RNAseq data analysis를 통한 DEG추출과 분석

활동 캠퍼스 :UST KRIBB school
지도교수 성명 : 김미랑
인턴 : 최지인

수행 과제의 목적

- 최근 NGS의 발달로 인한 다량의 유전체 데이터 분석의 필요성이 대두되고 있으며 난치병에 대한 원인 유전자는 상당히 다양하기 때문에 유전자 발현의 분석과 유전적 변이의 검출을 통하여 각종 질병에 대한 맞춤유전자 치료법을 수행하는데 도움을 줄 수 있다.

- 이 중 RNA-sequencing을 통하여 차등 발현 유전자(Differentially Expressed Gene)분석을 진행할 수 있다. 이러한 차등 발현 유전자를 기반으로 gene enrichment분석, gene ontology분석, pathway분석을 통하여 질환의 발병기전을 이해하는데 도움이 될 수 있다.

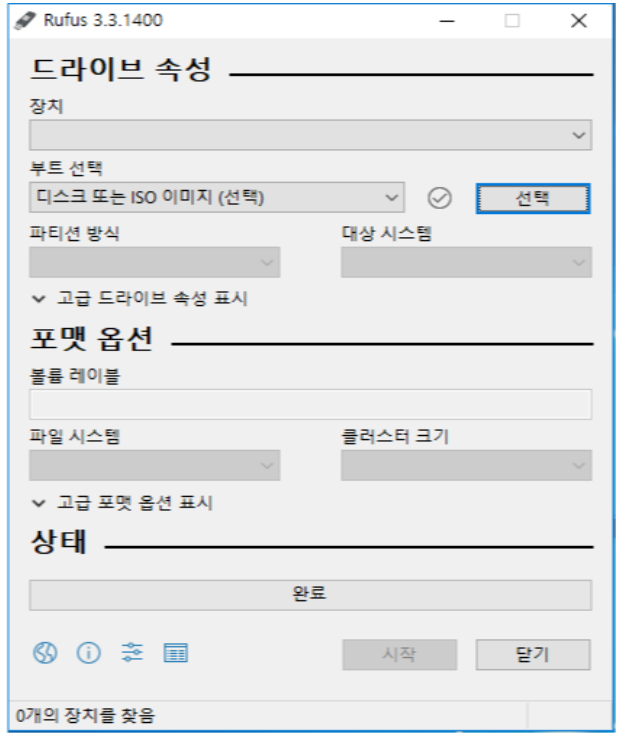
수행 과제의 기대효과

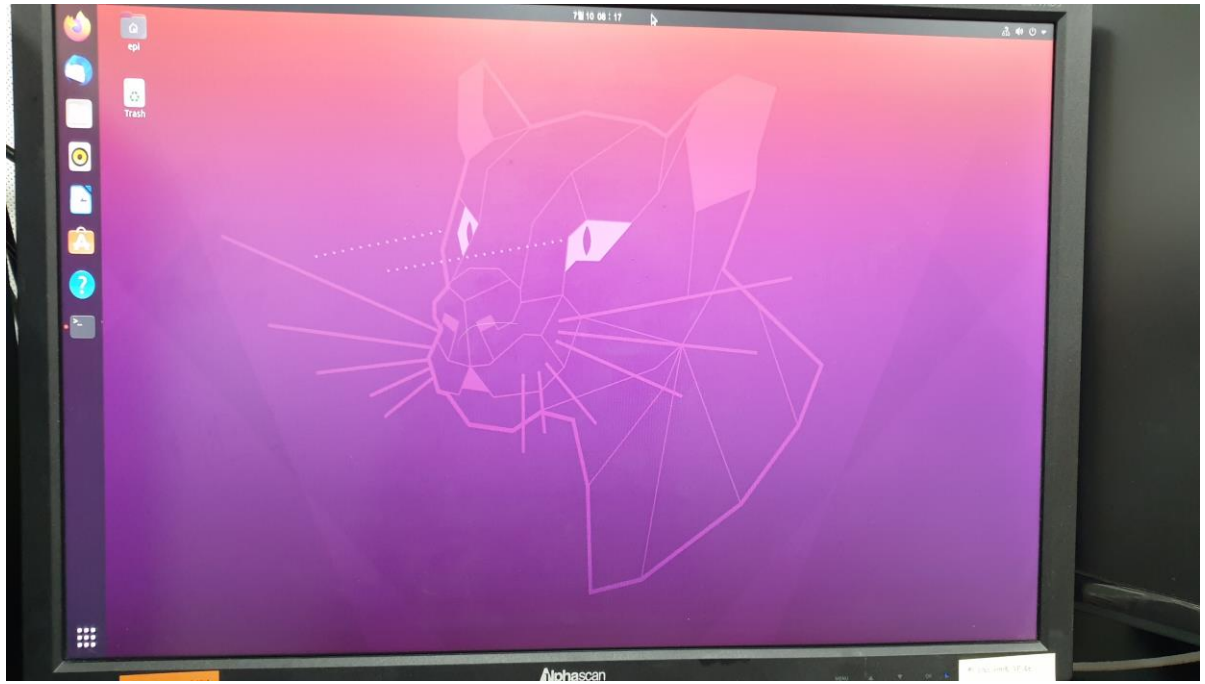
- RNAseq을 통한 NASH의 유전체 분석을 통해 해당 샘플에 대하여 expression이 많이 일어나는 유전자를 분석하여 NASH에 대한 유전적 정보와 DEG(Differentially Expressed Gene)을 알아낼 수 있게 되며 이에 따른 NASH에 대한 진단과 치료에 도움이 될 수 있도록 한다.

수행 경과

1. 분석을 위한 데스크톱 세팅 및 운영체제 설치(ubuntu linux 20.04.2 LTS)

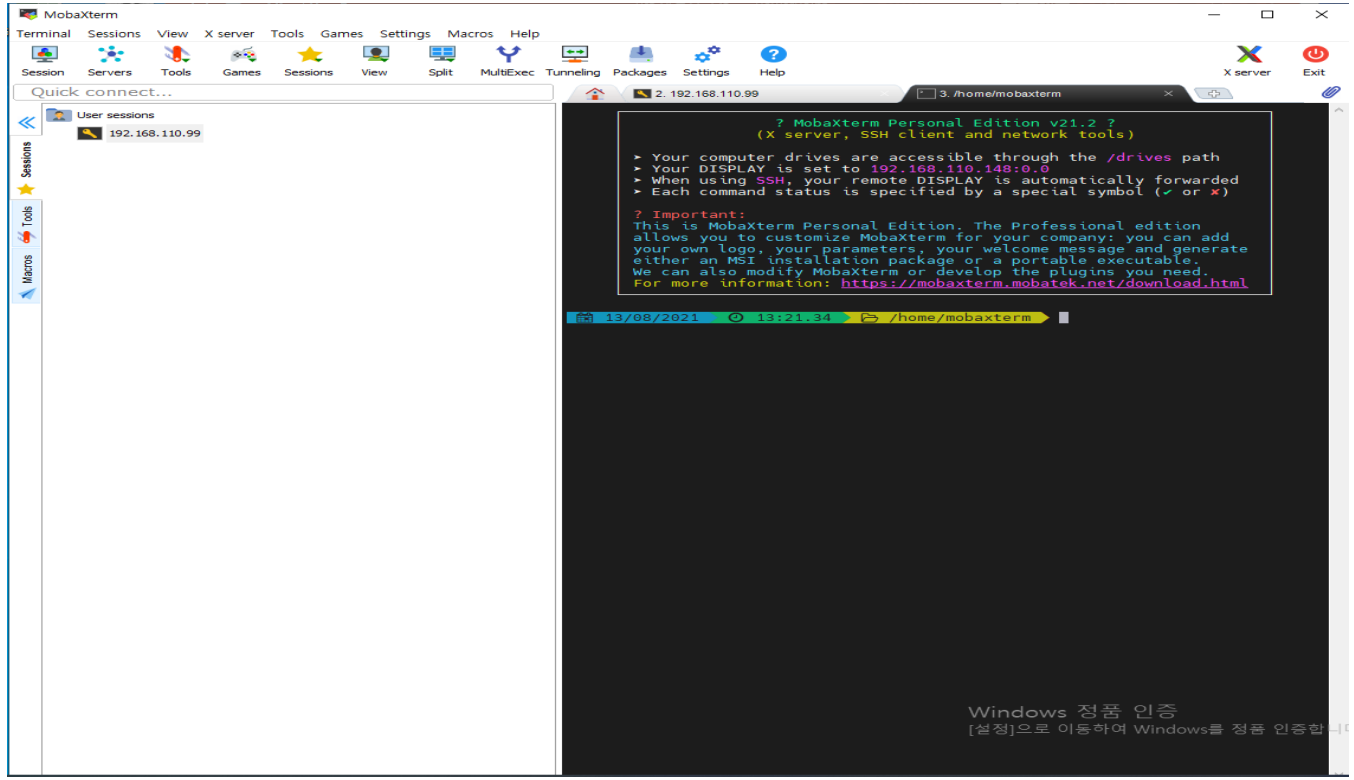
- RUFUS를 이용한 ubuntu linux 20.04.2 LTS설치용 usb를 만들고 해당 데스크톱에 설치한다.





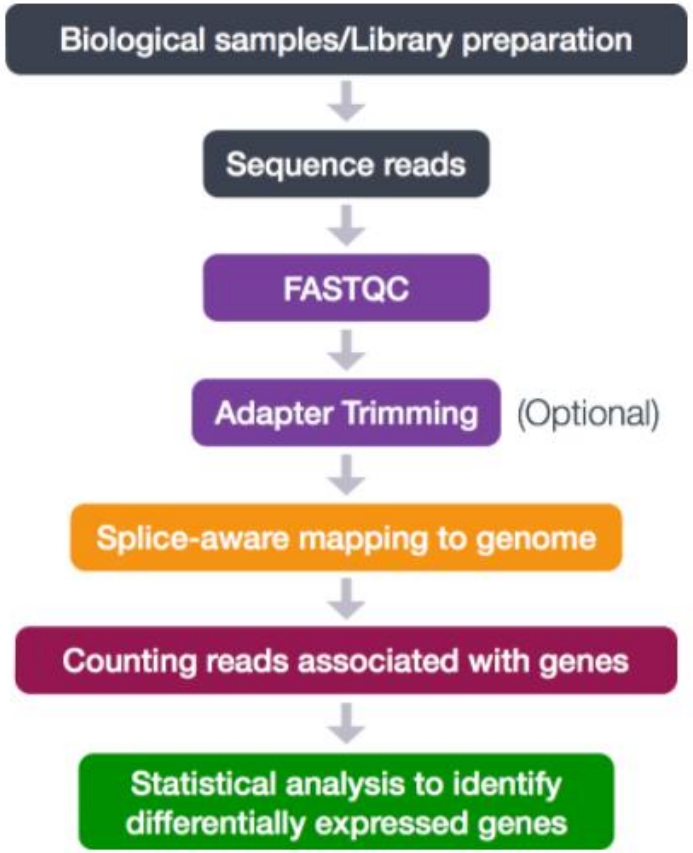
수행 경과

- SSH 프로토콜을 이용하여 윈도우 데스크톱에서 원격으로 연결하여 작업하기 위하여 Mobaxterm설치한다. Mobaxterm은 SSH툴 중 하나로 네트워크 상의 다른 컴퓨터에 로그인하거나 원격 시스템에서 명령을 실행하도록 도움을 주는 응용 프로그램이다.



수행 경과

2. RNAseq analysis workflow : 수행할 과제의 순서

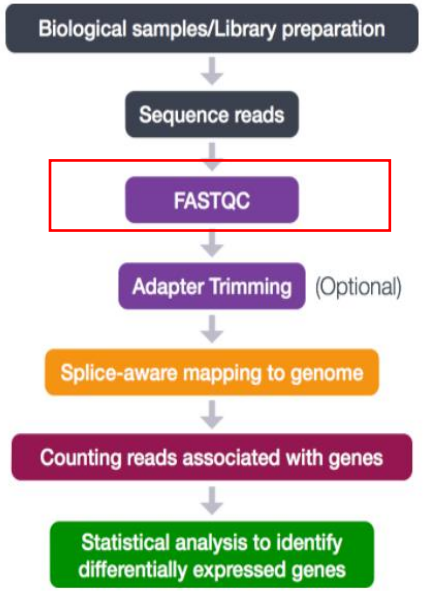



```
1XEPZ_4_1.fastq  1XEPZ_6_1.fastq  NASH_5_1.fastq  NCD_4_1.fastq  NCD_6_1.fastq
1XEPZ_4_2.fastq  1XEPZ_6_2.fastq  NASH_5_2.fastq  NCD_4_2.fastq  NCD_6_2.fastq
1XEPZ_5_1.fastq  NASH_4_1.fastq   NASH_6_1.fastq  NCD_5_1.fastq
1XEPZ_5_2.fastq  NASH_4_2.fastq   NASH_6_2.fastq  NCD_5_2.fastq
```

수행할 RNAseq analysis의 read files

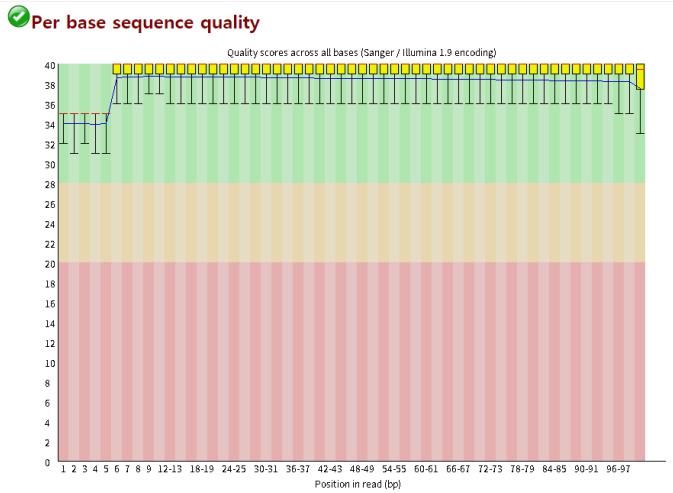
수행 경과

- FASTQC : raw data의 quality check단계로서 raw data가 좋게 보이는지, 다른 biases나 problems가 있는지 확인하기 위한 간단한 quality control check를 수행하는 단계이다. Raw data에서는 추정 오류 확률을 수치로 나타내며 phred score가 각 염기의 품질을 나타내는 지표로 활용되는데 이러한 phred score도 각 염기서열 별로 확인이 가능하다.



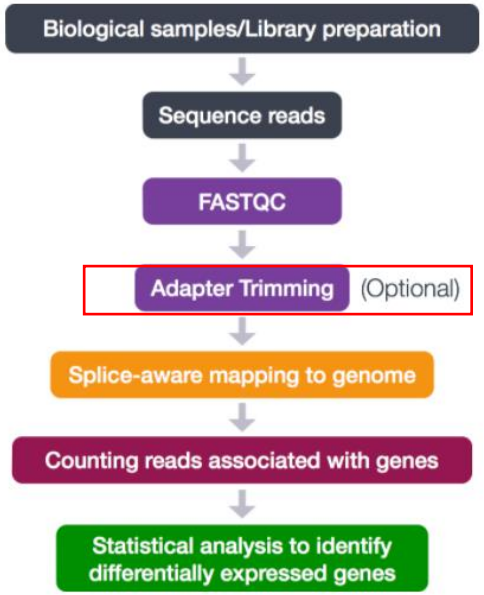

```
1XEPZ_4_1_fastqc.html  NASH_4_1_fastqc.html  NCD_4_1_fastqc.html
1XEPZ_4_2_fastqc.html  NASH_4_2_fastqc.html  NCD_4_2_fastqc.html
1XEPZ_5_1_fastqc.html  NASH_5_1_fastqc.html  NCD_5_1_fastqc.html
1XEPZ_5_2_fastqc.html  NASH_5_2_fastqc.html  NCD_5_2_fastqc.html
1XEPZ_6_1_fastqc.html  NASH_6_1_fastqc.html  NCD_6_1_fastqc.html
1XEPZ_6_2_fastqc.html  NASH_6_2_fastqc.html  NCD_6_2_fastqc.html
```

```
        echo "fastqc ./"${sampleNames[@]}" -o /data/" >> fastqcsub_script."${pha
seIdx}".sh
```



수행 경과

- Trimming (trim_galore 이용) : sequencing reads들의 adapter sequence 및 quality check과정에서의 낮은 phred quality score를 갖는 부분을 제거하는 과정이다.

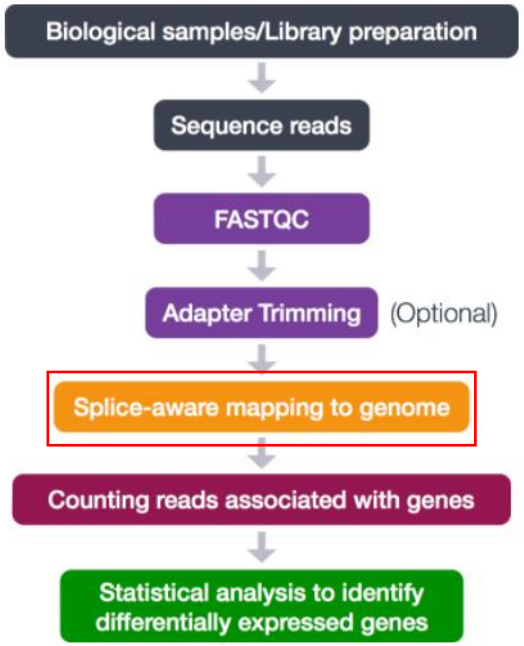



```
  echo "trim_galore --paired --fastqc -o /data/script/scripttest2 ./"${sampleName[@]}"1.fastq
./"${sampleName[@]}"2.fastq" >> trimgaloresub_script."${phaseIdx}".sh
```

```
1XEPZ_4_1.fastq_trimming_report.txt  NASH_5_1_val_1_fastqc.html
1XEPZ_4_1_val_1.fq                   NASH_5_1_val_1_fastqc.zip
1XEPZ_4_1_val_1_fastqc.html          NASH_5_2.fastq_trimming_report.txt
1XEPZ_4_1_val_1_fastqc.zip           NASH_5_2_val_2.fq
1XEPZ_4_2.fastq_trimming_report.txt  NASH_5_2_val_2_fastqc.html
1XEPZ_4_2_val_2.fq                   NASH_5_2_val_2_fastqc.zip
1XEPZ_4_2_val_2_fastqc.html          NASH_6_1.fastq_trimming_report.txt
1XEPZ_4_2_val_2_fastqc.zip           NASH_6_1_val_1.fq
1XEPZ_5_1.fastq_trimming_report.txt  NASH_6_1_val_1_fastqc.html
1XEPZ_5_1_val_1.fq                   NASH_6_1_val_1_fastqc.zip
1XEPZ_5_1_val_1_fastqc.html          NASH_6_2.fastq_trimming_report.txt
1XEPZ_5_1_val_1_fastqc.zip           NASH_6_2_val_2.fq
1XEPZ_5_2.fastq_trimming_report.txt  NASH_6_2_val_2_fastqc.html
1XEPZ_5_2_val_2.fq                   NASH_6_2_val_2_fastqc.zip
1XEPZ_5_2_val_2_fastqc.html          NCD_4_1.fastq_trimming_report.txt
1XEPZ_5_2_val_2_fastqc.zip           NCD_4_1_val_1.fq
1XEPZ_6_1.fastq_trimming_report.txt  NCD_4_1_val_1_fastqc.html
1XEPZ_6_1_val_1.fq                   NCD_4_1_val_1_fastqc.zip
1XEPZ_6_1_val_1_fastqc.html          NCD_4_2.fastq_trimming_report.txt
1XEPZ_6_1_val_1_fastqc.zip           NCD_4_2_val_2.fq
1XEPZ_6_2.fastq_trimming_report.txt  NCD_4_2_val_2_fastqc.html
1XEPZ_6_2_val_2.fq                   NCD_4_2_val_2_fastqc.zip
1XEPZ_6_2_val_2_fastqc.html          NCD_5_1.fastq_trimming_report.txt
1XEPZ_6_2_val_2_fastqc.zip           NCD_5_1_val_1.fq
MappedAligned.sortedByCoord.out.bam  NCD_5_1_val_1_fastqc.html
MappedLog.out                        NCD_5_1_val_1_fastqc.zip
MappedLog.progress.out               NCD_5_2.fastq_trimming_report.txt
Mapped_STARtmp                       NCD_5_2_val_2.fq
NASH_4_1.fastq_trimming_report.txt   NCD_5_2_val_2_fastqc.html
NASH_4_1_val_1.fq                    NCD_5_2_val_2_fastqc.zip
NASH_4_1_val_1_fastqc.html           NCD_6_1.fastq_trimming_report.txt
NASH_4_1_val_1_fastqc.zip            NCD_6_1_val_1.fq
NASH_4_2.fastq_trimming_report.txt   NCD_6_1_val_1_fastqc.html
NASH_4_2_val_2.fq                    NCD_6_1_val_1_fastqc.zip
NASH_4_2_val_2_fastqc.html           NCD_6_2.fastq_trimming_report.txt
NASH_4_2_val_2_fastqc.zip            NCD_6_2_val_2.fq
NASH_5_1.fastq_trimming_report.txt   NCD_6_2_val_2_fastqc.html
NASH_5_1_val_1.fq                    NCD_6_2_val_2_fastqc.zip
```

수행 경과

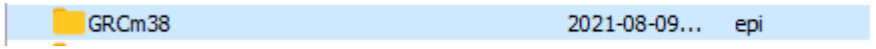
- Splice aware mapping to genome(STAR 사용) : trimming 과정을 마친 read들이 어떤 염색체 어느 위치에 있는 DNA인지에 대한 정보를 reference genome에서 위치를 찾아주는 작업이다.




```
 echo "STAR --runMode alignReads --runThreadN 2 --genomeDir /data/GRCm28/REFERENCE.STAR_idx/
--readFilesIn /data/useless2/"${sampleName[@]}"1_val_1.fq /data/useless2/"${sampleName[@]}
"2_val_2.fq --outSAMtype BAM SortedByCoordinate --outFileNamePrefix Mapped" >> STARsub_script.
"${phaseIdx}".sh
```

```
epi@epi-XPS-8700:/data$ STAR --runMode alignReads --runThreadN 16 --genomeDir /d
ata/GRCm38/REFERENCE.STAR_idx/ --readFilesIn /data/useless2/1XEPZ_4_1_val_1.fq /
data/useless2/1XEPZ_4_2_val_2.fq --outSAMtype BAM SortedByCoordinate --outFileNa
mePrefix Mapped
Aug 17 10:02:33 ..... started STAR run
Aug 17 10:02:33 ..... loading genome

EXITING: fatal error trying to allocate genome arrays, exception thrown: std::ba
d_alloc
Possible cause 1: not enough RAM. Check if you have enough RAM 28720419302 bytes
Possible cause 2: not enough virtual memory allowed with ulimit. SOLUTION: run u
limit -v 28720419302

Aug 17 10:02:33 ...... FATAL ERROR, exiting
```

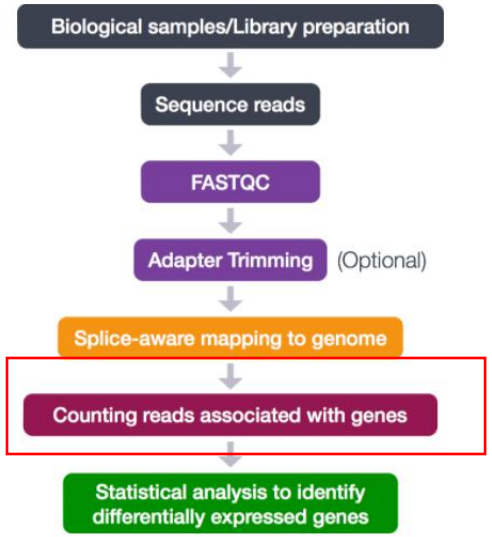
- Mapping의 과정 중 컴퓨터의 메모리 부족으로 인한 오류 발생



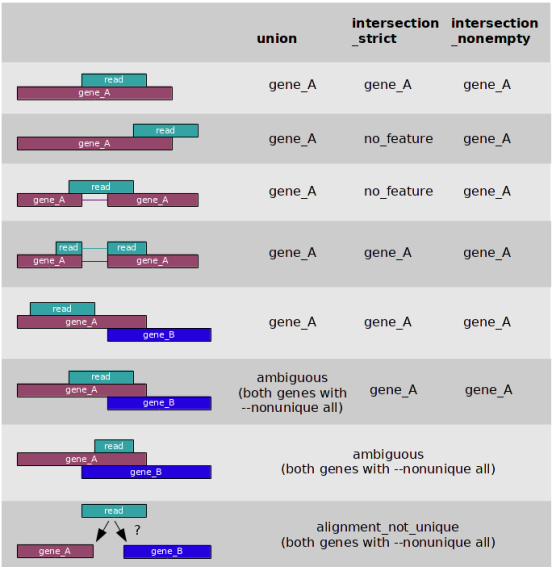
- Reference genome

수행 경과

- 수행 과정 중 불가피한 오류로 인한 이후 단계의 개념 공부 및 결론 도출 단계의 이론 공부 수행
- Counting reads associated with genes (htseq-count) : mapping이 완료된 파일을 gene annotation file을 이용하여 유전자의 발현 정도를 counting하는 단계이다.

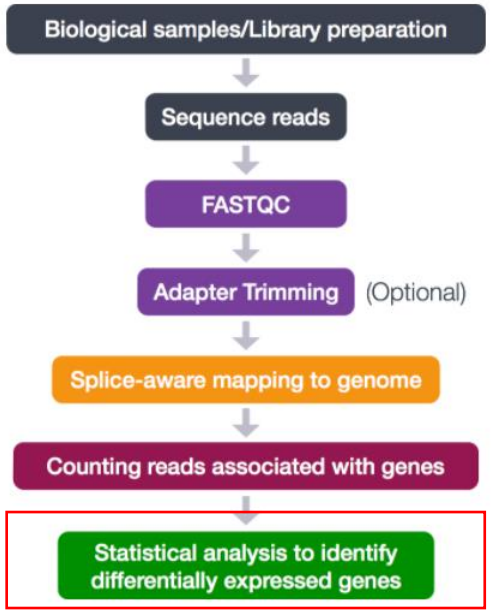



```
  echo "htseq-count -s reverse -m intersection-nonempty -f bam ./"${sampleName[@]}" ./"$
{sampleName[@]}" GCA_000001635.9_GRCm39_genomic.gff > ./"${sampleName[@]}"" >> htseqsub_script
."${phaseIdx}".sh
```



수행 경과

- Statistical analysis to identify differentially expressed genes : 이 후 데이터를 각 유전자의 발현량 (CPM/RPKM)으로 변환한다. counting된 데이터를 바탕으로 Normalization 및 Differentially expressed genes를 확인하여 이를 시각화하는 그래프로 나타낸다.(FDR값과 Fold change값을 고려)
- 이후의 과정은 인터넷의 example data를 통한 결론 도출을 진행하였다.



수행 결과

```
library(pathview)
library(gage)
library(gageData)

# working directory setup
setwd("C:/Users/CancerTeam/Desktop/rstudio연습")
res = read.delim("RNA_seq_example.txt", header = T, row.names = 1)
dim(res);View(res)

# Human KEGG pathway data
data(kegg.sets.hs)
str(kegg.sets.hs)

## set only signaling and metabolism
data(sigmet.idx.hs)
kegg.sets.hs = kegg.sets.hs[sigmet.idx.hs]
head(kegg.sets.hs,3)

## data organization
dim(res)
foldchanges = res$log2FoldChange

names(foldchanges)=rownames(res)

head(foldchanges)
head(kegg.sets.hs)

## get the results
keggres=gage(foldchanges,gsets=kegg.sets.hs,same.dir=T)
str(keggres)
view(keggres)
lapply(keggres,head)
```

- KEGG pathway code

```
library(enrichR)
listEnrichrSites()
setEnrichrSite("Enrichr") # Human genes
websiteLive <- TRUE

# find the list of all available databases from enrichr
dbs <- listEnrichrDbs()
if (is.null(dbs)) websiteLive <- FALSE
if (websiteLive) head(dbs)


if (is.null(dbs)) websiteLive <- FALSE
if (websiteLive) head(dbs)


# example : genes associated with embryonic haematopoiesis
dbs <- c("GO_Molecular_Function_2015", "GO_Cellular_Component_2015", "GO_Biological_Process_2015")
if (websiteLive) {
  enriched <- enrichr(c("Runx1", "Gfi1", "Gfi1b", "Spi1", "Gata1", "Kdr"), dbs)
}
### view the results table
if (websiteLive) enriched[["GO_Biological_Process_2015"]]

#### plot Enrichr GO-BP output
if (websiteLive) plotEnrich(enriched[[3]], showTerms = 20, numChar = 40, y = "Count", orderBy = "P.va
```

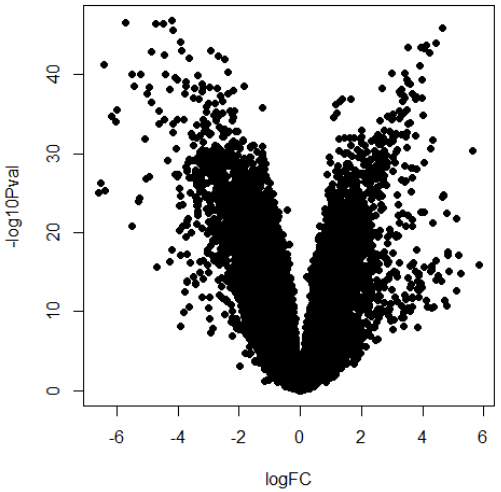
- Gene ontology analysis code

```
## Load the data
setwd("C:/Users/CancerTeam/Desktop")
Data <- read.table(GSE18842_DEG.xlsx,header=T,skip=1)
## header=T, skip=1옆에 부분은 첫줄 스킵하겠다는거

# excel파일 읽기
library(readxl)
Data<- read_excel("C:/Users/CancerTeam/Desktop/GSE18842_DEG.xlsx")
head(Data)# 데이터를 잘 불렀는지 체크 (데이터의 앞부분과 뒷부분(?) 출력)


####### create a Volcano plot
volcanoData <- cbind(Data$logFC, -log10(Data$P.Value))
colnames(volcanoData) <- c("logFC", "-log10Pval")
head(volcanoData)

plot(volcanoData, pch=19) ## pch는 숫자에 따라 그래프에 나타나는 모양이 다름(ex.원, 박스 등)
```

- Volcano plot(GSE18842_DEG)code

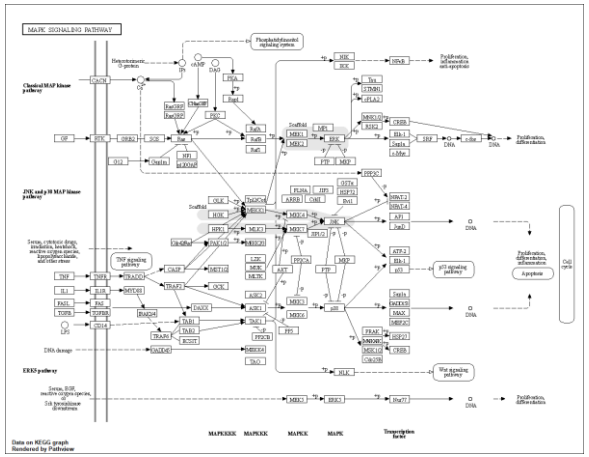
수행 결과

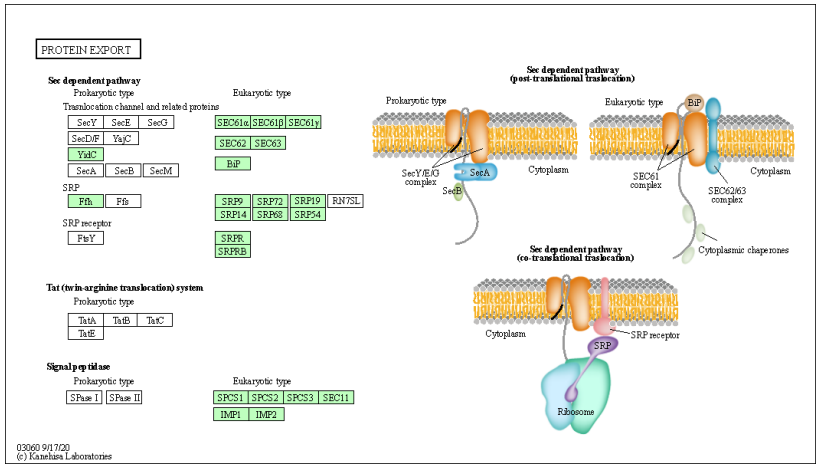
-



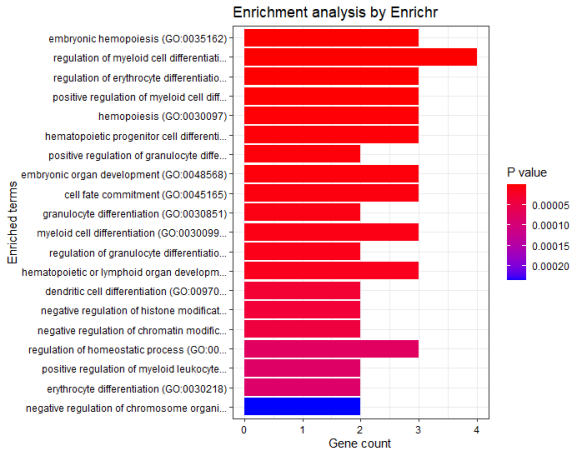
- GSE18842_DEG : FC값이 클수록 upregulation된 유전자이며 – logPval이 클수록 통계적으로 유의한 유전자들이다.

(volcano plot)





- Pathview analysis



- Gene ontology analysis(enrichR)