DEGs分析pipeline：FeatureCounts + DESeq2 + 过滤数据 + 去overlapping

作者：李凯旋，卢建璋，徐舟通（以上名字按姓氏拼音字母排序）

一、什么是DEGs分析？为什么要进行DEGs分析？

1. DEGs analysis，即 Differential Expression Genes，意为差异化基因表达分析。差异化基因表达分析有助于我们了解不同样本的基因表达差异，进而了解不同组织或者不同细胞的状态与功能差异。

2. 本文将使用非小细胞肺癌细胞系 A549 与人类正常肺组织样本对比为例，展示DEGs分析的具体流程。

二、FeatureCounts

- *reference: “featureCounts: an efficient general purpose program for assigning sequence reads to genomic features”*

1. 在RNA-seq分析得到比对文件 .bam (确保已经过排序和去重)后，我们使用featureCounts软件来统计样本文件中有多少reads匹配到了我们所感兴趣(features)的区域，比如exon区域

2. 代码

Bash

```bash
# 环境准备
source ~/.bashrc
conda activate twobittofa

# featureCounts格式：featureCounts [options] <input.file>
# 实例：
featureCounts \
-T 8
-p \
-t exon \
-g exon_id \
-a /public/workspace/jianl/Reference/Human/annotation/gencode.v19.annotation.gtf \
-o featurecounts_result_exonid_A549.txt \
# inputfile 有几个写几个，均为 .bam 文件，⚠注意：文件输入顺序务必为先normal后cancer 的顺序，以免影响后续Log2FoldChange的计算
ENCFF155XBJ_dedu.sort.bam \
ENCFF150VSG_dedu.sort.bam \
ENCLB555AUL_dedu.sort.bam \
ENCLB555AUK_dedu.sort.bam

```

3. 参数解释：

1. input file: Input bam/Sam files, supporting multiple file inputs(spreated by /)
2. -a < string >: Refer to GTF file name, support Gzipped file format **GTF_path: /public/workspace/jianl/Reference/Human/annotation/gencode.v19.annotation.gtf**
3. -o < string >: The name of the output file that contains the number of read statistics
4. -T: The value ranges from 1 to 32
5. -p: Used only in paired-end cases, the fragment is counted but the read is not
6. -f: If -f is set, feature level data such as exon-level will be counted, otherwise meta-feature level data such as gene-levels will be counted. **Through the comparison of -f parameters in this DEG analysis, it is found that the parameter most consistent with expectations is “exonid”**

- 为了避免占用运行内存过大导致进程被kill的情况，也可以使用bsub将程序交给服务器后台运行

Bash

```
#!/bin/sh
#PBS -l nodes=1:ppn=16
#PBS -l walltime=24:00:00
#PBS -N featurecounts_A549
#PBS -d /public/workspace/jianl/User/Zhoutong-Xu/zhoutong/A549_DEG/
#PBS -o featurecounts.out
#PBS -e featurecounts.err
source ~/.bashrc
conda activate twobittofa
featureCounts -T 8 -p -t exon -g exon_id -a /public/workspace/jianl/Reference/Human/annotation/gencode.v19.annotation.gtf \
-o featurecounts_result_exonid_A549.txt \
ENCFF155XBJ_dedu.sort.bam \
ENCFF150VSG_dedu.sort.bam \
ENCLB555AUL_dedu.sort.bam \
ENCLB555AUK_dedu.sort.bam
```

三、DESeq2

1. 文件准备：

上一步得到 featurecounts_result_exonid_A549.txt，查看文件会发现某些基因在一条染色体上的多个位置表达，或者在多条染色体上的不同位置表达（例如DNA表达控制基因Y_RNA）（图1）。由于这些基因有着实际的生物学意义，而我们最后需要的bed文件格式要求每一行只能有一个chr序号，一个chr开始位置，一个chr结束位置，所以我们需要把这些行分成多行。

ENSE00001718035.1	chr1;chr1	29321;29321	29370;29370	-;-	50	0	0	0				
ENSE00003624050.1	chr1	24738	24891	-	154	0	0	0	0			
ENSE00001642865.1	chr1	18268	18379	-	112	0	0	0	0			
ENSE00003638984.1	chr1;chr1;chr1	17915;17915;17915	18061;18061;18061	-;-;-	147	0	0	0	0			
ENSE00001699689.1	chr1;chr1	17602;17602	17742;17742	-;-	141	0	0	0	0			
ENSE00001656010.1	chr1;chr1;chr1	17233;17233;17233	17364;17364;17364	-;-;-	132	0	0	0	0			
ENSE00001760358.1	chr1;chr1	16854;16854	17055;17055	-;-	202	0	0	0	0			
ENSE00003618297.1	chr1	16607	16765	-	159	0	0	0	0			
ENSE00001375216.1	chr1;chr1	15904;15904	15947;15947	-;-	44	0	0	0	0			
ENSE00001388009.1	chr1;chr1	15796;15796	15901;15901	-;-	106	0	0	0	0			

图1: featurecounts_result_exonid_A549.txt截图

使用 format.py脚本对文件进行处理，代码如下：

Python

```python
infile = open('input_file_name.txt')
outfile = open('output_file_name.txt', 'w')
outfile.write(infile.readline())
for lines in infile:
    line = lines.split('\t')
    chrs = line[1].split(';')
    samples = '\t'.join(line[5:])
    if(len(chrs) < 1):
        print(line)
    elif(len(chrs) > 1):
        starts = line[2].split(';')
        ends = line[3].split(';')
        strand = line[4].split(';')
        out = []
        for i in range(0, len(starts)):
            out.append(starts[i] + '\t' + ends[i] + '\t' + strand[i])
        out = list(set(out))
        if(len(out) != 1):
            print(line[0])
        for j in range(0, len(out)):
            outfile.write(line[0] + '\t' + chrs[j] + '\t' + out[j] +'\t' + sam
ples)
    else:
        outfile.write(lines)
```

运行方法:将"/public/workspace/jianl/Software/DEGtools/format.py"拷贝到自己的文件夹下，并修改输入文件名和输出文件名

```python
infile = open('featurecounts_result_exonid_A549.txt')
outfile = open('featurecounts_result_exonid_A549_out.txt', 'w')
```

保存后运行:

Bash

```bash
conda activate tests
python3 format.py
```

得到符合要求的文件:

Geneid	Chr	Start	End	Strand	Length	ENCFF155XBJ_dedu.sort.bam	ENCFF150VSG_dedu.sort.bam	ENCLB555AUL_dedu.sort.bam	ENCLB555AUK_dedu.sort.bam
ENSE00002234944.1	chr1	11869	12227	+	359	0	0	0	0
ENSE00003582793.1	chr1	12613	12721	+	109	0	0	0	0
ENSE00002312635.1	chr1	13221	14409	+	1189	0	0	0	0
ENSE00002234632.1	chr1	11872	12227	+	356	0	0	0	0
ENSE00003608237.1	chr1	12613	12721	+	109	0	0	0	0
ENSE00002306041.1	chr1	13225	14412	+	1188	0	0	0	0
ENSE00002269724.1	chr1	11874	12227	+	354	0	0	0	0
ENSE00002270865.1	chr1	12595	12721	+	127	0	0	0	0
ENSE00002216795.1	chr1	13403	13655	+	253	0	0	0	0
ENSE00002303382.1	chr1	13661	14409	+	749	0	0	0	0
ENSE00001948541.1	chr1	12010	12057	+	48	0	0	0	0
ENSE00001671638.2	chr1	12179	12227	+	49	0	0	0	0
ENSE00001758273.2	chr1	12613	12697	+	85	0	0	0	0
ENSE00001799933.2	chr1	12975	13052	+	78	2	0	0	0
ENSE00001746346.2	chr1	13221	13374	+	154	0	0	0	0
ENSE00001863096.1	chr1	13453	13670	+	218	0	0	0	0
ENSE00001718035.1	chr1	29321	29370	-	50	0	0	0	0
ENSE00003624050.1	chr1	24738	24891	-	154	0	0	0	0
ENSE00001642865.1	chr1	18268	18379	-	112	0	0	0	0
ENSE00003638984.1	chr1	17915	18061	-	147	0	0	0	0
ENSE00001699689.1	chr1	17602	17742	-	141	0	0	0	0
ENSE00001656010.1	chr1	17233	17364	-	132	0	0	0	0
ENSE00001760358.1	chr1	16854	17055	-	202	0	0	0	0
ENSE00003618297.1	chr1	16607	16765	-	159	0	0	0	0
ENSE00001375216.1	chr1	15904	15947	-	44	0	0	0	0
ENSE00001388009.1	chr1	15796	15901	-	106	0	0	0	0
ENSE00003497546.1	chr1	14970	15038	-	69	0	0	0	0
ENSE00003511598.1	chr1	14363	14829	-	467	0	0	0	0
ENSE00002254515.1	chr1	24734	24886	-	153	0	0	0	0
ENSE00002303227.1	chr1	18268	18369	-	102	0	0	0	0
ENSE00003629019.1	chr1	17606	17742	-	137	0	0	0	0
ENSE00002285713.1	chr1	17498	17504	-	7	8	6	1	0
ENSE00003603734.1	chr1	24738	24891	-	154	0	0	0	0
ENSE00003513603.1	chr1	17915	18061	-	147	0	0	0	0
ENSE00003565315.1	chr1	17606	17742	-	137	0	0	0	0
ENSE00003685767.1	chr1	17233	17368	-	136	0	0	0	0
ENSE00003553898.1	chr1	16858	17055	-	198	0	0	0	0
ENSE00003621279.1	chr1	16607	16765	-	159	0	0	0	0
ENSE00002030414.1	chr1	15796	15947	-	152	0	0	2	3
ENSE00003591210.1	chr1	14970	15038	-	69	0	0	0	0
ENSE00003693168.1	chr1	14363	14829	-	467	0	0	0	0
ENSE00001890219.1	chr1	29534	29570	-	37	0	0	0	0
ENSE00003507205.1	chr1	24738	24891	-	154	0	0	0	0
ENSE00003477500.1	chr1	18268	18366	-	99	0	0	0	0
ENSE00003565697.1	chr1	17915	18061	-	147	0	0	0	0
ENSE00003475637.1	chr1	17606	17742	-	137	0	0	0	0
ENSE00003502542.1	chr1	17233	17368	-	136	0	0	0	0
ENSE00001935574.1	chr1	15005	15038	-	34	0	0	0	0
ENSE00001843071.1	chr1	14404	14501	-	98	0	0	0	0
ENSE00001378845.2	chr1	29534	29806	-	273	0	0	0	0

下载文件至本地

2. DESeq2 with R

- Reference: "*Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2*"

代码如下:

a. 读入数据:

R

```r
#载入DEseq2包
library(DESeq2)
#读入数据，按"\t"取数据  ⚠注意修改你自己的文件路径
data <- read.table("/Users/apple/LAB/A549_vs_Normal/featurecounts_result_exonid_A549_out.txt",header=TRUE,sep = "\t")
```

b. 数据整合与准备:

R

```R
#修改samplename
sampleNames <- c("ENCFF155XBJ","ENCFF150VSG","ENCLB555AUL","ENCLB555AUK")
names(data)[7:10] <- sampleNames
#提取数据构建表达矩阵
countData <- as.matrix(data[7:10])
#整合chr信息，防止丢失
data_1 <- unite(data, "idchr", c("Geneid","Chr","Start","End"), sep = "#", rem
ove = TRUE)
rownames(countData) <- data_1$idchr
#设置分组信息：确保样本与分组信息一一对应，比如"ENCFF155XBJ","ENCFF150VSG"对应于norma
l组，"ENCLB555AUL","ENCLB555AUK"对应于癌症组
database <- data.frame(name=sampleNames, condition=c("Normal","Normal","Tumor"
,"Tumor"))
rownames(database) <- sampleNames
```

c. 从表达矩阵countData和样品信息colData构建DESeqDataSet对象

R

```R
dds <- DESeqDataSetFromMatrix(countData, colData=database, design= ~ conditio
n)
dds <- dds[ rowSums(counts(dds)) > 1, ] #过滤低丰度数据
```

d. 差异化表达基因分析

R

```R
#差异化表达分析
dds <- DESeq(dds)
#生成res对象
res <- results(dds)
#查看结果，确保分组正确
head(res)
```

```
> head(res)
log2 fold change (MLE): condition Tumor vs Normal
Wald test p-value: condition Tumor vs Normal
DataFrame with 6 rows and 6 columns
```

e. 按照q-value的值进行分组

- Null (g0) : q-value >0.05

R

```r
write.csv(res, "/Users/apple/LAB/A549_vs_Normal/unilter.csv")
unfilter <- read.csv("/Users/apple/LAB/A549_vs_Normal/unfilter.csv")
unfilter <- separate(data = unfilter, "X", into = c("Geneid","Chr","Start","End"), sep = "#")
unfilter <- subset(unfilter, abs(log2FoldChange) > 0.585)

unfilter$Start <-  as.numeric(unfilter$Start)
unfilter$End <- as.numeric(unfilter$End)
g0_null_up <- arrange(unfilter[which(unfilter$padj > 0.05 & unfilter$log2FoldChange>0.585),], Chr, Start, End)
g0_up <- g0_null_up[,-c(1,5,7,8,9,10)]
g0_null_down <- arrange(unfilter[which(unfilter$padj > 0.05 & unfilter$log2FoldChange< -0.585),], Chr, Start, End)
g0_down <- g0_null_down[,-c(1,5,7,8,9,10)]
```

- Significant(g1-g5): q-value <0.05

R

```r
filter_res <- subset(res, padj < 0.05 & abs(log2FoldChange) > 0.585 )
write.csv(filter_res, "/Users/apple/LAB/A549_vs_Normal/filter.csv")
aa <- read.csv("/Users/apple/LAB/A549_vs_Normal/filter.csv")
#还原chr, start, end信息
DEGs_DESeq2 <- separate(data = aa, "X", into = c("Geneid","Chr","Start","End"), sep = "#")

#开始分组
n1 = 5e-2
n2 = 1e-2
n3 = 5e-3
n4 = 1e-3
n5 = 5e-4
#将start end 列的数据类型转化为数字，便于排序以及最后的去重叠。
DEGs_DESeq2$Start <-  as.numeric(DEGs_DESeq2$Start)
DEGs_DESeq2$End <- as.numeric(DEGs_DESeq2$End)
deg_up <- arrange(DEGs_DESeq2[which(DEGs_DESeq2$log2FoldChange > 0.585),], Chr,Start, End)
deg_up <- deg_up[,-c(1,5,7,8,9)]
deg_down <- arrange(DEGs_DESeq2[which(DEGs_DESeq2$log2FoldChange < -0.585),], Chr, Start, End)
deg_down <- deg_down[,-c(1,5,7,8,9)]

#246,183,198
g1_up <- arrange(deg_up[which(deg_up$padj <= n1 & deg_up$padj > n2),-c(5)], Ch
```

```
r, Start, End)
#191,203,249
g1_down <- arrange(deg_down[which(deg_down$padj <= n1 & deg_down$padj > n2),-c(5)], Chr, Start, End)

#244,106,139
g2_up <- arrange(deg_up[which(deg_up$padj <= n2 & deg_up$padj > n3),-c(5)], Chr, Start, End)

#109,139,248
g2_down <- arrange(deg_down[which(deg_down$padj <= n2 & deg_down$padj > n3),-c(5)], Chr, Start, End)

#245,8,65
g3_up <- arrange(deg_up[which(deg_up$padj <= n3 & deg_up$padj > n4),-c(5)], Chr, Start, End)
#6,57,249
g3_down <- arrange(deg_down[which(deg_down$padj <= n3 & deg_down$padj > n4),-c(5)], Chr, Start, End)

#161,4,42
g4_up <- arrange(deg_up[which(deg_up$padj <= n4 & deg_up$padj > n5),-c(5)], Chr, Start, End)
#4,37,161
g4_down <- arrange(deg_down[which(deg_down$padj <= n4 & deg_down$padj > n5),-c(5)], Chr, Start, End)

#95,3,25
g5_up <- arrange(deg_up[which(deg_up$padj <= n5),-c(5)], Chr, Start, End)
#2,21,91
g5_down <- arrange(deg_down[which(deg_down$padj <= n5),-c(5)], Chr, Start, End)
```

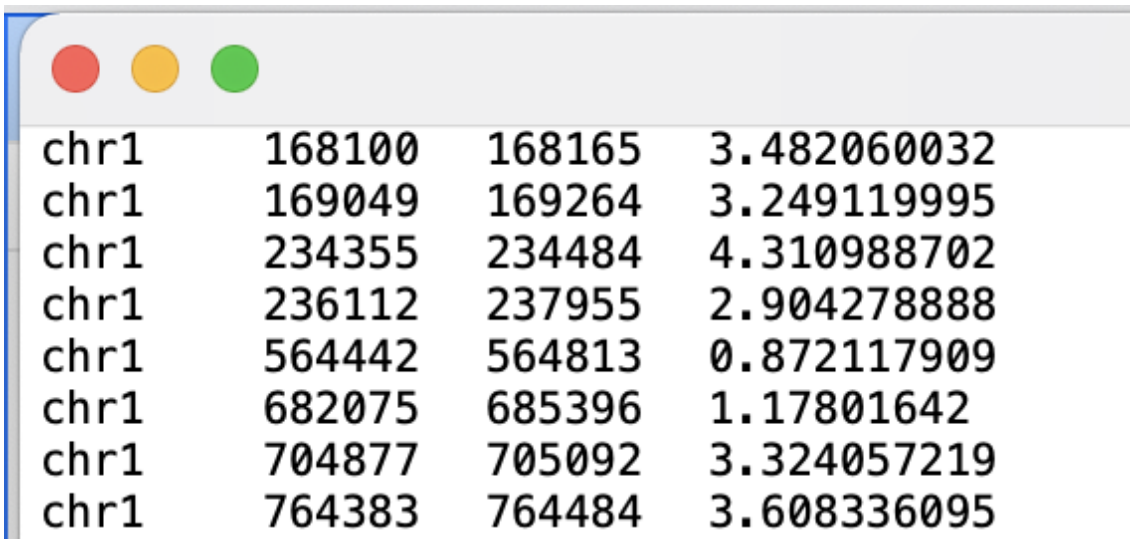
- 输出文件：

R

```R
setwd('/Users/apple/LAB/A549_vs_normal_unfilter/')

write.table(g0_up, file = "g0_up.txt", row.names = FALSE, col.names = FALSE, quote = FALSE)
write.table(g0_down, file = "g0_down.txt", row.names = FALSE, col.names = FALSE, quote = FALSE)
write.table(g1_up, file = "g1_up.txt", row.names = FALSE, col.names = FALSE, quote = FALSE)
write.table(g1_down, file = "g1_down.txt", row.names = FALSE, col.names = FALSE, quote = FALSE)
write.table(g2_up, file = "g2_up.txt", row.names = FALSE, col.names = FALSE, quote = FALSE)
write.table(g2_down, file = "g2_down.txt", row.names = FALSE, col.names = FALSE, quote = FALSE)
write.table(g3_up, file = "g3_up.txt", row.names = FALSE, col.names = FALSE, quote = FALSE)
write.table(g3_down, file = "g3_down.txt", row.names = FALSE, col.names = FALSE, quote = FALSE)
write.table(g4_up, file = "g4_up.txt", row.names = FALSE, col.names = FALSE, quote = FALSE)
write.table(g4_down, file = "g4_down.txt", row.names = FALSE, col.names = FALSE, quote = FALSE)
write.table(g5_up, file = "g5_up.txt", row.names = FALSE, col.names = FALSE, quote = FALSE)
write.table(g5_down, file = "g5_down.txt", row.names = FALSE, col.names = FALSE, quote = FALSE)
```

f. 去重叠区域

- **标准的bedgraph格式文件如下：**



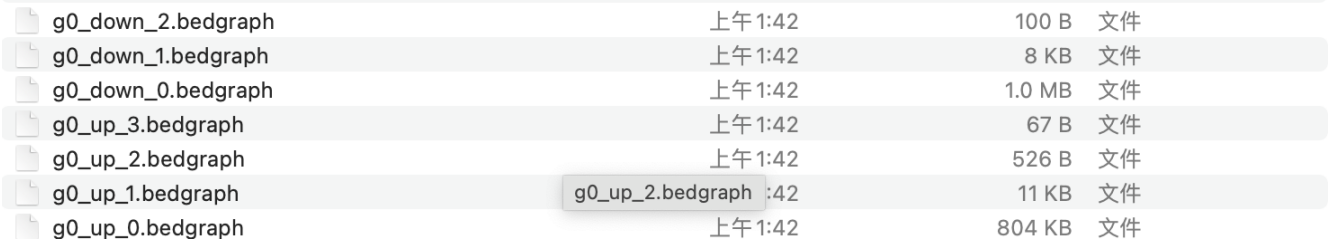
```
chr1    168100  168165  3.482060032
chr1    169049  169264  3.249119995
chr1    234355  234484  4.310988702
chr1    236112  237955  2.904278888
chr1    564442  564813  0.872117909
chr1    682075  685396  1.17801642
chr1    704877  705092  3.324057219
chr1    764383  764484  3.608336095
```

- 使用 de_overlap.py(路径：/public/workspace/jianl/Software/DEGtools/de_overlap.py)将文件中的重叠片段分为若干个文件

Python

```python
"""
made by Fang Anxuan & Shiyang Song & Zhoutong Xu
"""
with open("g5_down.txt") as f:
    all_geneData = f.readlines()
'''
print(all_geneData[0])
'''
res_list = [""]
previous_groupIndex = 1

previous_geneData = all_geneData[0]

res_list[0] += previous_geneData
p_end = previous_geneData.split(" ")[2]
p_chr = previous_geneData.split(" ")[0]

for geneData in all_geneData[1:]:
    chr = geneData.split(" ")[0]
    start = geneData.split(" ")[1]
    end = geneData.split(' ')[2]
    if p_chr == chr:
        if int(start) <= int(p_end):
            if(previous_groupIndex >= len(res_list)):
                res_list.append('')
            res_list[previous_groupIndex] += geneData
            previous_groupIndex += 1
            p_end = max(p_end, end)
        else:
            res_list[0] += geneData
            previous_groupIndex = 1
            p_end = geneData.split(" ")[2]
    else:
        res_list[0] += geneData
        previous_groupIndex = 1
        p_end = geneData.split(" ")[2]
        p_chr = geneData.split(" ")[0]

index = 0
for res in res_list:
    name = "g5_down_%d.bedgraph"
    with open(name%index,"w") as f:
        f.write(res)
    index += 1
```

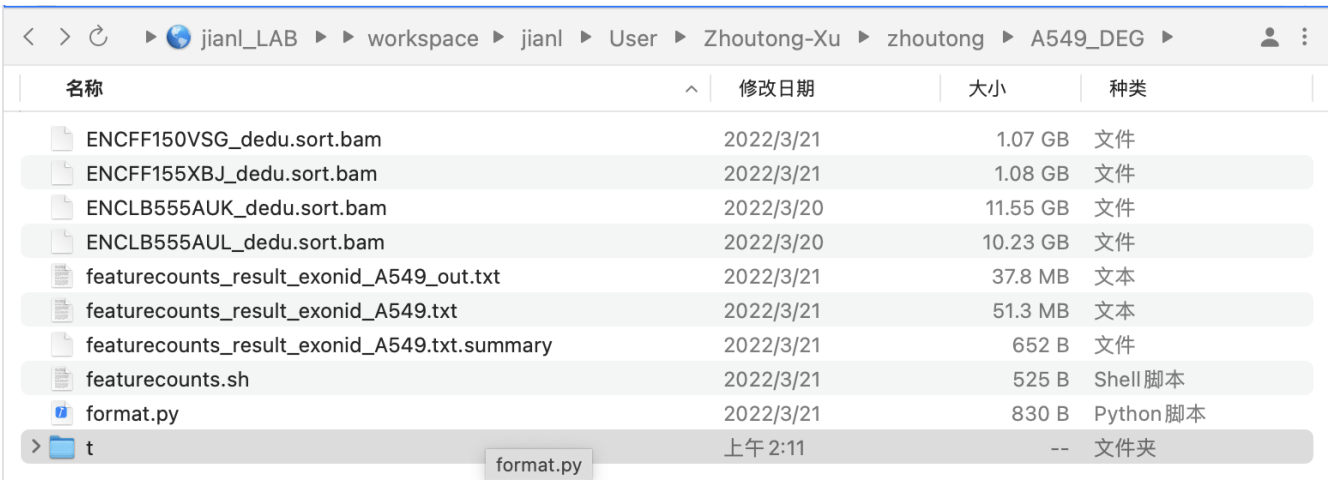
- 生成的文件如下（部分）：



四、Bedgraph 转 bw 文件（server）

- **background**：bw文件是最后能在网页上被可视化的文件格式，本部分将着重介绍如何批量将多个bedgraph文件转成bw文件

1. 将上一步生成的所有bedgraph格式文件上传至服务器中的“t”文件夹下（自己新建的文件夹）



2. 将/public/workspace/jianl/Software/DEGtools/generate_bedtobw.py 文件下载到“t”文件夹的上层目录中，修改相关的输入输出内容，运行此程序。代码及注意事项如下：

Python

```python
'''
generate bedtobw commands
'''
import os
path = '/root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/t'
os.chdir(path)

filenames = os.listdir(path)

outfile = open('/root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/bedtobw.txt','w')

for i in range(0,len(filenames)):
    outfile.write('/root/bin/x86_64/bedGraphToBigWig ./t/' +
                        filenames[i-1] +
                        ' /root/D/home/data/mysql/test/zhoutong/DEG/Reference/hg19.chrom.sizes ./bw/' +
                        filenames[i-1] +
                        '.bw\n\n')
```

- 获得的bedtobw.txt文件如下：

```
/root/bin/x86_64/bedGraphToBigWig ./t/g0_down_2.bedgraph /root/D/home/data/mysql/test/zhoutong/DEG/Reference/hg19.chrom.sizes ./bw/g0_down_2.bedgraph.bw
/root/bin/x86_64/bedGraphToBigWig ./t/g1_up_2.bedgraph /root/D/home/data/mysql/test/zhoutong/DEG/Reference/hg19.chrom.sizes ./bw/g1_up_2.bedgraph.bw
/root/bin/x86_64/bedGraphToBigWig ./t/g0_up_1.bedgraph /root/D/home/data/mysql/test/zhoutong/DEG/Reference/hg19.chrom.sizes ./bw/g0_up_1.bedgraph.bw
/root/bin/x86_64/bedGraphToBigWig ./t/g4_down_0.bedgraph /root/D/home/data/mysql/test/zhoutong/DEG/Reference/hg19.chrom.sizes ./bw/g4_down_0.bedgraph.bw
/root/bin/x86_64/bedGraphToBigWig ./t/g0_up_2.bedgraph /root/D/home/data/mysql/test/zhoutong/DEG/Reference/hg19.chrom.sizes ./bw/g0_up_2.bedgraph.bw
/root/bin/x86_64/bedGraphToBigWig ./t/g0_up_3.bedgraph /root/D/home/data/mysql/test/zhoutong/DEG/Reference/hg19.chrom.sizes ./bw/g0_up_3.bedgraph.bw
/root/bin/x86_64/bedGraphToBigWig ./t/g0_up_0.bedgraph /root/D/home/data/mysql/test/zhoutong/DEG/Reference/hg19.chrom.sizes ./bw/g0_up_0.bedgraph.bw
/root/bin/x86_64/bedGraphToBigWig ./t/g5_up_3.bedgraph /root/D/home/data/mysql/test/zhoutong/DEG/Reference/hg19.chrom.sizes ./bw/g5_up_3.bedgraph.bw
/root/bin/x86_64/bedGraphToBigWig ./t/g4_up_1.bedgraph /root/D/home/data/mysql/test/zhoutong/DEG/Reference/hg19.chrom.sizes ./bw/g4_up_1.bedgraph.bw
/root/bin/x86_64/bedGraphToBigWig ./t/g0_down_0.bedgraph /root/D/home/data/mysql/test/zhoutong/DEG/Reference/hg19.chrom.sizes ./bw/g0_down_0.bedgraph.bw
```

- 运行之： bash bedtobw.txt。即可得所有的bw文件。

五、添加track以及修改 trackDb.ra 文件

1. 在网站后台服务器的/root/D/home/data/mysql/test/ 下创建自己的文件夹，用于存放bw数据。

2. 文件夹外，新建a.py程序，代码如下

Python

```python
import os
import re

path = '/root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/bw/'
os.chdir(path)
filenames = os.listdir(path)


outfile = open('/root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/newtrack.txt','w')


file_title = []
file_regulate_info = []
for i in range(0,len(filenames)):
    file_title += re.findall(r'(\w+).bed',filenames[i])
    file_regulate_info += re.findall(r'_(\w+)_',filenames[i])

for i in range(0,len(filenames)):
    outfile.write('/root/bin/x86_64/hgBbiDbLink hg19 unfil_A549_' + file_regulate_info[i] + '_' + file_title[i] + ' /root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/bw/' + filenames[i] + '\n')
```

生成添加track的代码集合：

```
/root/bin/x86_64/hgBbiDbLink hg19 unfil_A549_up_g5_up_1 /root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/bw/g5_up_1.bedgraph.bw
/root/bin/x86_64/hgBbiDbLink hg19 unfil_A549_up_g0_up_1 /root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/bw/g0_up_1.bedgraph.bw
/root/bin/x86_64/hgBbiDbLink hg19 unfil_A549_down_g4_down_0 /root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/bw/g4_down_0.bedgraph.bw
/root/bin/x86_64/hgBbiDbLink hg19 unfil_A549_down_g5_down_1 /root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/bw/g5_down_1.bedgraph.bw
/root/bin/x86_64/hgBbiDbLink hg19 unfil_A549_down_g0_down_2 /root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/bw/g0_down_2.bedgraph.bw
/root/bin/x86_64/hgBbiDbLink hg19 unfil_A549_up_g5_up_3 /root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/bw/g5_up_3.bedgraph.bw
/root/bin/x86_64/hgBbiDbLink hg19 unfil_A549_down_g1_down_1 /root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/bw/g1_down_1.bedgraph.bw
/root/bin/x86_64/hgBbiDbLink hg19 unfil_A549_down_g2_down_0 /root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/bw/g2_down_0.bedgraph.bw
/root/bin/x86_64/hgBbiDbLink hg19 unfil_A549_up_g5_up_2 /root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/bw/g5_up_2.bedgraph.bw
/root/bin/x86_64/hgBbiDbLink hg19 unfil_A549_up_g2_up_2 /root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/bw/g2_up_2.bedgraph.bw
/root/bin/x86_64/hgBbiDbLink hg19 unfil_A549_up_g2_up_1 /root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/bw/g2_up_1.bedgraph.bw
/root/bin/x86_64/hgBbiDbLink hg19 unfil_A549_up_g4_up_0 /root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/bw/g4_up_0.bedgraph.bw
/root/bin/x86_64/hgBbiDbLink hg19 unfil_A549_down_g0_down_1 /root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/bw/g0_down_1.bedgraph.bw
```

运行之： bash newtrack_up.txt

3. 生成trackDb.ra的添加内容

- 运行b.py

Python

```python
import re
outfile = open('/root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/trackdb.txt','w')
inputfile = open('/root/D/home/data/mysql/test/zhoutong/DEG/A549-unfilter/newtrack.txt')
```

```
color_up = {'g0':'167,167,167', 'g1':'246,183,198', 'g2':'244,106,139', 'g3':'245,8,65', 'g4':'161,4,42', 'g5':'95,3,25'}
color_down = {'g0':'167,167,167', 'g1':'191,203,249', 'g2':'109,139,248', 'g3':'6,57,249', 'g4':'4,37,161', 'g5':'2,21,91'}
qvalue = {'g0':'adjusted p-value > 0.05', 'g1':'0.05 > adjusted p-value > 0.01', 'g2':'0.01 > adjusted p-value > 0.005', 'g3':'0.005 > adjusted p-value > 0.001','g4':'0.001 > adjusted p-value > 0.0005','g5':'adjusted p-value < 0.0005'}

line0 = []
line_all = []
track_name = []
for line in inputfile:
    line0 = line.strip('\n')
    line_all = line0.split(' ')
    track_name.append(line_all[2])
    line_all = []

qvalue_key = ''
file_regulate = ''
'''
qvalue_key = re.findall(r'g\d',track_name[2])

if re.findall(r'up', track_name[0]) == 'up' :
    file_regulate = 'up'
else :
    file_regulate = 'down'

print(qvalue_key[0])
print(file_regulate)
print(re.findall(r'up', track_name[0])[0])
'''

for i in range(0, len(track_name)):
    qvalue_key = re.findall(r'g\d',track_name[i])
    if re.findall(r'up', track_name[i]) == [] :
        file_regulate = 'down'
    elif re.findall(r'up', track_name[i])[0] == 'up' :
        file_regulate = 'up'
    if file_regulate == 'up' :
        outfile.write('track ' + track_name[i] + '\n' + 'shortLabel ' + track_name[i] + '\nlonglabel ' + qvalue[qvalue_key[0]] + ', up regulated\n' + 'type bigWig\n' + 'parent DEGs_A549_vs_normal_RNA\n' + 'color ' + color_up[qvalue_key[0]] + '\n\n')
    else :
        outfile.write('track ' + track_name[i] + '\n' + 'shortLabel ' + track_name[i] + '\nlonglabel ' + qvalue[qvalue_key[0]] + ', down regulated\n' + 'type bigWig\n' + 'parent DEGs_A549_vs_normal_RNA\n' + 'color ' + color_down[qvalu
```

```
    e_key[0]] + '\n\n')
42      qvalue_key = ''
43      file_regulate = ''
44
```

- 生成的文件如下：

```
track unfil_A549_up_g5_up_1
shortLabel unfil_A549_up_g5_up_1
longlabel adjusted p-value < 0.0005, up regulated
type bigWig
parent DEGs_A549_vs_normal_RNA
color 95,3,25

track unfil_A549_up_g0_up_1
shortLabel unfil_A549_up_g0_up_1
longlabel adjusted p-value > 0.05, up regulated
type bigWig
parent DEGs_A549_vs_normal_RNA
color 167,167,167

track unfil_A549_down_g4_down_0
shortLabel unfil_A549_down_g4_down_0
longlabel 0.001 > adjusted p-value > 0.0005, down regulated
type bigWig
parent DEGs_A549_vs_normal_RNA
color 4,37,161
```

- 将文件中的内容复制到trackDb.ra文件中，在server中运行 /root/bin/x86_64/hgTrackDb human hg19 trackDb_NEW /root/D/home/kent/src/hg/lib/trackDb.sql /root/D/home/data/mysql/ 即可更新服务器，完成数据的上传！

六、祝代码顺利，身体健康，少熬夜哦～