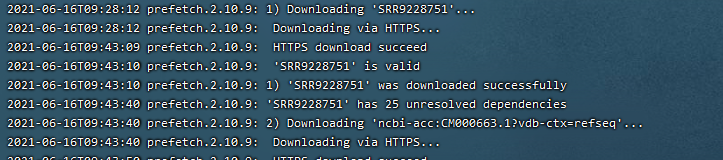
**###1.下载SRR数据通过SRAToolkit**

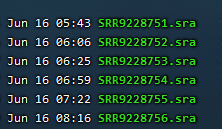
mkdir SRR\_all

cd ./SRR\_all

prefetch --option-file SRR\_Acc\_List .txt（SRR\_Acc\_List .txt包含需要下载的序列号）



共有6个SRA文件，2组不同的处理组各3个重复



**###2. SRA文件转换为fastq文件**

mkdir fastq

nohup bash rna.sh > srr2fastq.log 2>&1 &

sh文件代码（

file\_path=~/SRP200940/

indir=${file\_path}/SRR\_all

outdir=${file\_path}/fastq

for file in `ls $indir | grep .sra`;

do

a=${file%.sra\*};

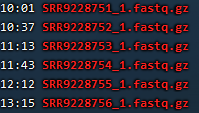
echo ----------

echo ${a}

~/software/SRAToolkit/sratoolkit.2.10.9-ubuntu64/bin/fastq-dump $indir/${a}.sra --gzip --split-files -O $outdir/

done

）



共六个fastq文件，为单端测序

**###3. Trimmomatic软件去除低质量read和接头**

mkdir clean\_fastq

nohup bash rna.sh > clean.log 2>&1 &

sh文件代码（

file\_path=~/SRP200940/

indir=${file\_path}/fastq

outdir=${file\_path}/clean\_fastq

for file in `ls $indir | grep 1.fastq.gz`;

do

a=${file%\_1.fastq.gz\*};

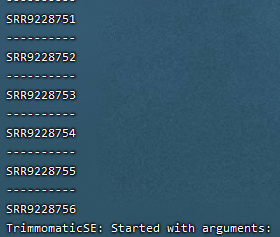
echo ----------

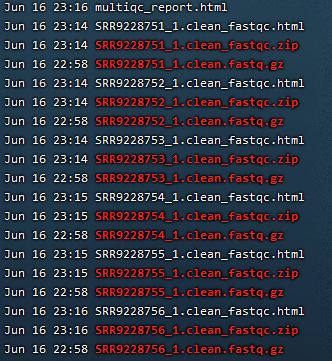
echo ${a};

java -jar ~/software/Trimmomatic-0.39/trimmomatic-0.39.jar SE -phred33 $indir/${a}\_1.fastq.gz $outdir/${a}\_1.clean.fastq.gz ILLUMINACLIP:~/software/Trimmomatic-0.39/adapters/TruSeq2-SE.fa:2:30:10 SLIDINGWINDOW:5:20 LEADING:20 TRAILING:20 MINLEN:75 &

done

）





得到clean fastq 文件

**###4. fastqc评估reads 的质量**

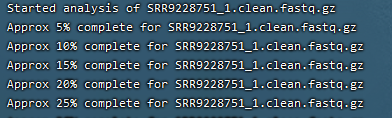
nohup bash rna.sh > fastq.log 2>&1 &

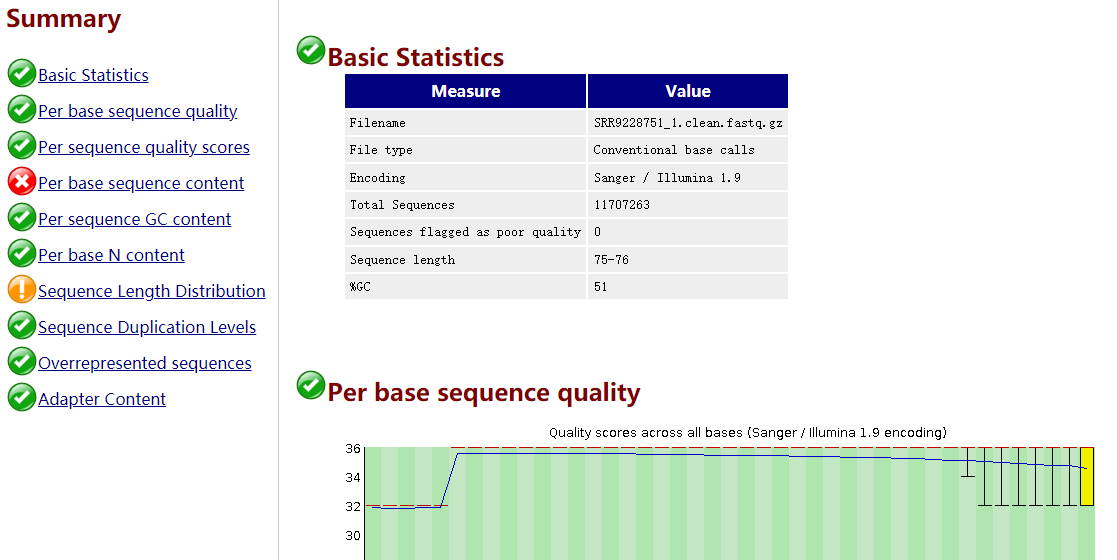
sh文件代码（

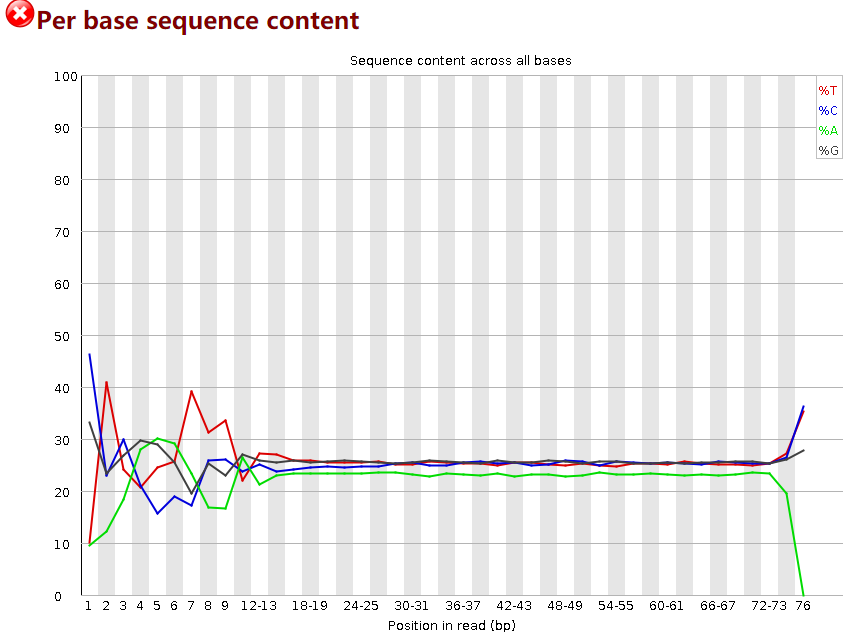
fastqc=~/software/FastQC/fastqc

cd ./clean\_fastq

$fastqc \*fastq.gz







可以看出该实验测序质量还是不错的，重复reads也不多。其中Per base sequence content（碱基含量）不好，但由于实际建库时，特别对于RNA\_seq文库时，在初始阶段，会存在碱基分布的偏倚，这个是正常现象，并不会影响下游的数据分析

**###5. 测序数据比对到参考基因组同时sam文件转换为bam文件**

mkdir bam

nohup bash rna.sh > hista2.log 2>&1 &

sh文件代码（

file\_path=~/SRP200940/

hisat=~/Genomic/hisat2-2.2.1/hisat2

GRCh38=~/ref\_genome/hisat2\_grch38\_tran/grch38\_tran/genome\_tran

outdir=${file\_path}/bam

indir=${file\_path}/clean\_fastq

for file in `ls $indir | grep 1.clean.fastq.gz`;

do

a=${file%\_1.clean.fastq.gz\*};

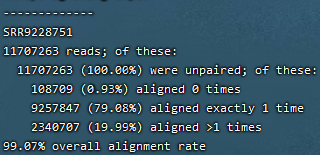
echo -------------;

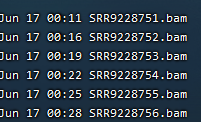
echo ${a};

$hisat -p 60 --dta -x $GRCh38 -U $indir/${a}\_1.clean.fastq.gz | samtools view -Sbh - > $outdir/${a}.bam

done

）





得到六个bam文件

**###6. 对 bam 文件中的内容进行排序**

mkdir sorted\_bam

nohup bash rna.sh > sorted\_bam.log 2>&1 &

sh文件代码（

file\_path=~/SRP200940/

indir=${file\_path}/bam

outdir=${file\_path}/sorted\_bam

for file in `ls $indir | grep .bam`;

do

a=${file%.bam\*};

echo ----------

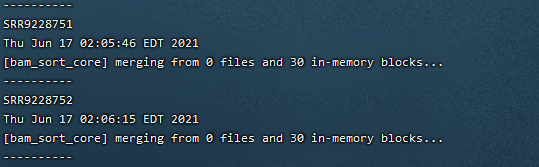
echo ${a}

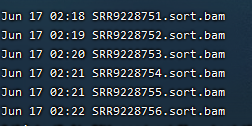
date

samtools sort -@ 30 -o $outdir/${a}.sort.bam $indir/${a}.bam

done

）





得到六个排序后的bam文件

**###7. 利用StringTie进行转录本组装，和量化基因表达**

mkdir stringtie\_gtf

mkdir ballgown

nohup bash rna.sh > StringTie.log 2>&1 &

sh文件代码（

file\_path=~/SRP200940/

stringtie=~/Genomic/stringtie-2.1.4.Linux\_x86\_64/stringtie

GRCh38\_gtf=~/ref\_genome/refdata-gex-GRCh38-2020-A/genes/genes.gtf

outdir=${file\_path}/stringtie\_gtf

indir=${file\_path}/sorted\_bam

ballgown\_dir=${file\_path}/ballgown

##对样本进行组装

for file in `ls $indir | grep .sort.bam`;

do

a=${file%.sort.bam\*};

echo -------------;

echo ${a};

date

$stringtie -p 50 -G $GRCh38\_gtf -o $outdir/${a}.gtf $indir/${a}.sort.bam

Done

##将所有转录本合并

echo ----------

echo "merge gtf files"

date

cd $outdir

$stringtie --merge -p 50 -G $GRCh38\_gtf -o $file\_path/merged.gtf `ls $outdir | grep .gtf`

echo "merge done"

##重新组装转录本并估算基因表达丰度

echo ----------

echo "counting transcript"

for file in `ls $indir | grep .sort.bam`;

do

a=${file%.sort.bam\*};

echo -------------;

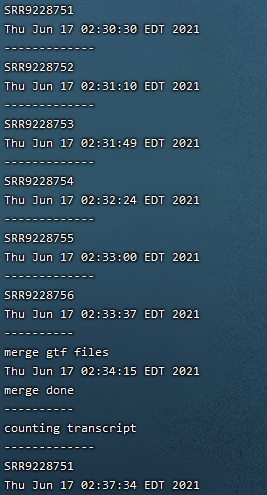
echo ${a};

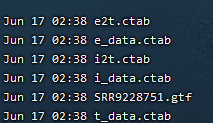
date

$stringtie -e -B -p 50 -G $file\_path/merged.gtf -o $ballgown\_dir/${a}/${a}.gtf $indir/${a}.sort.bam

done

）



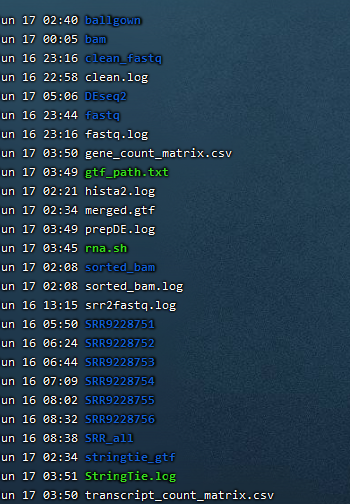


得到**记录组装的转录本信息的GTF文件**

**###8. stringtie输出的结果为ballgown所需要的格式，需要转换为deseq2需要的表格**

python2 ~/Genomic/stringtie-2.1.4.Linux\_x86\_64/prepDE.py

-i gtf\_path.txt -g gene\_count\_matrix.csv -t transcript\_count\_matrix.csv -l 150



生成两个csv文件

gene\_count\_matrix.csv

transcript\_count\_matrix.csv

**###9. DESeq2分析**

library(DESeq2)

#导入数据

CountMatrix1<-read.csv("~/SRP200940/gene\_count\_matrix.csv",sep=",",row.names="gene\_id")

##修改列名

names(CountMatrix1)<-c("ctrlrep1","ctrlrep2","ctrlrep3"," ISrep1","ISrep2","ISrep3")

#设置样本信息矩阵，包括处理信息：实验组ctrlrep vs. 对照组ISrep，每个有3个

ColumnData<- data.frame(row.names=colnames(CountMatrix1),samName=colnames(CountMatrix1), condition=rep(c("ctrlrep","ISrep"),each=3))

#生成DESeqDataSet数据集

dds<-DESeqDataSetFromMatrix(countData = CountMatrix1, colData = ColumnData, design = ~ condition)

#DESeq差异表达计算

dds<-DESeq(dds)

#生成差异表达结果

res<-results(dds)

summary(res)

#查看总结信息（表达上调，下调等）

head(res)

#统计padj（adjusted p-value）小于0.05的数目

table(res$padj <0.05)

#统计padj（adjusted p-value）小于0.05的数目

res<- res[order(res$padj),]#按padj排序

write.table(res, file = "~/SRP200940/DEseq2/diff\_express\_gene.txt")

filter\_up <- subset(res, pvalue < 0.05 & log2FoldChange > 1) #过滤上调基因

filter\_down <- subset(res, pvalue < 0.05 & log2FoldChange < -1) #过滤下调基因

print(paste('差异上调基因数量: ', nrow(filter\_up))) #打印上调基因数量

print(paste('差异下调基因数量: ', nrow(filter\_down))) #打印下调基因数量

write.table(filter\_up, file="~/SRP200940/DEseq2/filter\_up\_gene.txt", quote = F) #保存文件

write.table(filter\_down, file="~/SRP200940/DEseq2/filter\_down\_gene.txt", quote = F)

df = read.table("~/SRP200940/DEseq2/diff\_express\_gene.txt",header =T,stringsAsFactor = F) #读取差异表达基因数据

df = na.omit(df) ##绘图需要的padj列有NA值，故删掉包含NA的行

df$group = ifelse(df$log2FoldChange>=0.5 & df$padj<=0.05,"Up", ifelse(df$log2FoldChange<=-0.5 & df$padj<=0.05,"Down","Not sig")) ##设置上下调基因限定值

table(df$group) #如果数量太多可以适当修正上面logfoldchange值

ggplot(df,aes(x=log2FoldChange,y = -log10(padj)))+

geom\_point(aes(color=group))+ #绘制散点

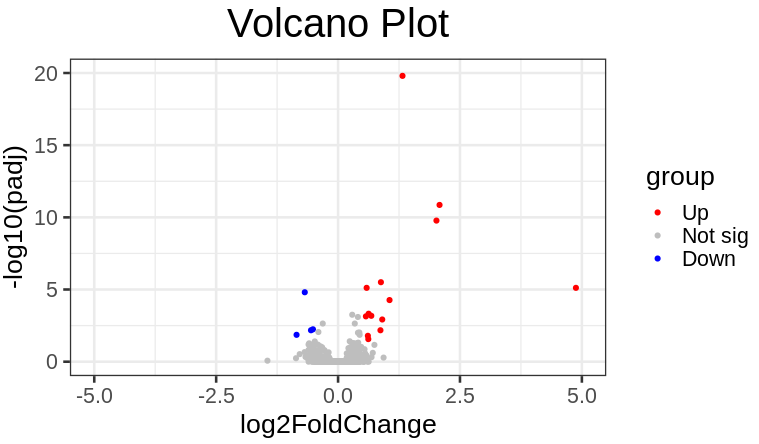
scale\_color\_manual(values = c("red","grey","blue"),limit = c("Up","Not sig","Down"))+ #定义点的颜色

theme\_bw(base\_size = 20)+ggtitle("Volcano Plot")+

theme(plot.title = element\_text(size=30,hjust = 0.5))+

coord\_cartesian(xlim = c(-5,5),ylim = c(0,20))

结果如下



##PCA

rld1<- rlog(dds)

data1 <- plotPCA(rld1, intgroup=c("condition","samName"), returnData=TRUE)

percentVar1 <- round(100 \* attr(data1, "percentVar"))

ggplot(data1, aes(PC1, PC2, color=condition)) +

geom\_text(aes(label = rownames(data1)))+

geom\_point(size=3) +

stat\_ellipse()+

xlab(paste0("PC1: ",percentVar1[1],"% variance")) +ylab(paste0("PC2: ",percentVar1[2],"% variance"))

结果如下,不同处理组还是各自分开的

