.eg.db)
library(knitr)
```

1.2 Load annotation data

```
infinium_annotation <- t(read.table("./GSE66836_series_matrix.txt",sep="\t
",fill=T))
infinium_annotation <- data.frame(ID = rownames(infinium_annotation), infin
ium_annotation)
infinium_annotation[1,1] <- "ID"
colnames(infinium_annotation) <- infinium_annotation[1,]
infinium_annotation <- infinium_annotation[-1,]
rownames(infinium_annotation) <- 1:nrow(infinium_annotation)
kable(head(infinium_annotation[,c(2,3,9,10,12,13,14)]))
```

title	geo_accession	source_name_ch1	organism_ch1	characteristics_ch1	characteristics_ch2	characteristics_ch3
Sample1_Tumor	GSM1632880	lung adenocarcinoma	Homo sapiens	tissue: Tumor	Stage: 1	p53 status: NA
Sample2_Normal	GSM1632881	normal lung	Homo sapiens	tissue: Normal	Stage: NA	p53 status: NA
Sample3_Tumor	GSM1632882	lung adenocarcinoma	Homo sapiens	tissue: Tumor	Stage: 4	p53 status: NA
Sample4_Tumor	GSM1632883	lung adenocarcinoma	Homo sapiens	tissue: Tumor	Stage: 1	p53 status: Mutated
Sample5_Normal	GSM1632884	normal lung	Homo sapiens	tissue: Normal	Stage: NA	p53 status: NA
Sample6_Tumor	GSM1632885	lung adenocarcinoma	Homo sapiens	tissue: Tumor	Stage: 1	p53 status: WildType

1.3 select data

1.3.1 Get specific data that we target

```
  # Pick necessary colums for choosing
annot <- infinium_annotation[c("!Sample_title","!Sample_geo_accession", "!
Sample_characteristics_ch1", "!Sample_characteristics_ch2", "!Sample_chara
cteristics_ch3", "!Sample_characteristics_ch4", "!Sample_characteristics_c
h5", "!Sample_source_name_ch1", "!Sample_description1", "!Sample_descripti
on2")]

# Change the name of elements to do remove unnecessary data
annot$`!Sample_characteristics_ch1` <- gsub('tissue: ','', annot$`!Sample_
characteristics_ch1`)
annot$`!Sample_description2` <- gsub('Sentrix_Position: ','', annot$`!Samp
le_description2`)
annot$`!Sample_description1` <- gsub('Sentrix_ID: ','', annot$`!Sample_des
cription1`)
kable(head(annot, 10))
```

title	geo_accession	characteristics_ch1	characteristics_ch2	characteristics_ch3	characteristics_ch4	characteristics_ch5	source_name_ch1	description1	description2
Sample1_Tumor	GSM1632880	Tumor	Stage: 1	p53 status: NA	egfr status: NA	kras status: WildType	lung adenocarcinoma	5775278068	R01C01
Sample2_Normal	GSM1632881	Normal	Stage: NA	p53 status: NA	egfr status: NA	kras status: NA	normal lung	5808922089	R03C02
Sample3_Tumor	GSM1632882	Tumor	Stage: 4	p53 status: NA	egfr status: WildType	kras status: WildType	lung adenocarcinoma	5808922089	R01C01
Sample4_Tumor	GSM1632883	Tumor	Stage: 1	p53 status: Mutated	egfr status: WildType	kras status: WildType	lung adenocarcinoma	5775278068	R06C01
Sample5_Normal	GSM1632884	Normal	Stage: NA	p53 status: NA	egfr status: NA	kras status: NA	normal lung	5775278017	R06C02
Sample6_Tumor	GSM1632885	Tumor	Stage: 1	p53 status: WildType	egfr status: Mutated	kras status: WildType	lung adenocarcinoma	5775278017	R05C02
Sample7_Tumor	GSM1632886	Tumor	Stage: 5	p53 status: WildType	egfr status: WildType	kras status: Mutated	lung adenocarcinoma	5808922089	R06C02
Sample8_Tumor	GSM1632887	Tumor	Stage: 1	p53 status: Mutated	egfr status: WildType	kras status: Mutated	lung adenocarcinoma	5808922086	R04C02
Sample9_Tumor	GSM1632888	Tumor	Stage: 1	p53 status: WildType	egfr status: WildType	kras status: Mutated	lung adenocarcinoma	5775278004	R03C01
Sample10_Tumor	GSM1632889	Tumor	Stage: 3	p53 status: WildType	egfr status: WildType	kras status: NA	lung adenocarcinoma	5775278017	R02C02

```
  # Pick necessary columns for choosing
annot_sel <- infinium_annotation[c("!Sample_title","!Sample_geo_accession"
, "!Sample_characteristics_ch1", "!Sample_characteristics_ch2", "!Sample_c
haracteristics_ch3", "!Sample_characteristics_ch4", "!Sample_characteristi
cs_ch5", "!Sample_source_name_ch1", "!Sample_description1", "!Sample_descr
iption2")]

# Change the name of elements to do remove unnecessary data
annot_sel$`!Sample_characteristics_ch1` <- gsub('tissue: ','', annot_sel$`
!Sample_characteristics_ch1`)
annot_sel$`!Sample_description2` <- gsub('Sentrix_Position: ','', annot_se
l$`!Sample_description2`)
annot_sel$`!Sample_description1` <- gsub('Sentrix_ID: ','', annot_sel$`!Sa
mple_description1`)


colnames(annot_sel) <- c("Sample_title","Geo_accession","Tissue", "Stage",
 "p53_status", "EGFR_status", "KRAS_status", "character","Sentrix_ID", "Sen
trix_Position")

annot_sel$Stage <- gsub("Stage: ", "", annot_sel$Stage)
annot_sel$p53_status <- gsub("p53 status: ", "", annot_sel$p53_status)
annot_sel$EGFR_status <- gsub("egfr status: ", "", annot_sel$EGFR_status)
annot_sel$KRAS_status <- gsub("kras status: ", "", annot_sel$KRAS_status)

kable(head(annot_sel, 10))
```

Sample_title	Geo_accession	Tissue	Stage	p53_status	EGFR_status	KRAS_status	character	Sentrix_ID	Sentrix_Position
Sample1_Tumor	GSM1632880	Tumor	1	NA	NA	WildType	lung adenocarcinoma	5775278068	R01C01
Sample2_Normal	GSM1632881	Normal	NA	NA	NA	NA	normal lung	5808922089	R03C02
Sample3_Tumor	GSM1632882	Tumor	4	NA	WildType	WildType	lung adenocarcinoma	5808922089	R01C01
Sample4_Tumor	GSM1632883	Tumor	1	Mutated	WildType	WildType	lung adenocarcinoma	5775278068	R06C01
Sample5_Normal	GSM1632884	Normal	NA	NA	NA	NA	normal lung	5775278017	R06C02
Sample6_Tumor	GSM1632885	Tumor	1	WildType	Mutated	WildType	lung adenocarcinoma	5775278017	R05C02
Sample7_Tumor	GSM1632886	Tumor	5	WildType	WildType	Mutated	lung adenocarcinoma	5808922089	R06C02
Sample8_Tumor	GSM1632887	Tumor	1	Mutated	WildType	Mutated	lung adenocarcinoma	5808922086	R04C02
Sample9_Tumor	GSM1632888	Tumor	1	WildType	WildType	Mutated	lung adenocarcinoma	5775278004	R03C01

Sample_title	Geo_accession	Tissue	Stage	p53_status	EGFR_status	KRAS_status	character	Sentrix_ID	Sentrix_Position
Sample10_Tumor	GSM1632889	Tumor	3	WildType	WildType	NA	lung adenocarcinoma	5775278017	R02C02

1.3.2 filtration for annotation

```
des1 <- annot$`!Sample_geo_accession`
des2 <- annot$`!Sample_description1`
des3 <- annot$`!Sample_description2`

# how sample name looks like
des_final <- paste(des1,des2,des3,sep = "_")

annot$marker <- des_final
annot_sel$marker <- des_final
```

1.3.3 Get annoation of stage 3 tumor samples with mutated EGFR

```
Tumor <- annot_sel[grep("Mutated", annot_sel$EGFR_status),]
Tumor <- Tumor[grep("WildType", Tumor$p53_status),]
Tumor <- Tumor[grep("WildType", Tumor$KRAS_status),]
Tumor3 <- Tumor[grep(3 ,Tumor$Stage),]
kable(Tumor3)
```

	Sample_title	Geo_accession	Tissue	Stage	p53_status	EGFR_status	KRAS_status	character	Sentrix_ID	Sentrix_Position	marker
23	Sample23_Tumor	GSM1632902	Tumor	3	WildType	Mutated	WildType	lung adenocarcinoma	5775278068	R03C01	GSM1632902_5775278068_R03C01
48	Sample48_Tumor	GSM1632927	Tumor	3	WildType	Mutated	WildType	lung adenocarcinoma	5775278004	R04C02	GSM1632927_5775278004_R04C02
137	Sample137_Tumor	GSM1633016	Tumor	3	WildType	Mutated	WildType	lung adenocarcinoma	5775278003	R02C02	GSM1633016_5775278003_R02C02

1.3.4 Get annotation of normal samples

```
Normal <- annot_sel[grep("Normal", annot_sel$Tissue),]

# Find normal samples that have same Sentrix position with tumor 3
Normal <- Normal[Normal$Sentrix_Position %in% Tumor3$Sentrix_Position,]
kable(Normal)
```

	Sample_title	Geo_accession	Tissue	Stage	p53_status	EGFR_status	KRAS_status	character	Sentrix_ID	Sentrix_Position	marker
96	Sample96_Normal	GSM1632975	Normal	NA	NA	NA	NA	normal lung	5775446011	R02C02	GSM1632975_5775446011_R02C02

	Sample_title	Geo_accession	Tissue	Stage	p53_status	EGFR_status	KRAS_status	character	Sentrix_ID	Sentrix_Position	marker
108	Sample 108_Normal	GSM1632987	Normal	NA	NA	NA	NA	normal lung	5775446011	R04C02	GSM1632987_5775446011_R04C02
112	Sample 112_Normal	GSM1632991	Normal	NA	NA	NA	NA	normal lung	5775278030	R02C02	GSM1632991_5775278030_R02C02
119	Sample 119_Normal	GSM1632998	Normal	NA	NA	NA	NA	normal lung	5775278034	R03C01	GSM1632998_5775278034_R03C01
178	Sample 178_Normal	GSM1633057	Normal	NA	NA	NA	NA	normal lung	5775278003	R04C02	GSM1633057_5775278003_R04C02

we have 19 normal samples and 164 tumor samples. Since there are lots of factors to be considered like stage, mutation of EGFR, KRAS, or TP53 genes, etc, we decided to use stage 3 with mutated EGFR samples (WT with KRAS and TP53 genes) for tumor samples. This can be further related to RNA-seq data which has stage 3 tumor tissue. Then we randomly chose 3 normal samples which have same sentrix position among 5 of samples.

1.4 Load the Infinium data

```
  # Load EPIC data
infdata <- readEPIC("./data/")
# Since there are 183 samples in raw file, we just made a new folder that on
ly contains that we only chose before
```

1.5 Take new annotation table that only contain necessary data

```
  # we already make marker column
annot <- annot %>% filter(annot$marker %in% sampleNames(infdata))
```

1.6 Have a look at the data and annotation

```
print(infdata)

##
## Object Information:
## MethyLumiSet (storageMode: lockedEnvironment)
## assayData: 485577 features, 6 samples
##   element names: betas, methylated, methylated.N, NBeads, pvals, unmethy
lated, unmethylated.N
## protocolData: none
## phenoData
##   sampleNames: GSM1632902_5775278068_R03C01
##     GSM1632927_5775278004_R04C02 ... GSM1633016_5775278003_R02C02 (6
##     total)
##   varLabels: barcode
##   varMetadata: labelDescription
## featureData
##   featureNames: cg00000029 cg00000108 ... rs9839873 (485577 total)
##   fvarLabels: Probe_ID DESIGN COLOR_CHANNEL
##   fvarMetadata: labelDescription
## experimentData: use 'experimentData(object)'
## Annotation: IlluminaHumanMethylation450k
## Major Operation History:
```

```
##             submitted            finished
## 1 2022-12-23 21:12:43 2022-12-23 21:12:54
## 2 2022-12-23 21:12:43 2022-12-23 21:12:54
## 3 2022-12-23 21:12:58 2022-12-23 21:12:59
##                                                               command
## 1 NChannelSetToMethyLumiSet2(NChannelSet = dats, parallel = parallel,
## 2                                                    n = n, oob = oob)
## 3                                           Subset of 485577 features.
```

```
print(dim(infdata))
```

```
## Features  Samples
##   485577        6
```

```
kable(annot)
```

title	geo_accession	characteristics_ch1	characteristics_ch2	characteristics_ch3	characteristics_ch4	characteristics_ch5	source_name_ch1	description1	description2	marker
Sample23_Tumor	GSM1632902	Tumor	Stage: 3	p53 status: WildType	egfr status: Mutated	kras status: WildType	lung adenocarcinoma	5775278068	R03C01	GSM1632902_5775278068_R03C01
Sample48_Tumor	GSM1632927	Tumor	Stage: 3	p53 status: WildType	egfr status: Mutated	kras status: WildType	lung adenocarcinoma	5775278004	R04C02	GSM1632927_5775278004_R04C02
Sample96_Normal	GSM1632975	Normal	Stage: NA	p53 status: NA	egfr status: NA	kras status: NA	normal lung	5775446011	R02C02	GSM1632975_5775446011_R02C02
Sample108_Normal	GSM1632987	Normal	Stage: NA	p53 status: NA	egfr status: NA	kras status: NA	normal lung	5775446011	R04C02	GSM1632987_5775446011_R04C02
Sample119_Normal	GSM1632998	Normal	Stage: NA	p53 status: NA	egfr status: NA	kras status: NA	normal lung	5775278034	R03C01	GSM1632998_5775278034_R03C01
Sample137_Tumor	GSM1633016	Tumor	Stage: 3	p53 status: WildType	egfr status: Mutated	kras status: WildType	lung adenocarcinoma	5775278003	R02C02	GSM1633016_5775278003_R02C02

```
kable(sum(is.na(exprs(infdata))))
```

x
48986

```
  # betas function retrieve beta value (=methylation percentage)
kable(head(betas(infdata)))
```

	GSM1632902_5775278068_R03C01	GSM1632927_5775278004_R04C02	GSM1632975_5775446011_R02C02	GSM1632987_5775446011_R04C02	GSM1632998_5775278034_R03C01	GSM1633016_5775278003_R02C02
cg00000029	0.5834395	0.3879270	0.1474793	0.1805667	0.2389768	0.3032751
cg00000108	0.7193320	0.6988593	0.8464406	0.8155882	0.8060476	0.7933194

	GSM1632902_5775278068_R03C01	GSM1632927_5775278004_R04C02	GSM1632975_5775446011_R02C02	GSM1632987_5775446011_R04C02	GSM1632998_5775278034_R03C01	GSM1633016_5775278003_R02C02
cg00000109	0.6011765	0.5210166	0.6682365	0.7194737	0.6942356	0.6163934
cg00000165	NA	0.4775758	0.2771689	0.2511721	0.2824829	0.2448394
cg00000236	0.4795918	0.5409836	0.7345242	0.6801454	0.6786818	0.6773387
cg00000289	0.2607973	0.3133245	0.4355576	0.3461876	0.3754845	0.3498205

```
 # exprs function retrieve M-value
kable(head(exprs(infdata)))
```

	GSM1632902_5775278068_R03C01	GSM1632927_5775278004_R04C02	GSM1632975_5775446011_R02C02	GSM1632987_5775446011_R04C02	GSM1632998_5775278034_R03C01	GSM1633016_5775278003_R02C02
cg00000029	0.4860570	-0.6579185	-2.5312223	-2.182095	-1.671070	-1.1999622
cg00000108	1.3577935	1.2145644	2.4626122	2.144910	2.055163	1.9404990
cg00000109	0.5920380	0.1213538	1.0102038	1.358806	1.183005	0.6842241
cg00000165	NA	-0.1294922	-1.3828928	-1.575958	-1.344850	-1.6249474
cg00000236	-0.1178365	0.2370392	1.4682295	1.088427	1.078732	1.0698567
cg00000289	-1.5030408	-1.1319707	-0.3739629	-0.917323	-0.733984	-0.8942232

1.7 Preprocessing the data

```
 # Remove all NA value both in M-value and Methylation percentage
infdata <- infdata[rowSums(is.na(exprs(infdata))) == 0,]
kable(head(exprs(infdata)))
```

	GSM1632902_5775278068_R03C01	GSM1632927_5775278004_R04C02	GSM1632975_5775446011_R02C02	GSM1632987_5775446011_R04C02	GSM1632998_5775278034_R03C01	GSM1633016_5775278003_R02C02
cg00000029	0.4860570	-0.6579185	-2.5312223	-2.1820946	-1.6710701	-1.1999622
cg00000108	1.3577935	1.2145644	2.4626122	2.1449103	2.0551629	1.9404990

	GSM1632902_5775278068_R03C01	GSM1632927_5775278004_R04C02	GSM1632975_5775446011_R02C02	GSM1632987_5775446011_R04C02	GSM1632998_5775278034_R03C01	GSM1633016_5775278003_R02C02
cg00000109	0.5920380	0.1213538	1.0102038	1.3588058	1.1830048	0.6842241
cg00000236	-0.1178365	0.2370392	1.4682295	1.0884269	1.0787324	1.0698567
cg00000289	-1.5030408	-1.1319707	-0.3739629	-0.9173230	-0.7339840	-0.8942232
cg00000292	1.7440042	1.4999111	0.6158344	0.6148945	0.6081127	1.7850728

1.8 Explore preprocessed data

```
kable(head(exprs(infdata)))
```

	GSM1632902_5775278068_R03C01	GSM1632927_5775278004_R04C02	GSM1632975_5775446011_R02C02	GSM1632987_5775446011_R04C02	GSM1632998_5775278034_R03C01	GSM1633016_5775278003_R02C02
cg00000029	0.4860570	-0.6579185	-2.5312223	-2.1820946	-1.6710701	-1.1999622
cg00000108	1.3577935	1.2145644	2.4626122	2.1449103	2.0551629	1.9404990
cg00000109	0.5920380	0.1213538	1.0102038	1.3588058	1.1830048	0.6842241
cg00000236	-0.1178365	0.2370392	1.4682295	1.0884269	1.0787324	1.0698567
cg00000289	-1.5030408	-1.1319707	-0.3739629	-0.9173230	-0.7339840	-0.8942232
cg00000292	1.7440042	1.4999111	0.6158344	0.6148945	0.6081127	1.7850728

1.9 Change samplNAMES to somthing more comprehensible

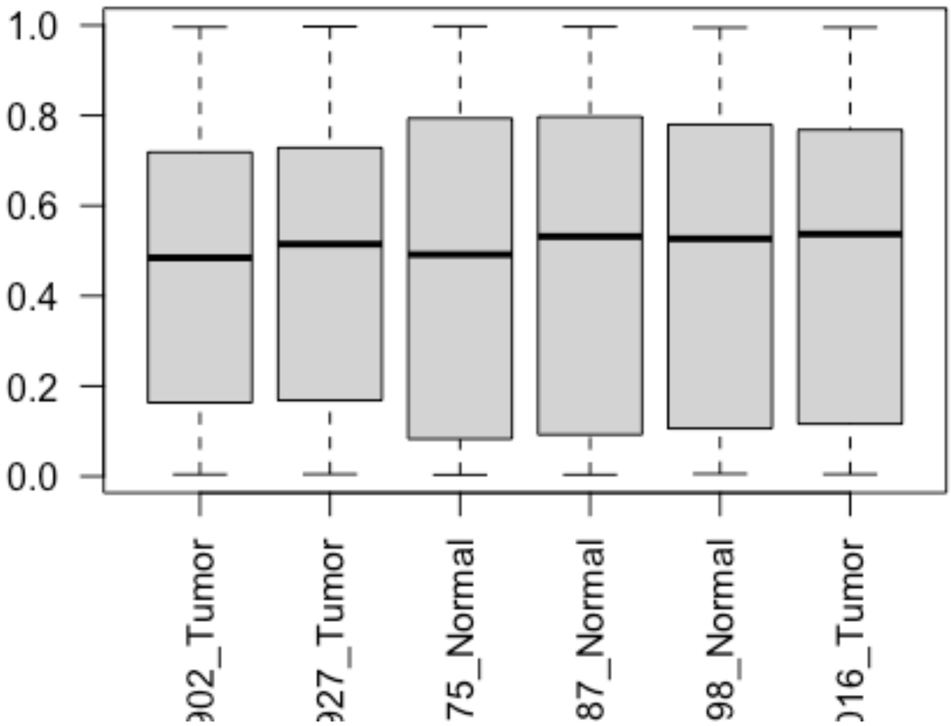
```
sampleNames(infdata) <- paste(annot[,2], annot[,3], sep = "_")
```

1.10 Remove probes for which calling p-value insufficient

```
infdata_filt <- pfilter(infdata)

## 0 samples having 1 % of sites with a detection p-value greater than 0.05
 were removed
## Samples removed:
## 1675 sites were removed as beadcount <3 in 5 % of samples
## 0 sites having 1 % of samples with a detection p-value greater than 0.05
 were removed
```

1.11 Comparison of average methylation between control and cancer samples

```
boxplot(betas(infdata_filt), las=2)
```



```
control <- (infdata_filt[,grep("Normal",annot[,3])])
cancer <- (infdata_filt[,grep("Tumor",annot[,3])])

meth_mean_CAF <- rep(0,ncol(cancer))
meth_mean_NAF <- rep(0,ncol(control))

for (i in 1:ncol(cancer)){
  meth_mean_CAF[i] <- mean(betas(cancer[,i]))
}

for (i in 1:ncol(control)){
  meth_mean_NAF[i] <- mean(betas(control[,i]))
}

meth_mean_CAF

## [1] 0.4616373 0.4738240 0.4737487

meth_mean_NAF

## [1] 0.4585227 0.4732641 0.4703541

t_test_res <- t.test(meth_mean_NAF, meth_mean_CAF, var.equal = F)
t_test_res

##
##  Welch Two Sample t-test
##
## data:  meth_mean_NAF and meth_mean_CAF
## t = -0.38885, df = 3.9549, p-value = 0.7174
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.01925677  0.01454405
## sample estimates:
## mean of x mean of y
## 0.4673803 0.4697367
```

1.12 Normalization and QC

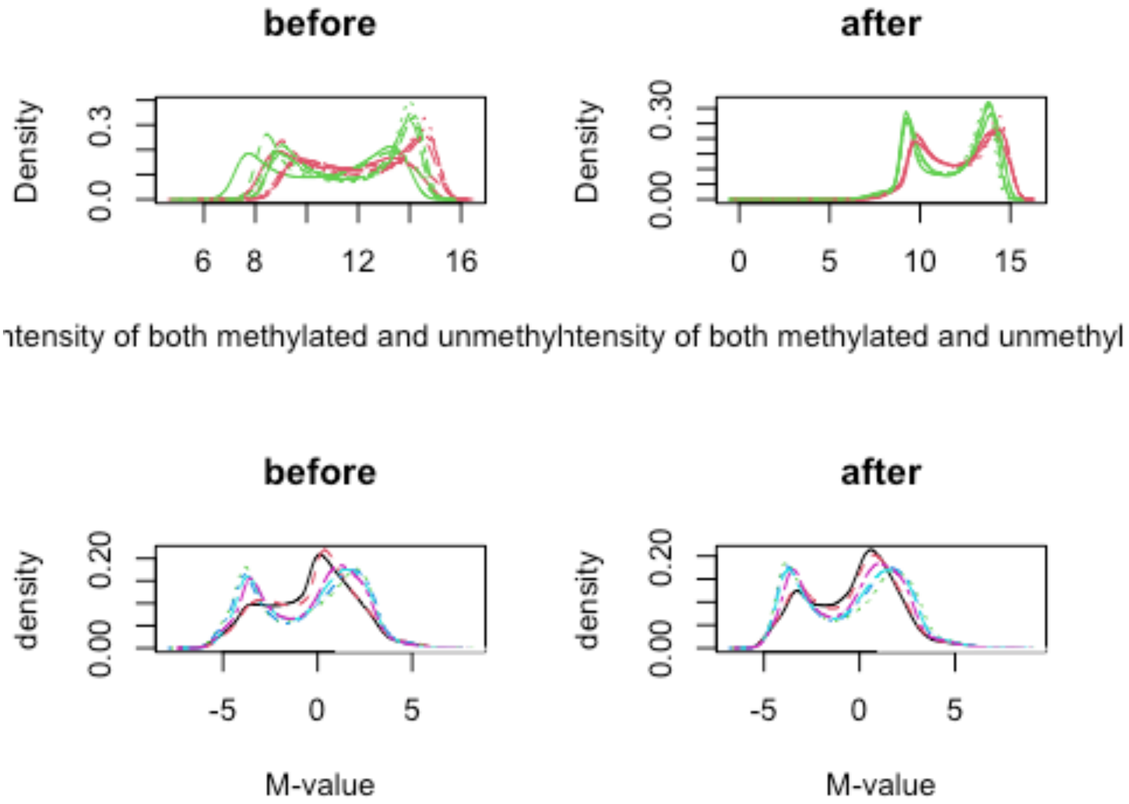
```
infdata_norm <- dasen(infdata_filt)
head(infdata_norm)

##
## Object Information:
## MethyLumiSet (storageMode: lockedEnvironment)
## assayData: 6 features, 6 samples
##   element names: betas, methylated, methylated.N, NBeads, pvals, unmethy
lated, unmethylated.N
## protocolData: none
## phenoData
##   sampleNames: GSM1632902_Tumor GSM1632927_Tumor ... GSM1633016_Tumor
##     (6 total)
##   varLabels: barcode
##   varMetadata: labelDescription
## featureData
##   featureNames: cg00000029 cg00000108 ... cg00000292 (6 total)
##   fvarLabels: Probe_ID DESIGN COLOR_CHANNEL
##   fvarMetadata: labelDescription
## experimentData: use 'experimentData(object)'
## Annotation: IlluminaHumanMethylation450k
## Major Operation History:
##            submitted           finished
## 1 2022-12-23 21:12:43 2022-12-23 21:12:54
## 2 2022-12-23 21:12:43 2022-12-23 21:12:54
## 3 2022-12-23 21:12:58 2022-12-23 21:12:59
## 4 2022-12-23 21:13:00 2022-12-23 21:13:01
## 5 2022-12-23 21:13:03 2022-12-23 21:13:04
## 6 2022-12-23 21:13:04 2022-12-23 21:13:04
## 7 2022-12-23 21:13:09 2022-12-23 21:13:16
## 8 2022-12-23 21:13:16 2022-12-23 21:13:16
##                                                           command
## 1 NChannelSetToMethyLumiSet2(NChannelSet = dats, parallel = parallel,
## 2                                                n = n, oob = oob)
## 3                                        Subset of 485577 features.
## 4                                        Subset of 451221 features.
## 5                                             Subset of 6 samples.
## 6                                        Subset of 449546 features.
## 7                          Normalized with dasen method (wateRmelon)
## 8                                            Subset of 6 features.
```

1.12.1 Mkake methylumi objects to check density and color bias adjustment

```
infdataM_norm <- as(infdata_norm, "MethyLumiM")
infdataM <- as(infdata_filt, "MethyLumiM")
```

1.12.2 Make QC plot

```
par(mfrow = c(2,2))
plotColorBias1D(infdataM, channel="both", main="before")
plotColorBias1D(infdataM_norm, channel="both",main="after")
density(infdataM, xlab="M-value", main="before", legend =F)
density(infdataM_norm, xlab="M-value", main="after", legend = F)
```

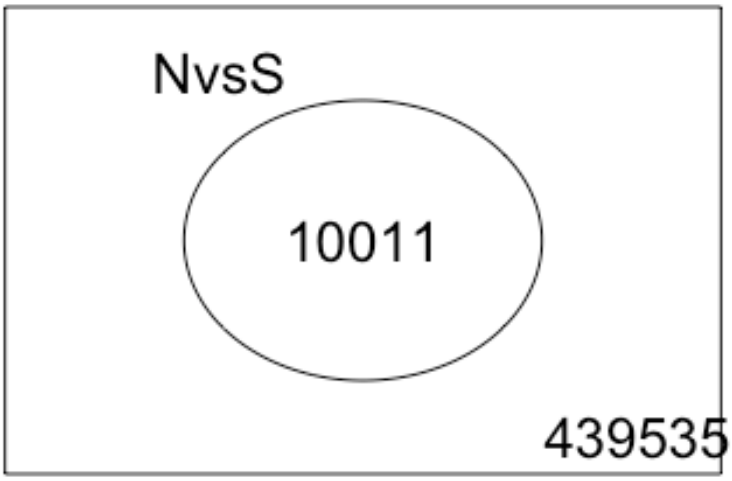


1.13 Differential expression analysis

```
  # Define design matrix
des <- factor(as.character(annot[,3]))
design <- model.matrix(~0 + des)
colnames(design) <- c("Tumor","Normal")
fit <- lmFit(infdataM_norm, design)

  # Fitting the model
cont.matrix <- makeContrasts(NvsS=Tumor-Normal,levels=design)
fit2 <- contrasts.fit(fit, cont.matrix)
fit2 <- eBayes(fit2)

# Getting top genes
kable(topTable(fit2, coef=1, adjust="BH"))
```

	Probe_ID	DESIGN	COLOR_CHANNEL	logFC	AveExpr	t	P.Value	adj.P.Val	B
cg05175020	cg05175020	I	Grn	-3.882420	-1.6578263	-17.02017	1e-07	0.0151440	7.142509
cg06995503	cg06995503	II	Both	-3.958196	-0.8458739	-16.04547	1e-07	0.0151440	6.882008
cg15908367	cg15908367	I	Red	-3.961045	-1.9822082	-15.51342	1e-07	0.0151440	6.726487
cg10303487	cg10303487	I	Red	-3.545936	-2.0892612	-15.00326	2e-07	0.0151440	6.567617
cg08269402	cg08269402	II	Both	3.533199	2.1404689	14.73035	2e-07	0.0151440	6.478455
cg02443967	cg02443967	I	Red	3.349599	4.4181254	14.72578	2e-07	0.0151440	6.476936
cg18428180	cg18428180	I	Grn	3.258170	-1.7652452	14.58903	2e-07	0.0151440	6.431088
cg13279673	cg13279673	I	Red	-3.031308	-1.9826669	-13.98426	3e-07	0.0182476	6.218655
cg00582971	cg00582971	I	Red	-3.428868	-1.9090109	-13.83636	4e-07	0.0182476	6.164196
cg13232075	cg13232075	II	Both	-4.079101	-1.1824778	-13.43286	5e-07	0.0185934	6.010307

```
results <- decideTests(fit2)
vennDiagram(results)
```



```
summary(results)

##         NvsS
## Down     7754
## NotSig 439535
## Up       2257
```

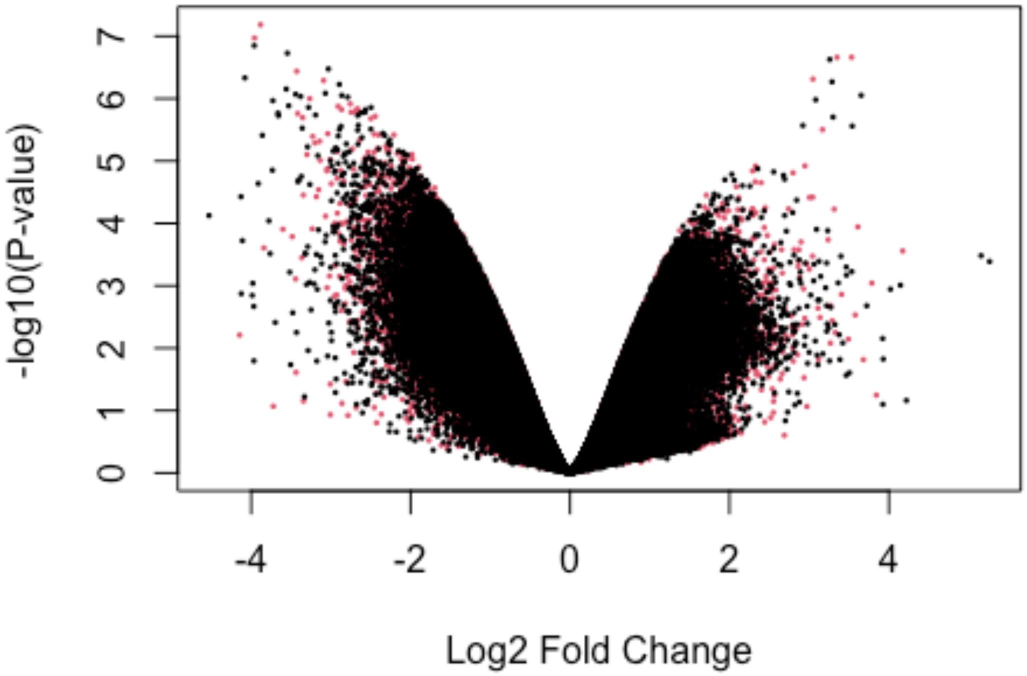
1.14 DE results

```
LIMMAout <- topTable(fit2, adjust="BH", number=nrow(exprs(infdataM)))
kable(head(LIMMAout, 10))
```

	Probe_ID	DESIGN	COLOR_CHANNEL	logFC	AveExpr	t	P.Value	adj.P.Val	B
cg05175020	cg05175020	I	Grn	-3.882420	-1.6578263	-17.02017	1e-07	0.0151440	7.142509
cg06995503	cg06995503	II	Both	-3.958196	-0.8458739	-16.04547	1e-07	0.0151440	6.882008
cg15908367	cg15908367	I	Red	-3.961045	-1.9822082	-15.51342	1e-07	0.0151440	6.726487
cg10303487	cg10303487	I	Red	-3.545936	-2.0892612	-15.00326	2e-07	0.0151440	6.567617
cg08269402	cg08269402	II	Both	3.533199	2.1404689	14.73035	2e-07	0.0151440	6.478455
cg02443967	cg02443967	I	Red	3.349599	4.4181254	14.72578	2e-07	0.0151440	6.476936
cg18428180	cg18428180	I	Grn	3.258170	-1.7652452	14.58903	2e-07	0.0151440	6.431088
cg13279673	cg13279673	I	Red	-3.031308	-1.9826669	-13.98426	3e-07	0.0182476	6.218655
cg00582971	cg00582971	I	Red	-3.428868	-1.9090109	-13.83636	4e-07	0.0182476	6.164196
cg13232075	cg13232075	II	Both	-4.079101	-1.1824778	-13.43286	5e-07	0.0185934	6.010307

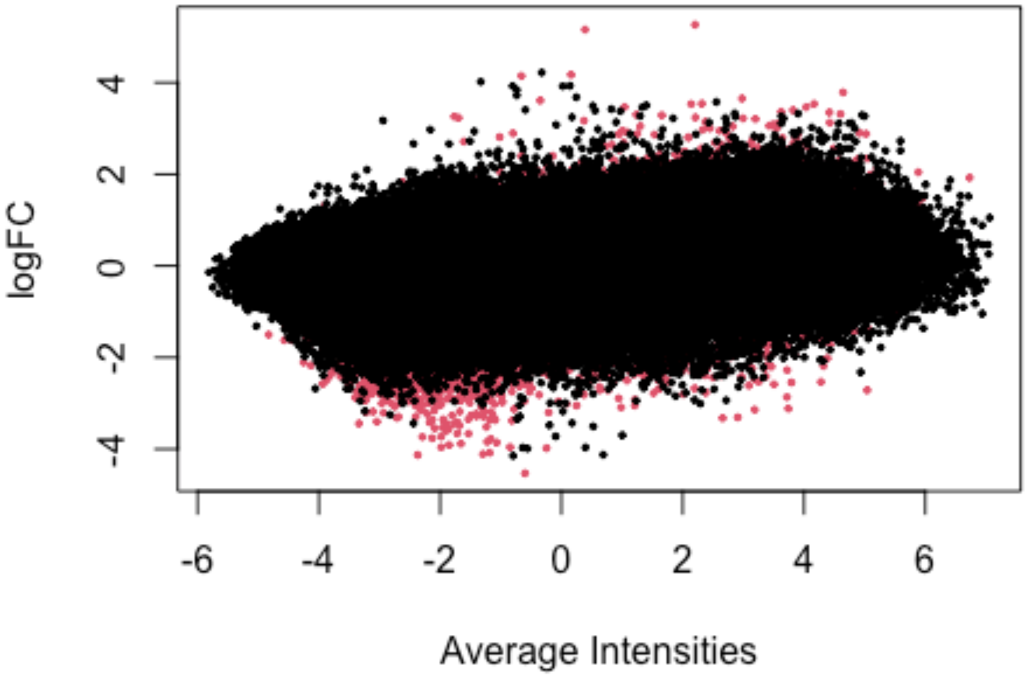
1.14.1 Volcano plot

```
  # There is few signifcant genes with threshold 0.05, so 0.15 is used inst
ead
volcanoplot(fit2, col = as.factor(LIMMAout$adj.P.Val < 0.15), style = "p-va
lue")
```



1.14.2 MA plot

```
plot(LIMMAout$AveExpr, LIMMAout$logFC,
     col = as.factor(LIMMAout$adj.P.Val < 0.05), pch = 20, cex = 0.50,
     xlab = "Average Intensities", ylab = "logFC")
```



1.15 Functional annotation of limma results

1.15.1 Load annotation and sort alphabetically on probe name

```
data("probe.features")
annotation_MA <- probe.features
kable(head(annotation_MA))
```

	CHR	MAPINFO	Strand	Type	gene	feature	cgi	feat.cgi	UCSC_CpG_Islands_Name	DHS	Enhancer	Phantom	Probe_SNPs	Probe_SNPs_10
cg00000029	16	53468112	F	II	RBL2	TSS1500	shore	TSS1500-shore	chr16:53468284-53469209	TRUE	NA			
cg00000108	3	37459206	F	II	C3orf35	Body	opensea	Body-opensea		NA	NA		rs9857774	
cg00000109	3	171916037	F	II	FNDC3B	Body	opensea	Body-opensea		NA	NA	low-CpG:173398671-173398760	rs9864492	
cg00000165	1	91194674	R	II		IGR	shore	IGR-shore	chr1:91190489-91192804	NA	TRUE			

	CHR	MAPINFO	Strand	Type	gene	feature	cgi	feat.cgi	UCSC_CpG_Islands_Name	DHS	Enhancer	Phantom	Probe_SNPs	Probe_SNPs_10
								shore						
cg00000236	8	42263294	R	II	VDAC3	3’UTR	opensea	3’UTR-opensea		NA	NA			
cg00000289	14	69341139	F	II	ACTN1	3’UTR	shore	3’UTR-shore	chr14:69341427-69341820	NA	NA			

```
annotation_MA <- annotation_MA[sort(rownames(annotation_MA),index.return =
 T)$ix,]
```

1.15.2 Check if all probes are present in both sets

```
dim(LIMMAout)

## [1] 449546      9

sum(LIMMAout$Probe_ID%in%rownames(annotation_MA))

## [1] 449546

sum(rownames(annotation_MA)%in%LIMMAout$Probe_ID)

## [1] 449546

  # Also check the reverse so no duplicate rows are present in annotation
```

1.15.3 Since more probes are present in the annotation file, remove unnecessary probes

```
annotation_MA <- annotation_MA[rownames(annotation_MA)%in%LIMMAout$Probe_I
D,]
```

1.15.4 Sort LIMMA output alphabetically on probe name

```
LIMMAout_sorted <- LIMMAout[sort(LIMMAout$Probe_ID,index.return=T)$ix,]
```

1.15.5 Add gene names to LIMMA output

```
LIMMAout_sorted$Gene <- annotation_MA$gene
LIMMAout_sorted$Feature <- annotation_MA$feature
LIMMAout_sorted$Chrom <- annotation_MA$CHR
LIMMAout_sorted$Pos <- annotation_MA$MAPINFO
LIMMAout_sorted$Chrom <- as.character(LIMMAout_sorted$Chrom)
LIMMAout_sorted$Gene <- as.character(LIMMAout_sorted$Gene)
LIMMAout_sorted$Feature <- as.character(LIMMAout_sorted$Feature)
```

1.16 Quantification of absolute methylation differences

1.16.1 Add gene names to LIMMA output

```
LIMMAout_sorted$Tumor_meth <- rowMeans(betas(infdata)[rownames(infdata)%in
%
                                                    LIMMAout_sorted$Probe_I
D,annot$`!Sample_characteristics_ch1`=="Tumor"])
LIMMAout_sorted$Control_meth <- rowMeans(betas(infdata)[rownames(infdata)%
in%
                                                      LIMMAout_sorted$Probe
_ID,annot$`!Sample_characteristics_ch1`=="Normal"])
LIMMAout_sorted$Abs_diff_meth <- abs(rowMeans(betas(infdata)[rownames(infd
ata)%in%
                                                          LIMMAout_sorted$P
robe_ID,annot$`!Sample_characteristics_ch1`=="Tumor"]) -
                                    rowMeans(betas(infdata)[rownames(infdat
a)
                                                          %in%LIMMAout_sort
ed$Probe_ID, annot$`!Sample_characteristics_ch1`=="Normal"]))
```

1.17 Resort results

```
LIMMAout_annot <- LIMMAout_sorted[sort(LIMMAout_sorted$P.Value,index.retur
n=T)$ix, c(1,12,13,10,11,4,7,8,5,14,15,16)]
# Sort on p-values to prevent errors in sorting due to equal FDR values
```

1.18 Interpretation results

1.18.1 Select CpGs in genic regions

```
sum(LIMMAout_annot$adj.P.Val<0.05)

## [1] 10011

sum(LIMMAout_annot$adj.P.Val[LIMMAout_annot$Gene!=""]<0.05)

## [1] 7435

LIMMAout_annot_gene <- LIMMAout_annot[LIMMAout_annot$Gene!="",]
```

1.18.2 Check genic results

```
kable(head(LIMMAout_annot_gene[c(4,5,6,8,10,11,12)]))
```

	Gene	Feature	logFC	adj.P.Val	Tumor_meth	Control_meth	Abs_diff_meth
cg05175020	TSC22D4	Body	-3.882420	0.015144	0.5941010	0.0560259	0.5380751
cg06995503	PFKP	3’UTR	-3.958196	0.015144	0.6847462	0.1055248	0.5792214
cg15908367	TSC22D4	Body	-3.961045	0.015144	0.5185570	0.0523593	0.4661977
cg10303487	DPYS	1stExon	-3.545936	0.015144	0.4638851	0.0580843	0.4058008

	Gene	Feature	logFC	adj.P.Val	Tumor_meth	Control_meth	Abs_diff_meth
cg08269402	HLA-DRB1	Body	3.533199	0.015144	0.6052022	0.9381730	0.3329707
cg02443967	TLL2	Body	3.349599	0.015144	0.9031419	0.9822053	0.0790634

```
topgenes_genic <- unique(LIMMAout_annot_gene$Gene[1:10])

for (i in 1:length(topgenes_genic)){
  LIMMAout_subset <- LIMMAout_annot_gene[(LIMMAout_annot_gene$Gene==topgen
es_genic [i]) &
                                         (LIMMAout_annot_gene$adj.P.Val<0.05
) &
                                         (abs(LIMMAout_annot_gene$logFC)>2),
]
  kable(LIMMAout_subset[sort(LIMMAout_subset$Pos,index.return=T)$ix,c (4,5
,6,8,10,11,12)])
  }
```

1.18.3 Select CpGs in promoter regions

```
LIMMAout_annot_prom <- LIMMAout_annot_gene[grepl("TSS",LIMMAout_annot_gene
$Feature) | (LIMMAout_annot_gene$Feature=="1stExon"),]

kable(head(LIMMAout_annot_prom))
```

Probe_ID	Chrom	Pos	Gene	Feature	logFC	P.Value	adj.P.Val	AveExpr
cg10303487	8	105479058	DPYS	1stExon	-3.545936	2e-07	0.0151440	-2.089261
cg18428180	6	24646492	KIAA0319	TSS1500	3.258170	2e-07	0.0151440	-1.765245
cg00582971	5	178422128	GRM6	TSS200	-3.428868	4e-07	0.0182476	-1.909011
cg22674699	2	176987918	HOXD9	1stExon	-3.559862	7e-07	0.0200073	-1.859010
cg10989862	7	62809331	LOC100287834	TSS200	3.652897	9e-07	0.0200073	2.978405
cg25774643	11	627175	SCT	TSS200	-3.379502	9e-07	0.0200073	-0.814335

```
## Look for multiple CpG in promoter regions undergoing similar methylation
 differences


topgenes_prom <- unique(LIMMAout_annot_prom$Gene[1:10])

for (i in 1:length(topgenes_prom)){
  LIMMAout_subset <- LIMMAout_annot_prom[(LIMMAout_annot_prom$Gene == topge
nes_prom[i]) & (LIMMAout_annot_prom$adj.P.Val < 0.10),]
  if (nrow(LIMMAout_subset) > 1) {kable(LIMMAout_subset[sort(LIMMAout_subse
t$Pos, index.return =T)$ix, c(4,5,6,8,10,11,12)])
    }
 }
```

1.19 Gene Set Analysis

Goana uses Entrez gene identifiers, we used to convert our gene symbols to entrez ids. For thus purpose we use the org.Hs.eg.db package.

```
LIMMAout_filtered <- LIMMAout_annot[LIMMAout_annot$adj.P.Val < 0.05,]

EntrezIDs <- mapIds(org.Hs.eg.db, LIMMAout_filtered$Gene, "ENTREZID", "SYMB
OL")

## 'select()' returned 1:many mapping between keys and columns
```

1.19.1 subset for non duplicated and mapped genes

```
EntrezIDs <- EntrezIDs[!(duplicated(EntrezIDs) | is.na(EntrezIDs))]
kable(t(head(EntrezIDs)))
```

TSC22D4	PFKP	DPYS	HLA-DRB1	TLL2	KIAA0319
81628	5214	1807	3123	7093	9856

1.19.2 Make table for comaprison with other methods

```
LIMMAout_filtered$EntrezIDs <- EntrezIDs[match(LIMMAout_filtered$Gene, nam
es(EntrezIDs))]
```

1.19.3 Overexpression analysis with goana

```
goanaOUT <- goana(de=unlist(EntrezIDs), species = "Hs", trend = T)
```

1.19.4 FDR multiple

```
goanaOUT <- goanaOUT[order(goanaOUT$P.DE, decreasing = F),]
goanaOUT$FDR.DE <- p.adjust(goanaOUT$P.DE, method = "BH")

topGOcpg <- topGO(goanaOUT, ontology = "BP", number = 50)
kable(head(topGOcpg))
```

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0048856	anatomical structure development	BP	5785	1493	0	0
GO:0007275	multicellular organism development	BP	4804	1304	0	0
GO:0048731	system development	BP	4345	1209	0	0
GO:0032502	developmental process	BP	6355	1574	0	0
GO:0007399	nervous system development	BP	2408	758	0	0
GO:0009653	anatomical structure morphogenesis	BP	2746	831	0	0

```
kable(head(topGOcpg[order(topGOcpg$N),]))
```

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0007610	behavior	BP	606	227	0	0
GO:0048598	embryonic morphogenesis	BP	607	232	0	0
GO:0048812	neuron projection morphogenesis	BP	620	226	0	0

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0120039	plasma membrane bounded cell projection morphogenesis	BP	635	231	0	0
GO:0048858	cell projection morphogenesis	BP	639	231	0	0
GO:0032990	cell part morphogenesis	BP	658	233	0	0

1.20 Write data for comaprison of results

```
write.table(unlist(EntrezIDs), sep = "\t", file = "EntrezIDs_CpG_results.tx
t")

CpG_GSA_res <- topGO(goanaOUT, ontology = "BP", number = 100)
write.table(CpG_GSA_res, sep="\t", file = "CpG_GSA_results.txt")
```

APPENDIX D

ChIP SEQUENCING ANALYSIS

ChIP sequencing analysis R markdown codes are shown in appendix D

Chip-seq_Analysis

Jihwan Lim & Inkyun Park

2022-12-23

1. Chip_seq Analysis

1.1 General info

Read length: 40bp.

Single/paired end sequencing: single end sequencing.

Started from fastq files provided by encode.

Platform used: Illumina HiSeq 2000.

GSE148461

1.2 Load in necessary packages

```
  library(DiffBind)
library(tidyverse)
library(GenomicRanges)
library(org.Hs.eg.db)
library(TxDb.Hsapiens.UCSC.hg38.knownGene)
library(AnnotationDbi)
library(knitr)
```

1.3 Read broadPeak

```
  # combined broadpeak file for combine of two untreated PC9 cell samples
d0 <- read.table("./H3K4me3_contorl_peaks.broadPeak", header=F,skip=1)
colnames(d0) <-c("seqnames","start","end","id","score","strand","enrichmen
t","log10p","log10q")

# combined broadpeak file for combine of two treated with Erlotinib for 11
days PC9 cell samples
d11 <- read.table("./H3K4me3_treat_peaks.broadPeak", header=F,skip=1)
colnames(d11) <-c("seqnames","start","end","id","score","strand","enrichme
nt","log10p","log10q")
```

1.3.1 add “chr” before chromosome ID (1 -> chr1)

```
 d0$seqnames = paste("chr", d0$seqnames, sep ="")
d11$seqnames <- paste("chr", d11$seqnames, sep ="")
```

1.3.2 Adjust strand data

```
 d0$strand <-as.factor("*")
d11$strand <-as.factor("*")

 kable(head(d0, 10))
```

seqnames	start	end	id	score	strand	enrichment	log10p	log10q
chr1	181420	181755	H3K4me3_contorl_peak_2	62	*	5.15813	9.18287	6.25897
chr1	198481	200617	H3K4me3_contorl_peak_3	1908	*	38.29220	194.76300	190.87600
chr1	354751	355415	H3K4me3_contorl_peak_4	38	*	4.21594	6.78472	3.89280
chr1	358757	359995	H3K4me3_contorl_peak_5	159	*	7.56905	18.98120	15.98590
chr1	376687	377391	H3K4me3_contorl_peak_6	63	*	4.80952	9.23648	6.31199
chr1	407045	407397	H3K4me3_contorl_peak_7	25	*	3.81579	5.43755	2.56516
chr1	587392	589027	H3K4me3_contorl_peak_8	205	*	9.05075	23.61080	20.59730
chr1	604836	605730	H3K4me3_contorl_peak_9	151	*	7.76668	18.14830	15.15310
chr1	642982	643240	H3K4me3_contorl_peak_10	29	*	3.88787	5.85650	2.98114
chr1	777660	780410	H3K4me3_contorl_peak_11	1389	*	27.70330	142.42700	138.99200

```
 kable(head(d11, 10))
```

seqnames	start	end	id	score	strand	enrichment	log10p	log10q
chr1	96528	97122	H3K4me3_treat_peak_2	29	*	3.96970	5.77147	2.92862
chr1	181453	181734	H3K4me3_treat_peak_3	40	*	4.76600	6.91585	4.03013
chr1	184611	184976	H3K4me3_treat_peak_4	15	*	3.65121	4.38874	1.58927
chr1	198390	200687	H3K4me3_treat_peak_5	819	*	20.26400	85.50930	81.90940
chr1	273441	273776	H3K4me3_treat_peak_6	18	*	3.08217	4.65624	1.84707
chr1	354748	355419	H3K4me3_treat_peak_7	27	*	4.18457	5.62488	2.78625
chr1	358516	360050	H3K4me3_treat_peak_8	105	*	6.75973	13.48730	10.51150
chr1	376676	377366	H3K4me3_treat_peak_9	75	*	6.03396	10.48380	7.53274
chr1	587250	589464	H3K4me3_treat_peak_10	126	*	7.48809	15.62090	12.63580
chr1	604889	605770	H3K4me3_treat_peak_11	91	*	6.46389	12.08260	9.12443

1.4 Analysis

1.4.1 Make GRanges object

```
 bed0 <- with(d0, GRanges(seqnames, IRanges(start, end), strand, score, re
fseq=id))
bed11 <- with(d11, GRanges(seqnames, IRanges(start, end), strand, score, re
fseq=id))
kable(head(bed0 ,10))
```

seqnames	start	end	width	strand	score	refseq
chr1	181420	181755	336	*	62	H3K4me3_contorl_peak_2
chr1	198481	200617	2137	*	1908	H3K4me3_contorl_peak_3
chr1	354751	355415	665	*	38	H3K4me3_contorl_peak_4
chr1	358757	359995	1239	*	159	H3K4me3_contorl_peak_5
chr1	376687	377391	705	*	63	H3K4me3_contorl_peak_6
chr1	407045	407397	353	*	25	H3K4me3_contorl_peak_7
chr1	587392	589027	1636	*	205	H3K4me3_contorl_peak_8
chr1	604836	605730	895	*	151	H3K4me3_contorl_peak_9
chr1	642982	643240	259	*	29	H3K4me3_contorl_peak_10
chr1	777660	780410	2751	*	1389	H3K4me3_contorl_peak_11

```
 kable(head(bed11, 10))
```

seqnames	start	end	width	strand	score	refseq
chr1	96528	97122	595	*	29	H3K4me3_treat_peak_2
chr1	181453	181734	282	*	40	H3K4me3_treat_peak_3
chr1	184611	184976	366	*	15	H3K4me3_treat_peak_4
chr1	198390	200687	2298	*	819	H3K4me3_treat_peak_5
chr1	273441	273776	336	*	18	H3K4me3_treat_peak_6
chr1	354748	355419	672	*	27	H3K4me3_treat_peak_7
chr1	358516	360050	1535	*	105	H3K4me3_treat_peak_8
chr1	376676	377366	691	*	75	H3K4me3_treat_peak_9
chr1	587250	589464	2215	*	126	H3K4me3_treat_peak_10
chr1	604889	605770	882	*	91	H3K4me3_treat_peak_11

1.4.2 Extract gene data

```
 hg38 <- genes(TxDb.Hsapiens.UCSC.hg38.knownGene)
```

1.4.3 Make overlap

```
 ranges0 <- subsetByOverlaps(hg38,bed0, ignore.strand = T)
ranges11 <- subsetByOverlaps(hg38,bed11, ignore.strand = T)
```

1.4.4 Get gene annotation

```
 symbols0 <- unique(ranges0@elementMetadata$gene_id)
bed_c <- AnnotationDbi::select(org.Hs.eg.db, symbols0, c('SYMBOL', 'GENENAME'))

symbols11 <- unique(ranges11@elementMetadata$gene_id)
bed_t <- AnnotationDbi::select(org.Hs.eg.db, symbols11, c('SYMBOL', 'GENENAME'))
```

1.4.5 Search for genes of interest

```
  colnames(bed_c) <- c("Entrez_ID","Gene_Symbol","Gene_Name")
colnames(bed_t) <- c("Entrez_ID","Gene_Symbol","Gene_Name")

  kable(head(bed_c[grepl("CDH",bed_c$Gene_Symbol),], 10))
```

	Entrez_ID	Gene_Symbol	Gene_Name
4	1000	CDH2	cadherin 2
18	1001	CDH3	cadherin 3
432	1004	CDH6	cadherin 6
479	1005	CDH7	cadherin 7
863	1006	CDH8	cadherin 8
957	1008	CDH10	cadherin 10
1130	1010	CDH12	cadherin 12
1160	1012	CDH13	cadherin 13
1172	1013	CDH15	cadherin 15
1207	1016	CDH18	cadherin 18

```
  kable(head(bed_t[grepl("CDH",bed_t$Gene_Symbol),], 10))
```

	Entrez_ID	Gene_Symbol	Gene_Name
4	1000	CDH2	cadherin 2
18	1001	CDH3	cadherin 3
398	1003	CDH5	cadherin 5
446	1004	CDH6	cadherin 6
496	1005	CDH7	cadherin 7
894	1006	CDH8	cadherin 8
995	1008	CDH10	cadherin 10
1181	1010	CDH12	cadherin 12
1215	1012	CDH13	cadherin 13
1227	1013	CDH15	cadherin 15

1.4.6 Save results

```
  write.table(bed_c,file="ChIPgenes_c.txt",col.names = T,row.names = F,quote = F, sep="\t")
```

```
write.table(bed_t,file="ChIPgenes_t.txt",col.names = T,row.names = F,quote
 = F, sep="\t")

# Sorting and only keep unique gnene
bed_c <- unique(sort(bed_c$Gene_Symbol))
bed_t <- unique(sort(bed_t$Gene_Symbol))
```

1.5 Visualization

1.5.1 Remove the unusual chromosome names

```
  subset_c <- d0[d0$seqnames %in% paste0("chr", c(1:21, "X", "Y")),]
subset_t <- d11[d11$seqnames %in% paste0("chr", c(1:21, "X", "Y")),]
```

1.5.2 Turn the strand information back into “.”

```
  subset_c$strand <- "."
subset_t$strand <- "."
```

1.5.3 Write to visualization file

```
  ## Write to visualization file
write('track type=broadPeak visibility=3 db=hg38 name="H3k4me" description=
"H3k4me enrichment"', file = "H3k4me3c_track.broadPeak")
write.table(subset_c, file = "H3k4me3c_track.broadPeak", append=T, sep = "\
t", quote =F, row.names=F, col.names=F)

write('track type=broadPeak visibility=3 db=hg38 name="H3k4me" description=
"H3k4me enrichment"', file = "H3k4me3t_track.broadPeak")
write.table(subset_t, file = "H3k4me3t_track.broadPeak", append=T, sep = "\
t", quote =F, row.names=F, col.names=F)
```

1.6 Differential enrichment analysis

Comparing the peaks identified by each of the treatment against each other. We can analyze what binding regions are present in control samples, but treated samples in PC9 cell lines (and vice versa)

1.6.1 Reading in Peaksets

```
  PC9 <- dba(sampleSheet="./PC9.csv")

## SRR11523573   EGFR-mutant control 1 macs

## SRR11523574   EGFR-mutant control 2 macs

## SRR11523575   EGFR-mutant Erlotinib 1 macs

## SRR11523576   EGFR-mutant Erlotinib 2 macs

  dbObj <- dba(PC9)
dbObj

## 4 Samples, 40513 sites in matrix (53895 total):
##             ID   Condition Treatment Replicate Intervals
## 1 SRR11523573 EGFR-mutant   control         1     34849
## 2 SRR11523574 EGFR-mutant   control         2     29996
## 3 SRR11523575 EGFR-mutant Erlotinib         1     48111
## 4 SRR11523576 EGFR-mutant Erlotinib         2     46695
```

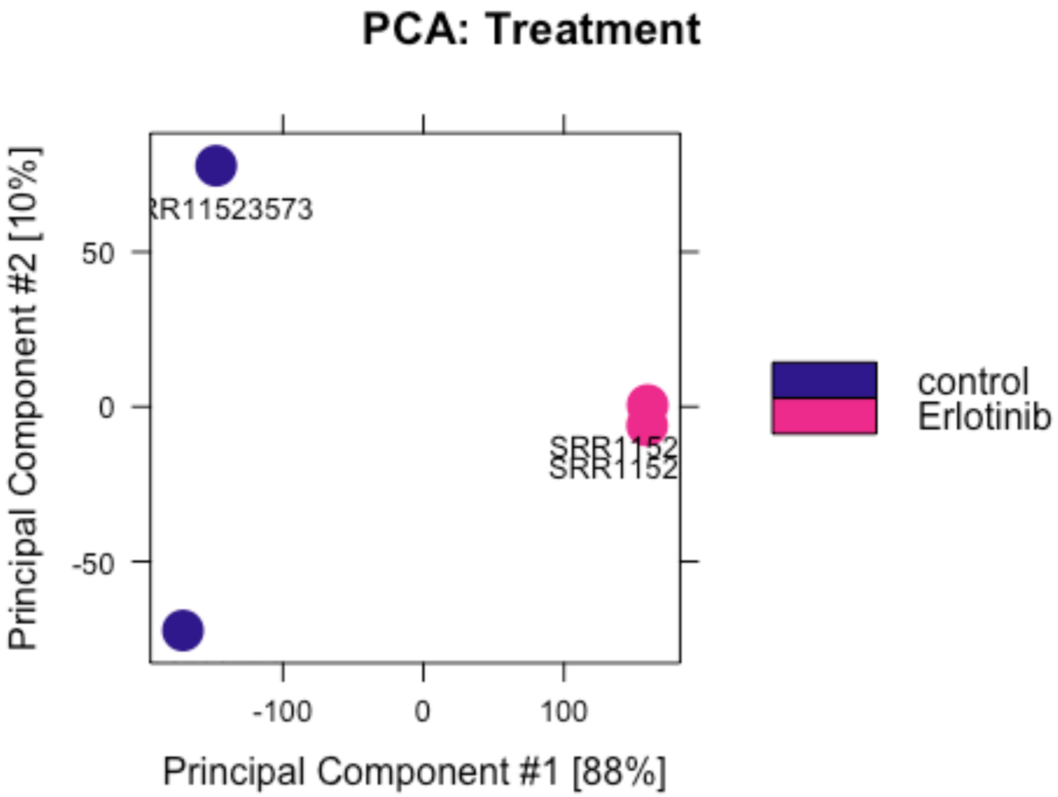
1.6.2 Affinity binding matrix

Compute count information for each of the peak/regions

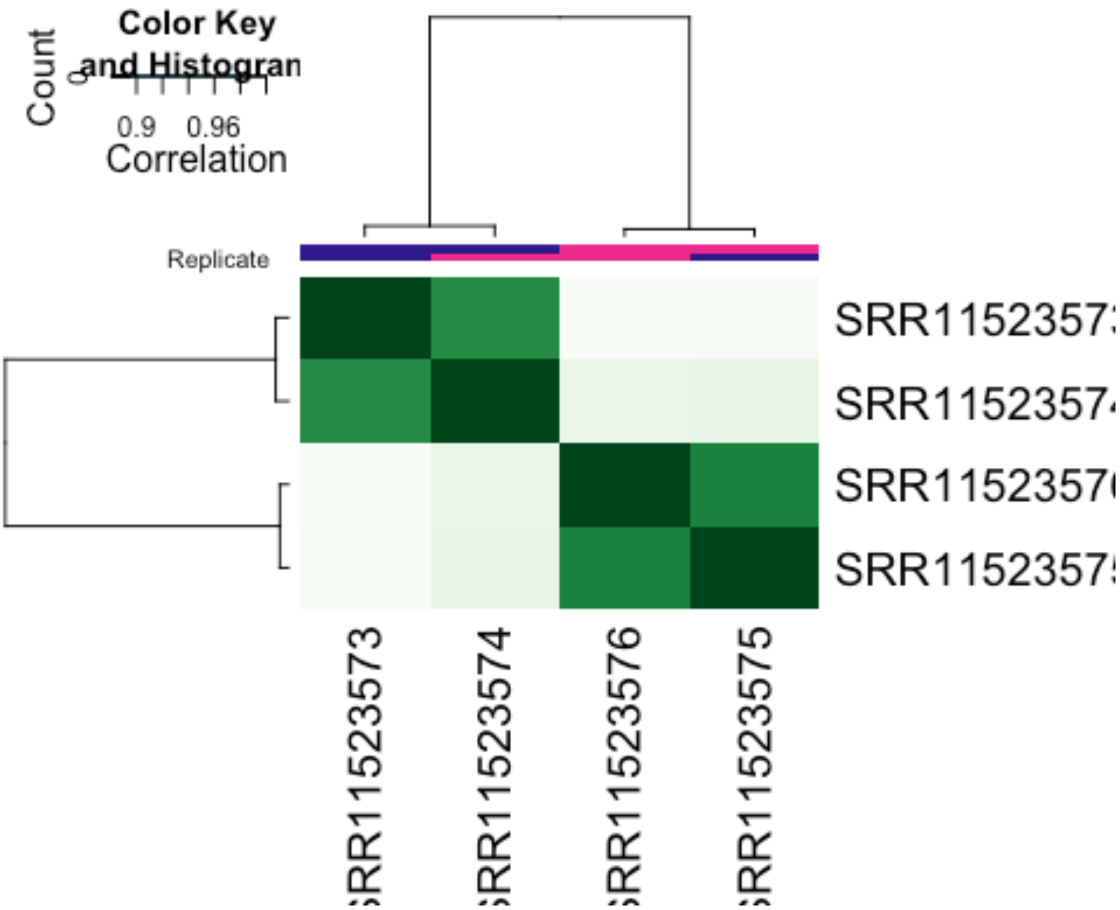
```
  dbObj <- dba.count(dbObj, bUseSummarizeOverlaps=T)

## Computing summits...

## Re-centering peaks...

  dbObj

## 4 Samples, 34710 sites in matrix:
##            ID   Condition Treatment Replicate    Reads FRiP
## 1 SRR11523573 EGFR-mutant   control         1 21466626 0.31
## 2 SRR11523574 EGFR-mutant   control         2 21760854 0.23
## 3 SRR11523575 EGFR-mutant Erlotinib         1 20643452 0.22
## 4 SRR11523576 EGFR-mutant Erlotinib         2 22949860 0.20
```

1.6.3 Explortry data analysis

```
  # PCA plot
dba.plotPCA(dbObj,  attributes=DBA_TREATMENT, label=DBA_ID)
```



```
  # Plot correlation heatmap
plot(dbObj)
```



1.6.4 Establishing a contrast

```
  dbObj <- dba.contrast(dbObj,minMembers = 2, categories=DBA_TREATMENT, des
ign = F, block=DBA_REPLICATE)
```

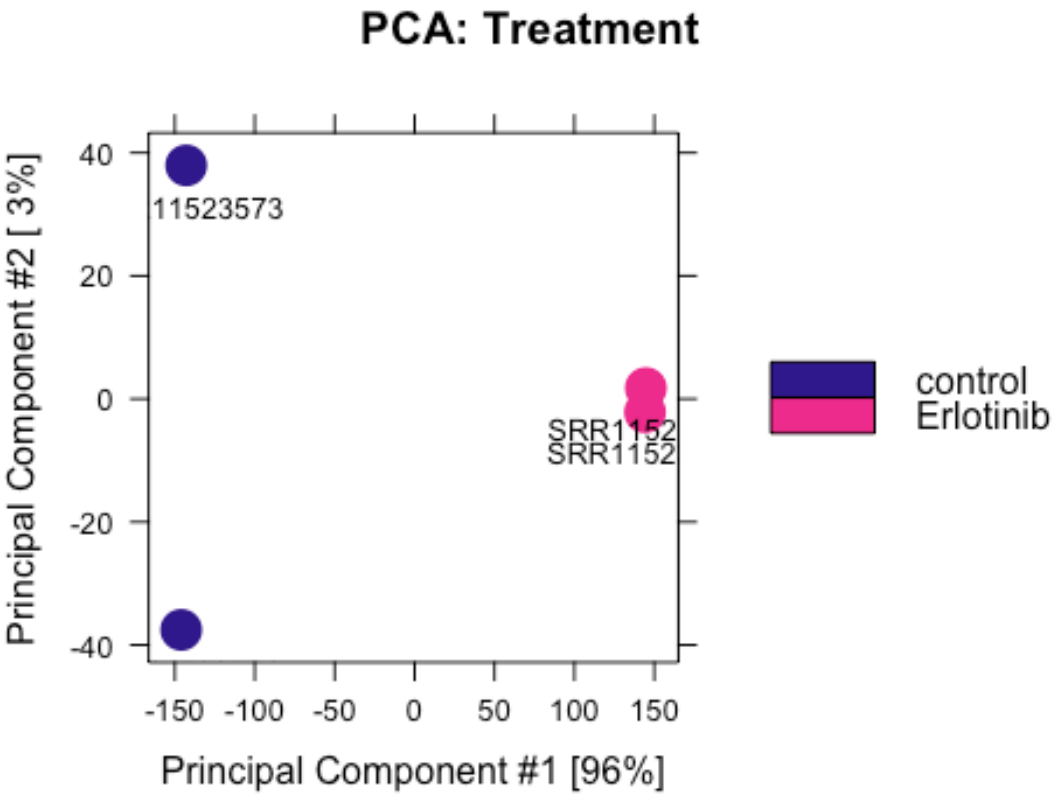
1.6.5 Perform the differential enrichment analysis

```
  # Perform both DESeq2 and edgeR method for analysis
dbObj <- dba.analyze(dbObj, method=DBA_ALL_METHODS,bGreylist = F)

  kable(dba.show(dbObj, bContrast=T))
```

Group	Samples	Group2	Samples2	Block1	Blk1Samps	Block2
control	2	Erlotinib	2	1	2	2

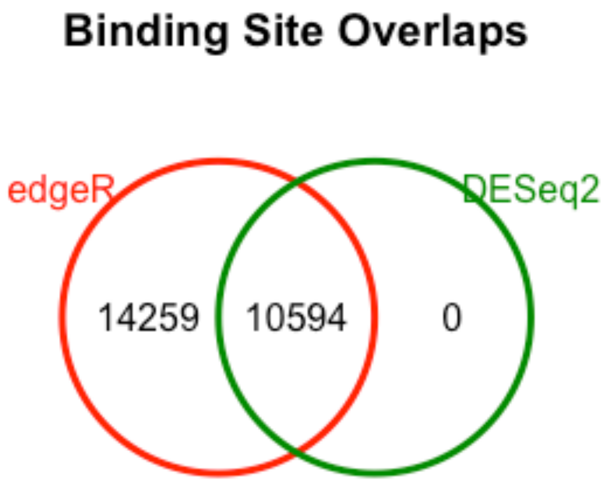
Blk2Samps	DB.edgeR	DB.edgeR.block	DB.DESeq2	DB.DESeq2.block
2	24853	24797	10594	8794

```
  # PCA Plot with regions identified as significant with under 0.05 FDR by
using DESeq2
dba.plotPCA(dbObj, contrast=1, method=DBA_DESEQ2, attributes=DBA_TREATMENT
, label=DBA_ID, th = 0.05)
```



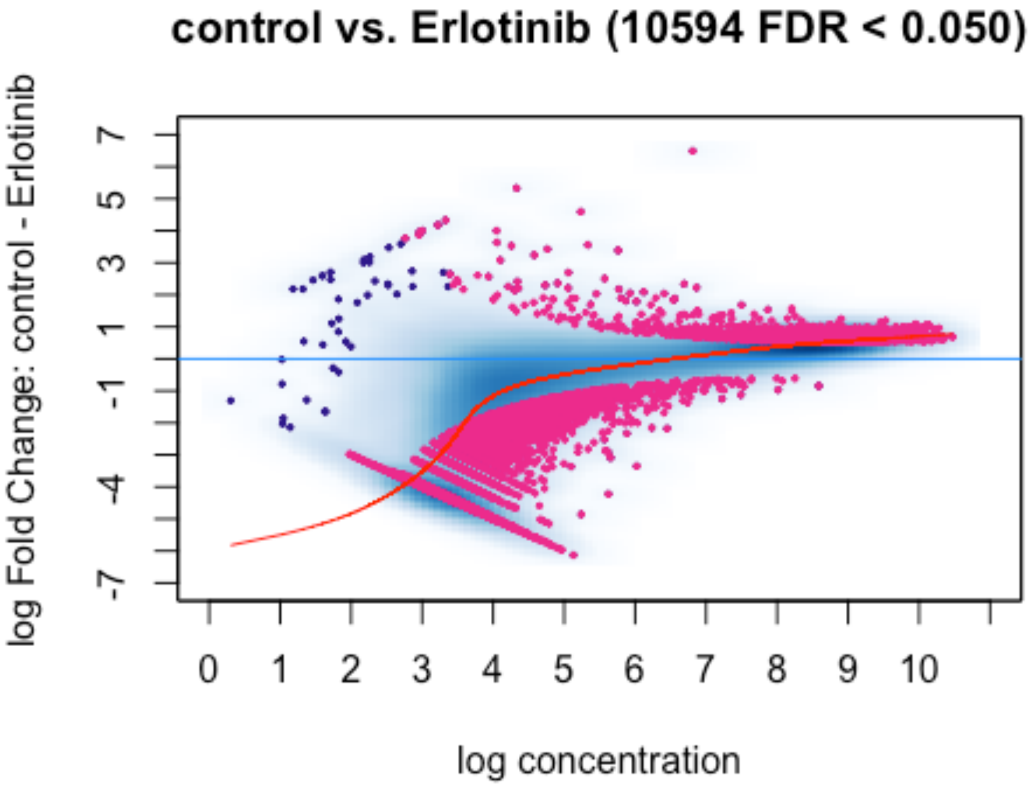
Visualizing the results

```
  dba.plotVenn(dbObj,contrast=1,method=DBA_ALL_METHODS)

## Generating report-based DBA object...
```

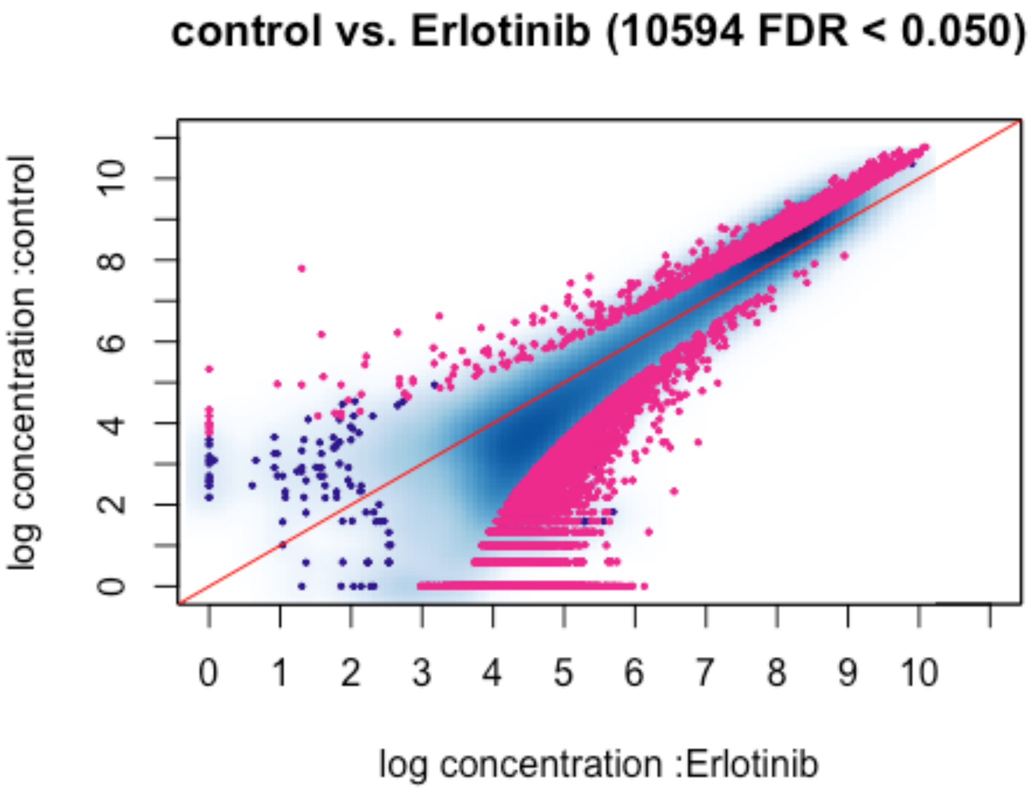


control vs. Erlotinib:DB:All

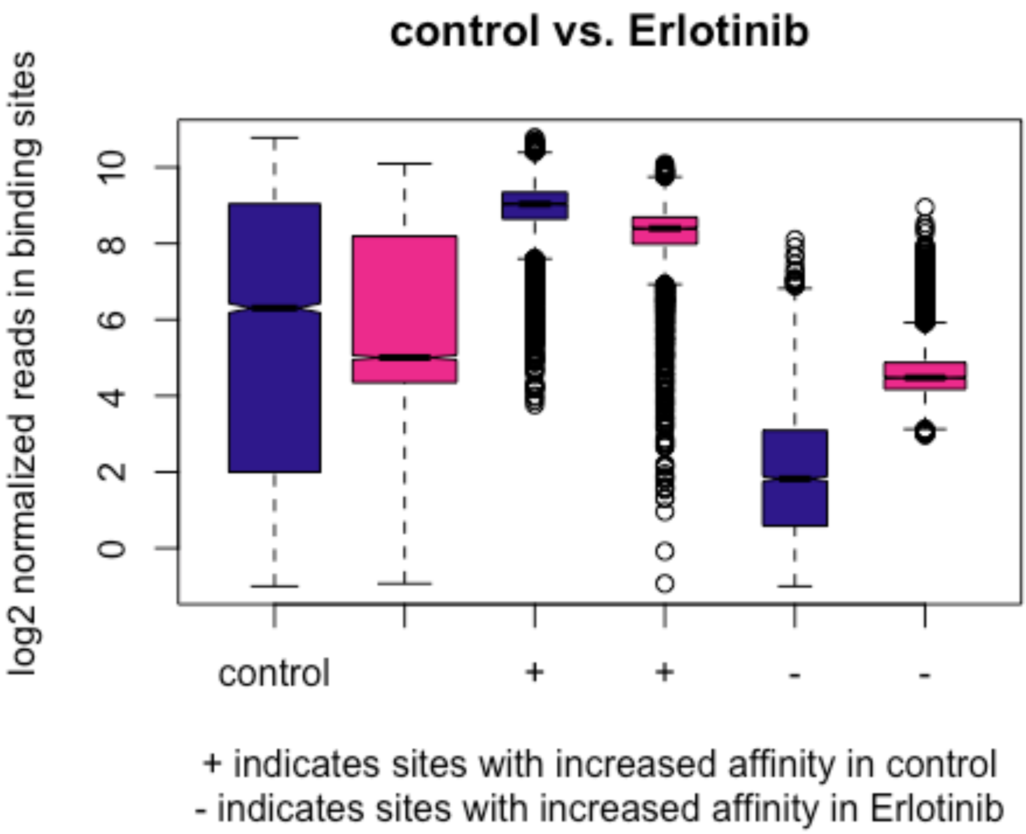
```
  # MA Plot
dba.plotMA(dbObj, method=DBA_DESEQ2)
```



```
dba.plotMA(dbObj, bXY=TRUE)
```



```r
pvals <- dba.plotBox(dbObj)
```



###

Extract results

```
  #Extract full results from DESeq2
res_deseq <- dba.report(dbObj, method=DBA_DESEQ2, contrast = 1, th=1)
kable(head(res_deseq ,10))
```

seqnames	start	end	width	strand	Conc	Conc_ctrl	Conc_Erlo	Fold	p.value	FDR
19	41755488	41755888	401	*	6.811171	7.795216	1.304559	6.490657	0	0e+00
16	28538336	28538736	401	*	5.623728	2.324153	6.548541	-4.224388	0	0e+00
17	64386043	64386443	401	*	5.761023	6.629357	3.241882	3.387474	0	0e+00
2	43073828	43074228	401	*	6.864933	7.585129	5.361012	2.224117	0	0e+00
12	89351777	89352177	401	*	8.206752	8.789948	7.212040	1.577908	0	0e+00
1	234656502	234656902	401	*	7.499585	8.107983	6.426903	1.681080	0	2e-07
4	11544194	11544594	401	*	5.234913	6.176356	1.582940	4.593416	0	2e-07
12	10171808	10172208	401	*	6.451271	4.985463	7.163177	-2.177713	0	2e-07
20	63254285	63254685	401	*	5.240887	1.333967	6.191979	-4.858012	0	3e-07
7	23247222	23247622	401	*	5.581191	3.530147	6.395692	-2.865545	0	4e-07

```
  # Add chr bbefore chromosome ID
diff_data <- as.data.frame(res_deseq)
```

```
diff_data$seqnames <- paste("chr", diff_data$seqnames, sep ="")
res_deseq@seqnames <- Rle(diff_data$seqnames)

  # Write to file
out <- as.data.frame(res_deseq)
write.table(out, file="./Control_vs_Erlotinib_deseq2.txt", sep="\t", quote
=F, row.names=F)
```

1.6.6 Extract bed files for furhter down stream analysis (Visualization)

```
  # Create bed files for each keeping only significant peaks (p < 0.05)

Control_enrich <- out %>%
  filter(FDR < 0.05 & Fold > 0) %>%
  dplyr::select(seqnames, start, end)

kable(head(Control_enrich ,10))
```

	seqnames	start	end
15837	chr19	41755488	41755888
13586	chr17	64386043	64386443
17190	chr2	43073828	43074228
7658	chr12	89351777	89352177
3149	chr1	234656502	234656902
23822	chr4	11544194	11544594
5607	chr11	65214062	65214462
26454	chr5	142324370	142324770
14684	chr19	1748286	1748686
3834	chr10	47407337	47407737

```
 Control_enrich <- Control_enrich[Control_enrich$seqnames %in% paste0("chr
", c(1:21, "X", "Y")),]
# Write to file
write.table(Control_enrich, file="./Control_enriched.bed", sep="\t", quote
=F, row.names=F, col.names=F)

Erlotinib_enrich <- out %>%
  filter(FDR < 0.05 & Fold < 0) %>%
  dplyr::select(seqnames, start, end)

kable(head(Erlotinib_enrich, 10))
```

	seqnames	start	end
11680	chr16	28538336	28538736
6686	chr12	10171808	10172208
20267	chr20	63254285	63254685
29152	chr7	23247222	23247622
16479	chr19	53962010	53962410

	seqnames	start	end
28978	chr7	6536521	6536921
25422	chr5	31854843	31855243
6661	chr12	8662226	8662626
19752	chr20	35263653	35264053
33733	chrKI270728.1	1791370	1791770

```
 Erlotinib_enrich <- Erlotinib_enrich[Erlotinib_enrich$seqnames %in% paste
0("chr", c(1:21, "X", "Y")),]
# Write to file
write.table(Erlotinib_enrich, file="./Erlotinib_enriched.bed", sep="\t", q
uote=F, row.names=F, col.names=F)
```

1.6.7 Explore data separately

```
 bed_control <- with(Control_enrich, GRanges(seqnames, IRanges(start, end)
))
bed_treat <- with(Erlotinib_enrich, GRanges(seqnames, IRanges(start, end)))

 ranges_control <- subsetByOverlaps(hg38,bed_control, ignore.strand = T)
ranges_treat <- subsetByOverlaps(hg38,bed_treat, ignore.strand = T)

 symbols_control <- unique(ranges_control@elementMetadata$gene_id)
bed_control <- AnnotationDbi::select(org.Hs.eg.db, symbols_control, c('SYM
BOL', 'GENENAME'))

## 'select()' returned 1:1 mapping between keys and columns

 symbols_treat <- unique(ranges_treat@elementMetadata$gene_id)
bed_treat <- AnnotationDbi::select(org.Hs.eg.db, symbols_treat, c('SYMBOL'
, 'GENENAME'))

## 'select()' returned 1:1 mapping between keys and columns

  colnames(bed_control) <- c("Entrez_ID","Gene_Symbol","Gene_Name")
colnames(bed_treat) <- c("Entrez_ID","Gene_Symbol","Gene_Name")

  kable(head(bed_control, 10))
```

Entrez_ID	Gene_Symbol	Gene_Name
1	A1BG	alpha-1-B glycoprotein
100009676	ZBTB11-AS1	ZBTB11 antisense RNA 1
100093630	SNHG8	small nucleolar RNA host gene 8
100101440	PMS2P7	PMS1 homolog 2, mismatch repair system component pseudogene 7
100113386	UCKL1-AS1	UCKL1 antisense RNA 1
100113407	TMEM170B	transmembrane protein 170B
100126348	MIR760	microRNA 760
100128055	SMARCA5-AS1	SMARCA5 antisense RNA 1
100128191	TMPO-AS1	TMPO antisense RNA 1

Entrez_ID	Gene_Symbol	Gene_Name
100128398	LOC100128398	uncharacterized LOC100128398

```
kable(head(bed_treat, 10))
```

Entrez_ID	Gene_Symbol	Gene_Name
1000	CDH2	cadherin 2
10001	MED6	mediator complex subunit 6
10006	ABI1	abl interactor 1
100126791	EGOT	eosinophil granule ontogeny transcript
100128076	LOC100128076	protein tyrosine phosphatase receptor type H pseudogene
100128590	SLC8A1-AS1	SLC8A1 antisense RNA 1
100128782	ERCC6L2-AS1	ERCC6L2 antisense RNA 1
100128885	LOC100128885	uncharacterized LOC100128885
100128905	LINC01960	long intergenic non-protein coding RNA 1960
100129075	KTN1-AS1	KTN1 antisense RNA 1

1.7 Find significant genes from each contorl and treatemnt ovelapping with other results

```
 overlap <- read.csv("./overlap_gene.csv")

 find_control <- bed_control[bed_control$Gene_Symbol %in% overlap$x,]
find_treat <- bed_treat[bed_treat$Gene_Symbol %in% overlap$x,]

 kable(head(find_control))
```

	Entrez_ID	Gene_Symbol	Gene_Name
1084	2013	EMP2	epithelial membrane protein 2
1219	22998	LIMCH1	LIM and calponin homology domains 1
1308	23242	COBL	cordon-bleu WH2 repeat protein
1402	23645	PPP1R15A	protein phosphatase 1 regulatory subunit 15A
1588	27242	TNFRSF21	TNF receptor superfamily member 21
1855	347735	SERINC2	serine incorporator 2

```
kable(head(find_treat))
```

	Entrez_ID	Gene_Symbol	Gene_Name
944	154810	AMOTL1	angiomotin like 1
1072	2070	EYA4	EYA transcriptional coactivator and phosphatase 4
1160	22998	LIMCH1	LIM and calponin homology domains 1
1328	26153	KIF26A	kinesin family member 26A
1364	27242	TNFRSF21	TNF receptor superfamily member 21
1391	283209	PGM2L1	phosphoglucomutase 2 like 1

APPENDIX E

RESULTS INTEGRATION

Results integration R markdown codes are shown in appendix E

Result Comparison

Jihwan Lim & Inkyun Park

2022-12-23

1. Comparison of Microarray Data and RNAseq Data

1.1 Data Preparation

```
  library(knitr)
# Load files with differentially expressed genes
DEgenes_microarray <- read.csv("DEgenes_microarray.csv", sep=',')
DEgenes_RNAseq <- read.csv("DEgenes_edger_RNAseq.csv", sep=',')

DEgenes_microarray$entrez_id <- as.character(DEgenes_microarray$entrez_id)
DEgenes_RNAseq$entrezIDs <- as.character(DEgenes_RNAseq$entrezIDs)

  # Brief look up on data
dim(DEgenes_microarray)

## [1] 4457    8

  dim(DEgenes_RNAseq)

## [1] 2425    7

  kable(head(DEgenes_microarray))
```

X	logFC	AveExpr	t	P.Value	adj.P.Val	B	entrez_id
1557371_a_at	2.648723	6.013027	16.57061	0	6.10e-06	14.06004	158376
1569608_x_at	3.446956	8.505440	14.35398	0	1.54e-05	12.49478	NA
242009_at	6.115768	7.757118	14.21228	0	1.54e-05	12.38401	6532
230469_at	4.058742	6.607292	13.91617	0	1.54e-05	12.14798	219790
206702_at	3.012743	5.616195	13.57615	0	1.54e-05	11.86911	7010
225660_at	3.532686	7.882041	13.41607	0	1.54e-05	11.73483	57556

```
  kable(head(DEgenes_RNAseq))
```

X	logFC	logCPM	LR	PValue	FDR	entrezIDs
ENSG00000185686	-12.810472	3.5336159	22.92416	1.70e-06	0.0001726	NA
ENSG00000060718	-10.247213	6.3031616	30.16906	0.00e+00	0.0000084	NA
ENSG00000286037	-10.104493	0.8760820	22.00106	2.70e-06	0.0002524	NA
ENSG00000164093	-9.753728	0.8633366	17.10830	3.53e-05	0.0017767	NA
ENSG00000257342	-9.466246	0.2626271	31.72162	0.00e+00	0.0000043	NA
ENSG00000219159	-9.384153	0.2720294	18.22322	1.96e-05	0.0011756	NA

```
  # Make all gene names to entrez IDs to compare each other
library("org.Hs.eg.db")

  DEgenes_microarray <-
DEgenes_microarray[!duplicated(DEgenes_microarray$entrez_id), ]

entrez_ids_microarray <- na.omit(DEgenes_microarray$entrez_id)
entrez_ids_RNAseq <- na.omit(DEgenes_RNAseq$entrezIDs)
```

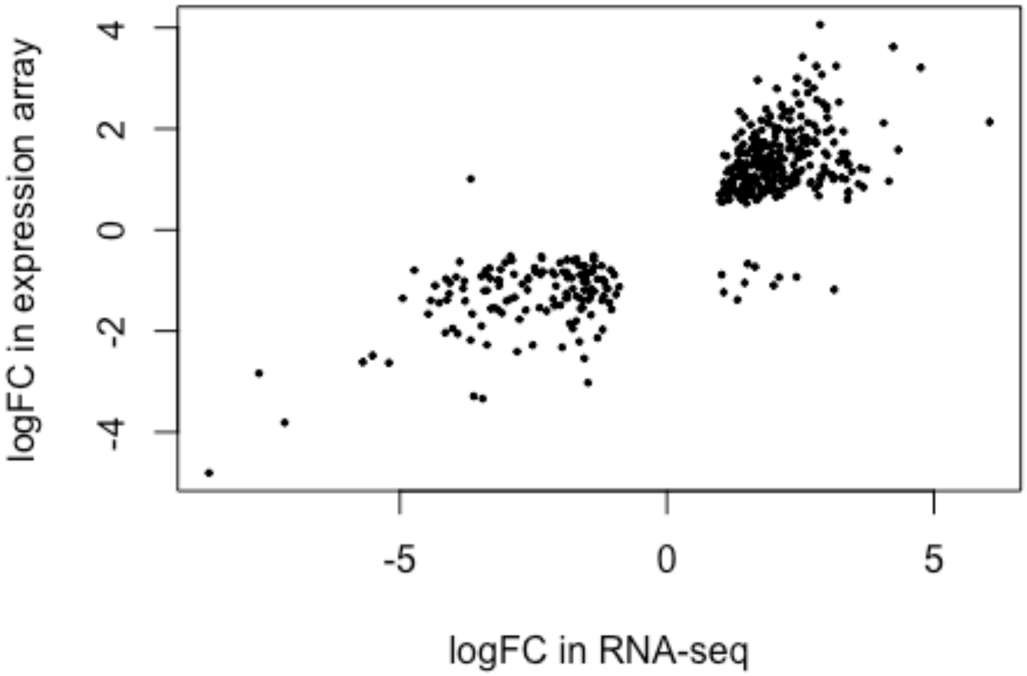
In microarray and RNAseq, we found out 420 genes are statistically significant differential expressed genes. This founding will be further investigated by scatterplot of logFC betwen microarray and RNAseq.

```
  # Compare genes
common_genes_MvR <- intersect(entrez_ids_microarray, entrez_ids_RNAseq)

head(common_genes_MvR)

## [1] "219790" "2823"   "8436"   "104"    "2869"   "51208"

  length(common_genes_MvR)

## [1] 420
```

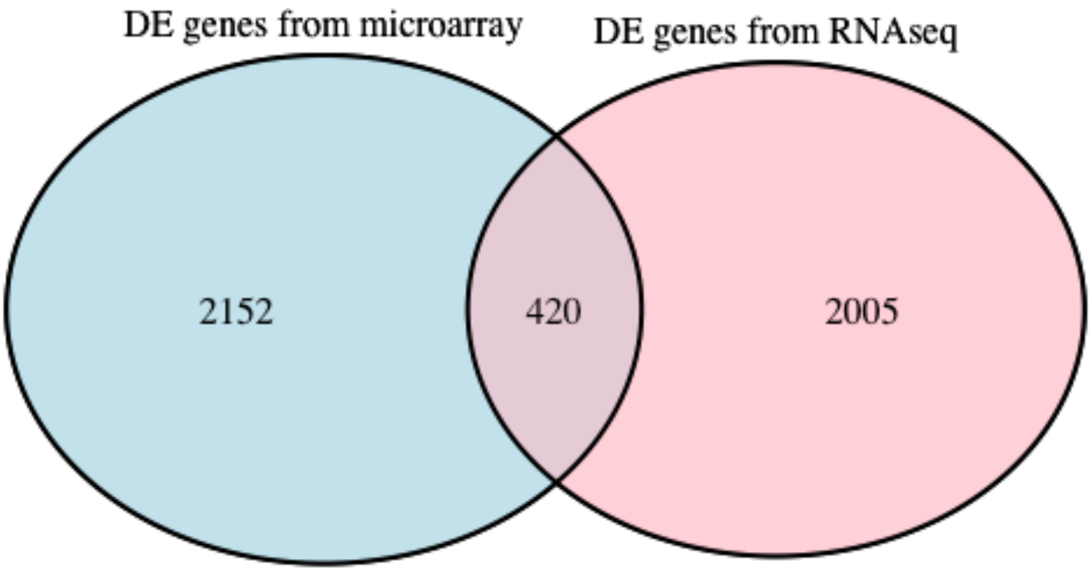
1.2 Visualization

Here, we will see trends of DE genes in common.

```
  # Filter genes not in common
DEgenes_microarray_filtered <- DEgenes_microarray[which(DEgenes_microarray
$entrez_id %in% common_genes_MvR), ]
DEgenes_microarray_filtered <- DEgenes_microarray_filtered[order(DEgenes_m
icroarray_filtered$entrez_id), ]

  DEgenes_RNAseq_filtered <- DEgenes_RNAseq[which(DEgenes_RNAseq$entrezIDs
%in% common_genes_MvR), ]
DEgenes_RNAseq_filtered <- DEgenes_RNAseq_filtered[order(DEgenes_RNAseq_fi
ltered$entrezIDs), ]

plot(DEgenes_RNAseq_filtered$logFC, DEgenes_microarray_filtered$logFC,
     xlab="logFC in RNA-seq",ylab="logFC in expression array",
     pch=20, cex=0.50)
```



```
  library("VennDiagram")

## Loading required package: grid

## Loading required package: futile.logger

  grid.newpage()
vennplot <- draw.pairwise.venn(area1 = nrow(DEgenes_microarray),
                 area2 = nrow(DEgenes_RNAseq),
                 cross.area = length(common_genes_MvR),
                 category = c("DE genes from microarray",
                             "DE genes from RNAseq"),
                 cat.pos = c(0, 0),
                 fill = c("light blue", "pink")
                 )
grid.draw(vennplot)
```



2. Comparison of Microarray Data, RNAseq Data and Infinum Data

2.1 Data Preparation

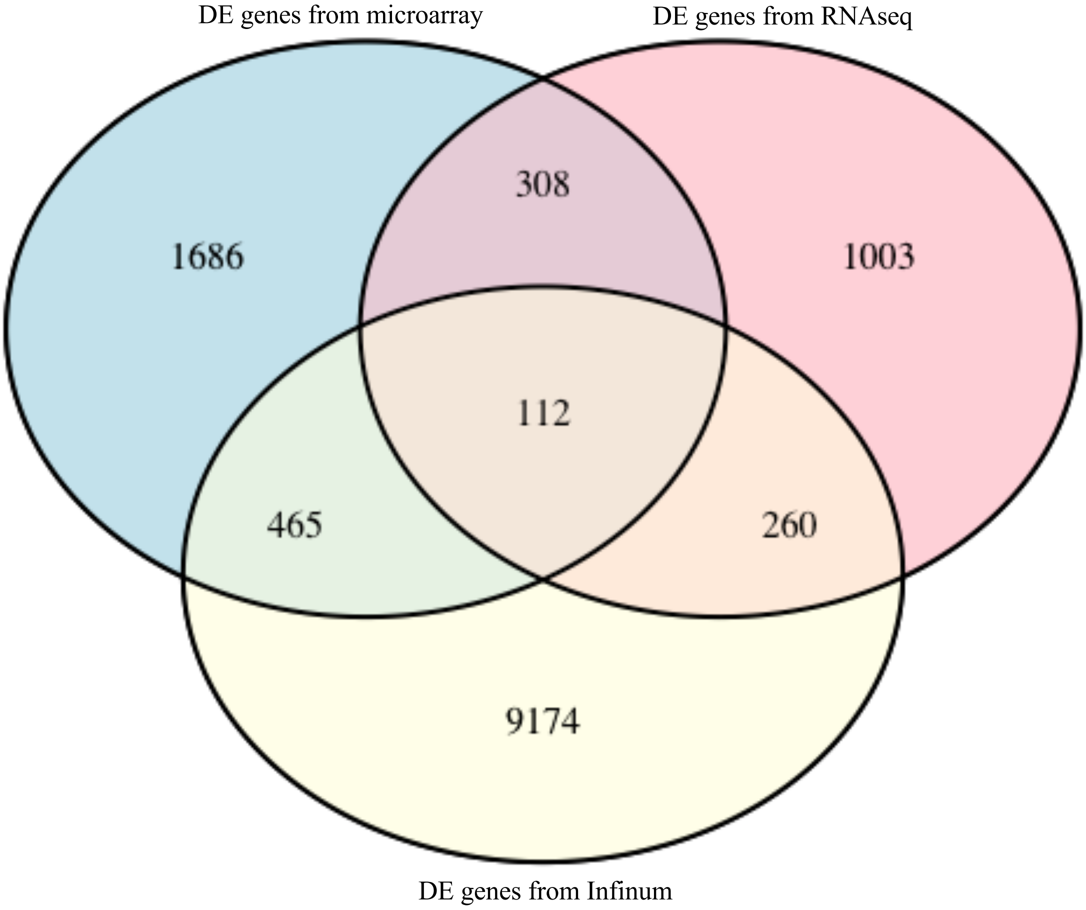
```
 DEgenes_methylation <- read.csv("DEgenes_methylation.csv", sep=",")
entrez_ids_infinum <- na.omit(DEgenes_methylation$EntrezIDs)

common_genes_MvRvI <- Reduce(intersect, list(entrez_ids_microarray,
                     entrez_ids_infinum,
                     entrez_ids_RNAseq))
common_genes_MvI <- intersect(entrez_ids_microarray, entrez_ids_infinum)
common_genes_RvI <- intersect(entrez_ids_RNAseq, entrez_ids_infinum)

# Gene symbol of common genes in three analysis
sig_gene_symbol <- AnnotationDbi::select(org.Hs.eg.db,
                                  common_genes_MvRvI,
                                  "SYMBOL",
                                  "ENTREZID")

## 'select()' returned 1:1 mapping between keys and columns

 sig_gene_symbol$SYMBOL

##   [1] "RTKN2"     "GRK5"      "CLDN18"    "CCBE1"     "SASH1"
##   [6] "TNNC1"     "FAM107A"   "SVEP1"     "SLIT2"     "ACSS3"
##  [11] "ADRA1A"    "TOX3"      "FAM189A2"  "SH3GL3"    "AKAP12"
##  [16] "TGFBR3"    "TACC1"     "DNAH14"    "ACADL"     "CDO1"
##  [21] "ITGA8"      "GRIA1"      "LIMCH1"     "ITPRIP"     "LEPR"
##  [26] "GATA6"      "AHNAK"      "AMOTL1"     "PHACTR1"    "SOX17"
##  [31] "CP"         "EMP2"       "LTBP4"      "SEMA5A"     "HSPB6"
##  [36] "TTC28"      "SLIT3"      "ID3"        "SULF1"      "MYH10"
##  [41] "ADAMTS8"    "MCC"        "ADAM12"     "FXYD1"      "EBF1"
##  [46] "SPN"        "NET1"       "TNXB"       "KIF26B"     "ROR1"
##  [51] "TRAF4"      "AFF3"       "ZFP36L2"    "MGAT3"      "GALNT13"
##  [56] "DLC1"       "EFEMP1"     "ETV1"       "DES"        "KIF26A"
##  [61] "HBEGF"      "RAPGEF3"    "MAMDC2"     "HYAL1"      "NCKAP5"
##  [66] "BDNF"       "C14orf132"  "DLL1"       "CLDN11"     "SERINC2"
##  [71] "UBASH3B"    "SLC22A3"    "CDH3"       "AQP4"       "ST6GALNAC5"
##  [76] "CRIM1"      "AGAP11"     "ID4"        "DPP6"       "PHACTR2"
##  [81] "FBLN5"      "CLU"        "CYBRD1"     "PTGER4"     "TNFRSF21"
##  [86] "LIFR"       "LATS2"      "NFIA"       "FRAS1"      "NEDD9"
##  [91] "MBP"        "EYA4"       "DCN"        "PGM2L1"     "BDH1"
##  [96] "PTPRN2"     "GAB2"       "CADM1"      "PPP1R15A"   "C11orf80"
## [101] "SALL4"      "CLDN3"      "NAALAD2"    "LAMP3"      "COBL"
## [106] "THBD"       "AOX1"       "SOCS2"      "SNX25"      "LYPD1"
## [111] "CD59"       "NHSL1"
```

2.2 Top 4 genes in three analyses

```
sigsigMA <- DEgenes_microarray[which(DEgenes_microarray$entrez_id %in% si
g_gene_symbol$ENTREZID), ]
sigsigRNA <- DEgenes_RNAseq[which(DEgenes_RNAseq$entrezIDs %in% sig_gene_s
ymbol$ENTREZID), ]
sigsigInf <- DEgenes_methylation[which(DEgenes_methylation$EntrezIDs %in%
sig_gene_symbol$ENTREZID), ]

sigsigMA <- sigsigMA[order(sigsigMA$adj.P.Val),]
sigsigRNA <- sigsigRNA[order(sigsigRNA$FDR),]
sigsigInf <- sigsigInf[order(sigsigInf$adj.P.Val),]

topgenes <- 30
sigsig_MvRvI <- Reduce(intersect, list(head(sigsigMA$entrez_id, topgenes),
                     head(sigsigRNA$entrezIDs, topgenes),
                     head(sigsigInf$EntrezIDs, topgenes)))
mapIds(org.Hs.eg.db, sigsig_MvRvI, "SYMBOL", "ENTREZID")

## 'select()' returned 1:1 mapping between keys and columns

##      7049      3953    221692     64321
##  "TGFBR3"    "LEPR" "PHACTR1"   "SOX17"
```

2.3 Visualization

```
grid.newpage()
vennplot <- draw.triple.venn(area1 = length(entrez_ids_microarray),
               area2 = length(entrez_ids_RNAseq),
               area3 = length(entrez_ids_infinum),
               n12 = length(common_genes_MvR),
               n23 = length(common_genes_RvI),
               n13 = length(common_genes_MvI),
               n123 = length(common_genes_MvRvI),
               catoegory = c("DE genes from microarray",
                           "DE genes from RNAseq",
                           "DE genes from Infinum"),
               fill = c("light blue", "pink", "light yellow"),
               )
grid.draw(vennplot)
```



2.4 GSA for common genes in 3 techniques

```
  library(org.Hs.eg.db)
library(AnnotationDbi)
library(edgeR)

## Loading required package: limma

##
## Attaching package: 'limma'

## The following object is masked from 'package:BiocGenerics':
##
##     plotMA

 goana_out <- goana(de=common_genes_MvRvI, species="Hs", trend=T)

goana_out <- goana_out[order(goana_out$P.DE, decreasing=FALSE),]
goana_out$FDR.DE <- p.adjust(goana_out$P.DE, method="BH")
topGOcpg <- topGO(goana_out, ontology="BP", number=Inf)
kable(head(topGOcpg, 10))
```

	Term	Ont	N	DE	P.DE	FDR.DE
GO:0009653	anatomical structure morphogenesis	BP	2746	41	0	3.1e-06
GO:0048731	system development	BP	4345	53	0	3.1e-06
GO:0051239	regulation of multicellular organismal process	BP	2767	41	0	3.1e-06
GO:0007275	multicellular organism development	BP	4804	56	0	3.1e-06
GO:0048856	anatomical structure development	BP	5785	62	0	3.9e-06
GO:0040007	growth	BP	947	23	0	3.9e-06
GO:0032501	multicellular organismal process	BP	7480	72	0	3.9e-06
GO:0032502	developmental process	BP	6355	65	0	5.1e-06
GO:0042221	response to chemical	BP	4410	52	0	5.9e-06
GO:0032879	regulation of localization	BP	2808	40	0	5.9e-06