下载数据

GEO数据

https://mp.weixin.qq.com/s?__biz=MzAxMDkxODM1Ng==&mid=2247486063&idx=1&sn=156bee5397e979722b36b78284188538&scene=21#wechat_redirect

- GEO Platform (GPL)
- GEO Sample (GSM)
- GEO Series (GSE)
- GEO Dataset (GDS)

SRA数据

https://mp.weixin.qq.com/s?__biz=MzAxMDkxODM1Ng==&mid=2247486054&idx=1&sn=209975adee162228cfe6e6c5065c5c8c&scene=21#wechat_redirect

SRP(项目)—>SRS(样本)—>SRX(数据产生)—>SRR(数据本身)伴随数据库是project，层级是PRJNA—> SAMN

安装aspera

```
conda create -n download #创建环境
conda activate download  #激活环境
conda install -y -c hcc aspera-cli
conda install -y -c bioconda sra-tools
which ascp
```

用EBI下载数据

https://mp.weixin.qq.com/s?__biz=MzAxMDkxODM1Ng==&mid=2247492889&idx=2&sn=bc2ef17a3b96a257fb692f73338c6b0f&scene=21#wechat_redirect

```
time ascp -QT -l 300m -P33001 -i
/home/xiaoxiao/software/.aspera/connect/etc/asperaweb_id_dsa.openssh era-
fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR292/002/SRR2927022/SRR2927022_1.fastq.gz
/home/xiaoxiao/data/xiewei016/
```

安装aspera

下载到sofeware文件夹

```
wget http://download.asperasoft.com/download/sw/connect/3.7.4/aspera-connect-
3.7.4.147727-linux-64.tar.gz
tar zxvf aspera-connect-3.7.4.147727-linux-64.tar.gz
bash aspera-connect-3.7.4.147727-linux-64.sh
echo 'export PATH=~/home/xiaoxiao/.aspera/connect/bin:$PATH' >> ~/.bashrc
source ~/.bashrc
```

下载数据

```
time ascp -QT -l 300m -P33001 -i
/home/xiaoxiao/.aspera/connect/etc/asperaweb_id_dsa.openssh era-
fasp@fasp.sra.ebi.ac.uk:/vol1/fastq/SRR340/007/SRR3401567/SRR3401567.fastq.gz
/media/xiaoxiao/Zhanglabbbb/XLY/download/bovine_atac_NatComm
#-i /home/xiaoxiao/.aspera/connect/etc/asperaweb_id_dsa.openssh 密钥文件路径
#-K1
#-QT 断点续传
#-l 宽带限制
```

批量下载数据

```
cat /media/xiaoxiao/Zhanglabbbb/XLY/download/bovine_atac_NatComm/aspera|while read id;do
echo $id
time ascp -QT -l 300m -P33001 -i
/home/xiaoxiao/.aspera/connect/etc/asperaweb_id_dsa.openssh era-fasp@$id
/media/xiaoxiao/Zhanglabbbb/XLY/download/bovine_atac_NatComm
done
```

2021-01-20

```
#2-cell_rep1_ATACseq
ftp.sra.ebi.ac.uk/vol1/fastq/SRR108/021/SRR10887621/SRR10887621_1.fastq.gz
ftp.sra.ebi.ac.uk/vol1/fastq/SRR108/021/SRR10887621/SRR10887621_2.fastq.gz
......
```

批量下载，在EBI搜索PRJNA， Reade file 选择fastq_aspera, 得到全部下载链接。

选择experiment_tile

合并fastq文件

```
cat Sample_test_1.R1.fastq.gz Sample_test_2.R2.fastq.gz  >  test2.fastq.gz
```

质控

fastqc (base)

```
mamba install -c bioconda fastqc -y
```

multiqc (python34)

```
conda create -n python34 python=3.4 -y #配置python3.4 环境
conda activate python34
mamba install multiqc
```

```
fastqc *gz
#-o 输出到文件夹
#-t 线程
#-q 安静运行模式，不设时会实时报告运行状况
multiqc fastqc结果报告存放路径 -o 输出路径
```

trim-galore(base)

```
mamba install -c bioconda trim-galore -y #下载
```

trimmomatic(base)

```
单端
trimmomatic SE -phred33 input.fq.gz output.fq.gz ILLUMINACLIP:TruSeq3-SE:2:30:10
LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36
双端
trimmomatic PE -phred33 input_forward.fq.gz input_reverse.fq.gz
output_forward_paired.fq.gz output_forward_unpaired.fq.gz output_reverse_paired.fq.gz
output_reverse_unpaired.fq.gz ILLUMINACLIP:TruSeq3-PE.fa:2:30:10 LEADING:3 TRAILING:3
SLIDINGWINDOW:4:15 MINLEN:36
```

```
#2020-3-2 RBBP CHIP数据循环
for k in $(cat sample.list)
do
trimmomatic PE -phred33 1.rawdata/${k}/${k}_1.fq.gz   1.rawdata/${k}/${k}_2.fq.gz  -
baseout cleandata/${k}_clean.fq.gz ILLUMINACLIP:TruSeq3-PE-2.fa:2:30:10:8:true LEADING:3
TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:51  -threads 16
done
```

> 常用参数： -threads 线程数，最大是CPU核数 -trimlog 生成日志名，强烈建议不开这个参数，生成的log文件巨大且大多数情况下，基本不会看 -quiet 静默模式

- ILLUMINACLIP：从reads中剪切adapter和其他Illumina特定序列。
- SLIDINGWINDOW：执行滑动窗口修剪，一旦窗口内的平均质量低于阈值，则切割。

- LEADING：如果低于阈值质量，则在reads起始处剪切碱基
- TRAILING：如果低于阈值质量，则在reads末尾处剪切碱基
- CROP：将reads从末尾切割为指定长度
- HEADCROP：从reads剪切后低于指定长度，则删除
- MINLEN：如果reads低于指定长度，则删除
- TOPHRED33：将质量得分转换为Phred-33
- TOPHRED64：将质量得分转换为Phred-64

参考文档：http://www.usadellab.org/cms/index.php?page=trimmomatic

比对

bowtie2(base)

```
#下载小鼠参考基因组索引文件
wget -4 -q ftp://ftp.ccb.jhu.edu/pub/data/bowtie2_indexes/mm10.zip #-q:–quiet 不显示输出信息
unzip mm10.zip
```

```
#参考
bowtie2 -p 10 -x genome_index -1 input_1.fq -2 input_2.fq | samtools sort -O bam -@ 10 -o - > output.bam

bowtie2 -p 10 -t -q -N 1 -L 25 -X 2000 --no-mixed --no-discordant -x reference/mm10/mm10 -1 atac/xiewei2016/clean/8_cell_rep2_clean_val_1.fq.gz -2 atac/xiewei2016/clean/8_cell_rep2_clean_val_2.fq.gz | samtools sort -O bam -o atac/xiewei2016/align/8_cell_rep2.bam  &> atac/xiewei2016/align/bowtie2.log
```

```
#循环
#构建config.clean 文件，类似矩阵
ls *_1.fastq.gz > 1
ls *_2.fastq.gz > 2
ls *_2.fastq.gz |cut -d"/" -f 7|cut -d"_" -f 1  > 0
paste 0 1 2  > config.clean

cat config.clean |while read id;
do echo $id
arr=($id)
fq2=${arr[2]}
fq1=${arr[1]}
sample=${arr[0]}
bowtie2  -p 10 -t -q -N 1 -L 25 -X 2000 --no-mixed --no-discordant -x /media/xiaoxiao/zhanglab300/xly/reference/UCD1.2_bt2/UCD1.2_bt2  -1 $fq1 -2 $fq2 | samtools sort -O bam -o /media/xiaoxiao/zhanglab300/xly/bovine_NatComm_ATAC/align/${sample}.raw.bam &>/media/xiaoxiao/zhanglab300/xly/bovine_NatComm_ATAC/align/${sample}.log
done
```

```
#2020-03-02 RBBP4/7 chip数据,这个循环好用一些
for k in $(cat sample.list)
do
echo ${k}
bowtie2 -p 4 -x /media/xiaoxiao/zhanglab300/xly/reference/mm10/mm10 -1
cleandata/${k}_clean_1P.fq.gz -2 cleandata/${k}_clean_2P.fq.gz | samtools sort - O bam -o
bam/${k}.bam
done
```

必须参数

-x <bt2-idx> 由bowtie2-build所生成的索引文件的前缀。首先 在当前目录搜寻，然后在环境变量 BOWTIE2_INDEXES 中制定的文件夹中搜寻。

-1 <m1> 双末端测寻对应的文件1。可以为多个文件，并用逗号分开；多个文件必须和 -2 <m2> 中制定的文件一一对应。比如:"-1 flyA_1.fq,flyB_1.fq -2 flyA_2.fq,flyB_2.fq". 测序文件中的reads的长度可以不一样。

-2 <m2> 双末端测寻对应的文件2.

-U <r> 非双末端测寻对应的文件。可以为多个文件，并用逗号分开。测序文件中的reads的长度可以不一样。

-S <hit> 所生成的SAM格式的文件前缀。默认是输入到标准输出。

输入参数

-X 2000 参数，是最大插入片段，宽泛的插入片段范围(10-1000bp)

-p 线程

-q 输入的文件为FASTQ格式文件，此项为默认值。

输出参数

-t --time

Paired-end 参数

--no-mixed 默认设置下, 一对reads不能成对比对到参考序列上, 则单独对每个read进行比对. 该选项则阻止此行为.

--no-discordant 默认设置下, 一对reads不能和谐比对(concordant alignment, 即满足-I, -X, --fr/--rf/--ff的条件)到参考序列上, 则搜寻其不和谐比对(disconcordant alignment, 即两条reads都能独一无二地比对到参考序列上, 但是不满足-I,-X,--fr/--rf/--ff的条件). 该选项阻止此行为.

比对参数

-N <int> 进行种子比对时允许的mismatch数. 可以设为0或者1. Default: 0.

-L <int> 设定种子的长度.

bam文件

flag

1： 代表这个序列采用的是PE双端测序

2： 代表这个序列和参考序列完全匹配，没有插入缺失

4： 代表这个序列没有mapping到参考序列上

8： 代表这个序列的另一端序列没有比对到参考序列上，比如这条序列是R1,它对应的R2端序列没有比对到参考序列上

16：代表这个序列比对到参考序列的负链上

32： 代表这个序列对应的另一端序列比对到参考序列的负链上

64： 代表这个序列是R1端序列， read1;

128 : 代表这个序列是R2端序列，read2;

256： 代表这个序列不是主要的比对，一条序列可能比对到参考序列的多个位置，只有一个是首要的比对位置，其他都是次要的

512： 代表这个序列在QC时失败了，被过滤不掉了（# 这个标签不常用）

1024: 代表这个序列是PCR重复序列（#这个标签不常用）

2048: 代表这个序列是补充的比对（#这个标签具体什么意思，没搞清楚，但是不常用）

上面的这几个标签都是2的n次方，这样的数列有一个特点，就是随机挑选其中的几个，它们的和是唯一的，比如

65 只能是1 和 64 组成，代表这个序列是双端测序，而且是read1

https://www.cnblogs.com/xudongliang/p/5437850.html

```
1 @HD，说明符合标准的版本、对比序列的排列顺序；
2 @SQ，参考序列说明；
3 @RG，比对上的序列（read）说明；
4 @PG，使用的程序说明；
5 @CO，任意的说明信息。
```

过滤

去重 picard(base)

理论上来讲，不同的序列在进行PCR扩增时，扩增的倍数应该是相同的。但是由于聚合酶的偏好性，PCR扩增次数过多的情况下，会导致一些序列持续扩增，而另一些序列扩增到一定程度后便不再进

行，也就是我们常说的PCR偏好性

samtools如果多个reads具有相同的比对位置时，rmdup将它们标记为duplicates，然后去除重复，通常只保留第一个识别到的reads。

picard不仅考虑reads的比对位置，还会考虑其中的插入错配等情况（即会利用sam/bam文件中的CIGAR值），甚至reads的tail，lanel以及flowcell。Picard主要考虑reads的5'端的比对位置，一个每个reads比对上的方向。

因此我们可以从一定程度上认为，5'端的位置，方向，以及碱基比对情况相同，Picard就将这些reads中碱基比对值Q>15的看作是best pair而其他的reads则当作是duplicate reads。甚至当reads的长度不同时，Picard依然利用上述原理进行去重。

> 对Picard来说，reads的5'端信息更为重要.若duplicates是PCR重复，那么它们的序列不一定完全相同。但是由于PCR扩增时，酶的前进方向是5'->3'方向，PCR重复序列中5'端的部分相似的可能性更高。

```
mamba install picard -y

#1.排序(mapping 后samtools已经排序，测试这一步是否需要)
for k in $(cat sample.list)
do
echo ${k}
java -jar /home/xiaoxiao/miniconda3/share/picard-2.25.0-0/picard.jar SortSam -I
bam/${k}.bam -SORT_ORDER coordinate -O remove_duplicate_bam/${k}.sorted.bam
done

#2.直接去除重复
for k in $(cat sample.list)
do
echo ${k}
java -jar /home/xiaoxiao/miniconda3/share/picard-2.25.0-0/picard.jar MarkDuplicates -I
remove_duplicate_bam/${k}.sorted.bam  -O remove_duplicate_bam/${k}.bam  --METRICS_FILE
${k}.dupmarked.txt -REMOVE_DUPLICATES true
done

#只标记，不去除
for k in $(cat sample.list)
do
echo ${k}
java -jar /home/xiaoxiao/miniconda3/share/picard-2.25.0-0/picard.jar SortSam -I
remove_duplicate_bam/${k}.sorted.bam -SORT_ORDER coordinate -O
remove_duplicate_bam/${k}.markdup.bam -METRICS_FILE metrics.${k}.markdup.txt
done
```

去低MAPQ+线粒体

ChIP

```
#remove low quality mapping reads. 一般15
samtools view -b -q 15  -@16 输入文件名 > 输出文件名

for k in $(cat sample.list)
do
echo ${k}
samtools view -b -q 15  -@16 remove_duplicate_bam/${k}.bam > bam2/${k}.bam
done
```

```
-b output BAM
# 该参数设置输出 BAM 格式，默认下输出是 SAM 格式文件
-o FILE  output file name [stdout]
# 输出文件的名称
-O --output-fmt FORMAT[,OPT[=VAL]]...
               Specify output format (SAM, BAM, CRAM)
      --output-fmt-option OPT[=VAL]
               Specify a single output file format option in the form of OPTION or
OPTION=VALUE
-q INT   only include reads with mapping quality >= INT [0]
# 比对的最低质量值，一般认为20就为unique比对了，可以结合上述-bF参数使用使用提取特定的比对结果
-f INT   only include reads with all  of the FLAGs in INT present [0]
-@ 线程
```

ATAC-seq（去线粒体基因组+ 低MAPQ）****

```
#小鼠mm10  chrM
samtools view  -q 20 -@16 2-cell_amanitin_rep1.redup.bam |grep -v chrM| samtools sort  -O
bam  -@ 16 -o- > test.last.bam
#牛UCD1.2_bt2 MT
samtools view -h -q 20 -@16 2-cell_amanitin_rep1.redup.bam |grep -v MT| samtools sort  -O
bam  -@ 16 -o- > test.last.bam

for k in $(cat sample.list); do echo ${k}; samtools view -h -q 15 -@16
bam_raw/${k}.redup.bam |grep -v MT| samtools sort  -O bam  -@ 16 -o- > bam/${k}.bam; done
```

尝试能否三步和在一起

callpeak

macs2(base)

```
#下载
mamba install macs2 -y

macs2 callpeak -t remove_duplicate_bam/NC_IP1_FKDL210003323-1a.bam -c
remove_duplicate_bam/NC_Input1_FKDL210003322-1a.bam -f BAM -B -g mm -q 0.05 --nomodel --
```

```
shift 100 --extsize 200 -n NC_1 --outdir callpeak/ &> callpeak/NC_1.log

#H3K27ac
macs2 callpeak -c remove_duplicate_bam/NC_Input1_FKDL210003322-1a.bam -t
remove_duplicate_bam/NC_IP1_FKDL210003323-1a.bam -g 2652783500 -B -f BAMPE -p 1e-5 --
nomodel --broad --extsize 73 --SPMR  -n NC_1 --outdir callpeak2/ & > callpeak2/NC_1.log
```

-t/--treatment FIELNAME和-c/--control FILENAME表示处理样本和对照样本输入

-g表示实际可比对的基因组大小

--nomodel: 这个参数说明不需要MACS去构建模型，也就是说下面的参数除了--shift, --extsize外都会被无视

--extsize： MACS使用这个参数将read以5'-> 3'衍生至等长片段。比如说你知道你的转录因子的结合区域是200bp，那么参数就是--extsize 200。当且仅当--nomodel和--fix-bimodal设置使用

--shift: 这个参数是绝对的偏移值，会先于--extsize前对read进行整体移动。MACS会通过建模的方式自动计算出read需要偏移的距离，除非你对自己的数据非常了解，或者前期研究都表明结合中心在read后面的那个位置上，你才能比较放心的用这个这个参数了。正数表示从5'往3'偏移延长到片段中心，如果是负数则是3'往5'偏移延长到片段中心。

-n/--name表示实验的名字,

-f/--format FORMAT用来声明输入的文件格式，目前MACS能够识别的格式有 "ELAND", "BED", "ELANDMULTI", "ELANDEXPORT", "ELANDMULTIPET" (双端测序), "SAM", "BAM", "BOWTIE", "BAMPE", "BEDPE". 除"BAMPE", "BEDPE"需要特别声明外，其他格式都可以用AUTO自动检测

-B 输出bedgraph格式文件

deeptools可视化

deeptools(base)

```
mamba install deeptools -y
```

合并bam文件

```
#合并test_L1.bam和test_L2.bam文件
samtools merge -h test.sam \
test_L1_L2.bam \
test_L1.sorted.bam \
test_L2.sroted.bam
#合并test_L1.bam和test_L2.bam文件中的指定区域chr7
samtools merge -h test.sam \
```

```
-R chr7
test_L1_L2.bam \
test_L1.sorted.bam \
test_L2.sroted.bam
```

1.bam文件转bw文件

```
for k in $(cat sample.list)
do
echo ${k}
samtools index align/8_cell_rep2.last.bam
done    #会生成api文件

for k in $(cat sample.list)
do
echo ${k}
bamCoverage -b remove_duplicate_bam/${k}.bam -o bw/${k}.bw -of bigwig -bs 100 -p 8 --
normalizeUsing RPKM  --ignoreDuplicates --minMappingQuality 10
done    #bw文件在igv中可视化
```

--binSize, -bs	Size of the bins, in bases, for the output of the bigwig/bedgraph file. (Default: 50)
--numberOfProcessors, -p	

2.mm10的Refgene文件下载http://genome.ucsc.edu/cgi-bin/hgTables

mm10 Refgene文件TXT转为bed

```
perl -alne '{next if /^#/;if($F[3] eq "+")
{$start=$F[4];$end=$F[5]}else{$start=$F[4];$end=$F[5]}print
join("\t",$F[2],$start,$end,$F[12],0,$F[3])}' mm10.refseq.txt |sort -u >mm10.refseq.bed
```

3.生成上下游10kb的峰图

```
computeMatrix scale-regions -S bw/*.bw -R mm10.refseq.bed -p 6 -b 10000 -a 10000 --
regionBodyLength 10000 --skipZeros -o deeptool/matrix_body.gz --outFileNameMatrix
deeptool/matrix_body.tab --outFileSortedRegions deeptool/regions_body.bed

#RBBP CHIP
computeMatrix reference-point -S bw/*.bw -R ucsc.mm10.refseq_noID.tss.bed -p 10 -a 3000 -
b 3000 --referencePoint center -o deeptool/computeMatrix/tss.gz --skipZeros

plotHeatmap -m results/matrix2_8_cell_rep2_body.gz  -out
results/8_cell_rep2_body_Heatmap.png | plotProfile -m results/matrix2_8_cell_rep2_body.gz
-out results/8_cell_rep2_body_Profile.png

plotProfile -m matrix_body.gz -out profile2.png --perGroup #--perGroup 生成一个图
```

```
plotProfile -m deeptool/computeMatrix/tss.gz -out deeptool/computeMatrix/tss3.png --
perGroup --legendLocation upper-right --dpi 750
```

bedtools

```
mamba install bedtools -y
```

计算重复性

```
##将genome分成2000bp的bin
##下载chrom.sizes文件ftp://hgdownload.soe.ucsc.edu/goldenPath/bosTau8/bigZips/
##下载chrom.sizes文件ftp://hgdownload.soe.ucsc.edu/goldenPath/mm10/bigZips/
bedtools makewindows -g bosTau8.chrom.sizes -w 2000 > bosTau8_2000bin.bed
##sort
sort -k1,1 -k2,2n -k3,3n bosTau8_2000bin.bed > sorted_bosTau8_2000bin.bed  #或者先排序
bed，再makewindows.
sort -k1,1V -k2,2n -k3,3n sorted_mm10_2000bin.bed > test.bed #-V 用绝对值排序
##查看sort结果：cut -f 1 sorted_bosTau8_100bin.bed | uniq | head -n 40
##计算每个bin的reads数（不是RPKM）
##注意bam文件必须sorted，查看bam文件sort结果：samtools view -H Q20bam/2cip1_Q20.bam | head
-n 50
samtools view -H remove_duplicate_bam/NC_IP1_FKDL210003323-1a.bam

#计算测序深度,此步之前需要sort和index
samtools idxstats NC1.bam | awk '{if($1!="*"){total=total+$3}}END{print 1000000/total}'>
NC1.scale

##使用bedtools intersect或者coverage时，需要加-sorted，不然内存会爆掉
bedtools intersect -b Q20bam/2cip1_Q20.bam -a sorted_bosTau8_2000bin.bed -c -bed  -sorted
> bin2000/readCoverage/2cip1.readCoverage
bedtools intersect -b Q20bam/2cip2_Q20.bam -a sorted_bosTau8_2000bin.bed -c -bed  -sorted
> bin2000/readCoverage/2cip2.readCoverage

for k in $(cat sample.list)
do
echo ${k}
bedtools intersect -b Q20bam/2cip1_Q20.bam -a sorted_bosTau8_2000bin.bed -c -bed  -sorted
> bin2000/readCoverage/2cip1.readCoverage
done

#计算rpkm
awk -v FS='\t' '{print $1,$2,$3,$4*0.0415489}' bin2000/readCoverge/2cip1.readCoverage >
bin2000/readCoverage/rpkm/2cip1.rpkm
awk -v FS='\t' '{print $1,$2,$3,$4*0.042115}' bin2000/readCoverge/2cip2.readCoverage >
bin2000/readCoverage/rpkm/2cip2.rpkm

awk -v FS='\t' '{print $1,$2,$3,$4*0.0735089}' bin2000/readCoverage2/NC_IP1.readcoverage
> rpkm2/NC1.rpkm
```

```
#R
setwd('bovine_chipseq/bin2000/readCoverage/rpkm')
s2cip1<-read.table("2cip1.rpkm",sep='')
s2cip2<-read.table("2cip2.rpkm",sep='')

s8cip2<-read.table("8cip2.rpkm",sep='')
s8cip3<-read.table("8cip3.rpkm",sep='')

s16cip2<-read.table("16cip2.rpkm",sep='')
s16cip3<-read.table("16cip3.rpkm",sep='')

BLip1<-read.table("BLip1.rpkm",sep='')
BLip2<-read.table("BLip2.rpkm",sep='')

GVip1<-read.table("GVip1.rpkm",sep='')
GVip3<-read.table("GVip3.rpkm",sep='')

MIIip1<-read.table("MIIip1.rpkm",sep='')
MIIip2<-read.table("MIIip2.rpkm",sep='')

Moip2<-read.table("Moip2.rpkm",sep='')
Moip3<-read.table("Moip3.rpkm",sep='')

pw_plot <- function(x, y,
                    xlab="x",
                    ylab="y", ...){
  log2x <- log2(x)
  log2y <- log2(y)
  smoothScatter(log2x,log2y,
                cex=1.2,
                xlim=c(0,12),ylim=c(0,12),
                xlab=xlab,
                ylab=ylab)
  text(3,10,paste("R = ",round(cor(x,y),2),sep=""))

}

par(mfrow=c(2,3))

pw_plot(s2cip1[,4], s2cip2[,4],
        xlab = "2C_Rep1 (Log2 RPKM)",
        ylab = "2C_Rep2 (Log2 RPKM)")


pw_plot(s8cip2[,4], s8cip3[,4],
        xlab = "8C_Rep1 (Log2 RPKM)",
        ylab = "8C_Rep2 (Log2 RPKM)")


pw_plot(s16cip2[,4], s16cip3[,4],
        xlab = "16C_Rep1 (Log2 RPKM)",
        ylab = "16C_Rep2 (Log2 RPKM)")


pw_plot(BLip1[,4], BLip2[,4],
        xlab = "BL_Rep1 (Log2 RPKM)",
        ylab = "BL_Rep2 (Log2 RPKM)")
```

```
pw_plot(GVip1[,4], GVip3[,4],
        xlab = "GV_Rep1 (Log2 RPKM)",
        ylab = "GV_Rep2 (Log2 RPKM)")



pw_plot(MIIip1[,4], MIIip2[,4],
        xlab = "MII_Rep1 (Log2 RPKM)",
        ylab = "MII_Rep2 (Log2 RPKM)")



pw_plot(RMoip2[,4], Moip3[,4],
        xlab = "Mo_Rep1 (Log2 RPKM)",
        ylab = "Mo_Rep2 (Log2 RPKM)")
```

R

sep

是函数的形式参数，多数情况下， seq 参数用来指定字符的分隔符号。 不仅用在你所提到的输出，也用在输入，也用在字符串的合并与拆分上。

csv 文件是用逗号分隔的，故而 sep = "," tsv 文件是用制表符分隔的，故而 sep = "\t" 常用的分隔符还有空格 sep = " " 分隔符是任意的，可根据具体情况指定的。

在输入的时候，原内容是用什么符号分隔的，sep就要保持一致，否则可能无法正确读取。

在输出时虽说分隔符是可以任意指定，但也要遵循一个原则，就是分隔符号不要与待输出内容中的字符有重复。否则输出后的文件，重新读取的时候该分隔符并不能有效正确分开，可能出错。

samtools

view命令的主要功能是：将sam文件与bam文件互换；然后对bam文件进行各种操作，比如数据的排序(sort)和提取(这些操作是对bam文件进行的，因而当输入为sam文件的时候，不能进行该操作)；最后将排序或提取得到的数据输出为bam或sam（默认的）格式。 bam文件优点：bam文件为二进制文件，占用的磁盘空间比sam文本文件小；利用bam二进制文件的运算速度快。 view命令中，对sam文件头部（序列ID）的输入(-t或-T）和输出(-h)是单独的一些参数来控制的

1.view

```
-b output BAM
# 该参数设置输出 BAM 格式，默认下输出是 SAM 格式文件
-h print header for the SAM output
# 默认下输出的 sam 格式文件不带 header，该参数设定输出sam文件时带 header 信息
-H print SAM header only (no alignments)
# 仅仅输出文件的头文件
-S input is SAM
# 默认下输入是 BAM 文件，若是输入是 SAM 文件，则最好加该参数，否则有时候会报错。
-u uncompressed BAM output (force -b)
# 该参数的使用需要有-b参数，能节约时间，但是需要更多磁盘空间。
-c print only the count of matching records
# 仅输出匹配的统计记录
-L FILE  only include reads overlapping this BED FILE [null]
#  仅包括和bed文件存在overlap的reads
-o FILE  output file name [stdout]
# 输出文件的名称
-F INT  only include reads with none of the FLAGS in INT present [0]
# 过滤flag，仅输出指定FLAG值的序列
-q INT   only include reads with mapping quality >= INT [0]
# 比对的最低质量值，一般认为20就为unique比对了，可以结合上述-bF参数使用使用提取特定的比对结果
-@ Number of additional threads to use [0]
# 指使用的线程数



# 将sam文件转换成bam文件
samtools view -bS abc.sam > abc.bam
# BAM转换为SAM
samtools view -h -o out.sam out.bam
# 提取比对到参考序列上的比对结果
samtools view -bF 4 abc.bam > abc.F.bam
# 提取paired reads中两条reads都比对到参考序列上的比对结果，只需要把两个4+8的值12作为过滤参数即
可
samtools view -bF 12 abc.bam > abc.F12.bam
# 提取没有比对到参考序列上的比对结果
samtools view -bf 4 abc.bam > abc.f.bam
# 提取bam文件中比对到caffold1上的比对结果，并保存到sam文件格式
samtools view abc.bam scaffold1 > scaffold1.sam
# 提取scaffold1上能比对到30k到100k区域的比对结果
samtools view abc.bam scaffold1:30000-100000 $gt; scaffold1_30k-100k.sam
# 根据fasta文件，将 header 加入到 sam 或 bam 文件中
samtools view -T genome.fasta -h scaffold1.sam > scaffold1.h.sam

-b       output BAM
  -C       output CRAM (requires -T)
  -1       use fast BAM compression (implies -b)
  -u       uncompressed BAM output (implies -b)
  -h       include header in SAM output
  -H       print SAM header only (no alignments)
  -c       print only the count of matching records
  -o FILE  output file name [stdout]
  -U FILE  output reads not selected by filters to FILE [null]
  -t FILE  FILE listing reference names and lengths (see long help) [null]
  -X       include customized index file
```

```
  -L FILE  only include reads overlapping this BED FILE [null]
  -r STR   only include reads in read group STR [null]
  -R FILE  only include reads with read group listed in FILE [null]
  -d STR:STR
           only include reads with tag STR and associated value STR [null]
  -D STR:FILE
           only include reads with tag STR and associated values listed in
           FILE [null]
  -q INT   only include reads with mapping quality >= INT [0]
  -l STR   only include reads in library STR [null]
  -m INT   only include reads with number of CIGAR operations consuming
           query sequence >= INT [0]
  -f INT   only include reads with all  of the FLAGs in INT present [0]
  -F INT   only include reads with none of the FLAGS in INT present [0]
  -G INT   only EXCLUDE reads with all  of the FLAGs in INT present [0]
  -s FLOAT subsample reads (given INT.FRAC option value, 0.FRAC is the
           fraction of templates/read pairs to keep; INT part sets seed)
  -M       use the multi-region iterator (increases the speed, removes
           duplicates and outputs the reads as they are ordered in the file)
  -x STR   read tag to strip (repeatable) [null]
  -B       collapse the backward CIGAR operation
  -?       print long help, including note about region specification
  -S       ignored (input format is auto-detected)
  --no-PG  do not add a PG line
      --input-fmt-option OPT[=VAL]
               Specify a single input file format option in the form
               of OPTION or OPTION=VALUE
  -O, --output-fmt FORMAT[,OPT[=VAL]]...
               Specify output format (SAM, BAM, CRAM)
      --output-fmt-option OPT[=VAL]
               Specify a single output file format option in the form
               of OPTION or OPTION=VALUE
  -T, --reference FILE
               Reference sequence FASTA FILE [null]
  -@, --threads INT
               Number of additional threads to use [0]
      --write-index
               Automatically index the output files [off]
      --verbosity INT
               Set level of verbosity
```

2.flagstat

给出BAM文件的比对结果，并输出比对统计结果。除了`-@`参数指定线程，没有其他的参数

```
samtools flagstat tmp.bam > tmp.stat

cat SRR3545580.raw.stat
20000 + 0 in total (QC-passed reads + QC-failed reads)# QC pass的reads的数量为20000，未通过QC的reads数量为0，意味着一共有20000条reads
0 + 0 secondary#0条reads比对到第二个地方
0 + 0 supplementary#0条reads只比对上部分
0 + 0 duplicates#重复reads的数量，QC pass和failed
18995 + 0 mapped (94.98% : N/A)# 比对到参考基因组上的reads数量，总体上reads的匹配率
```

```
20000 + 0 paired in sequencing#paired reads数据数量
10000 + 0 read1# reads1中的reads数
10000 + 0 read2# reads2中的reads数
18332 + 0 properly paired (91.66% : N/A# 完美匹配的reads数和比例：比对到同一条参考序列，并且两条reads之间的距离符合设置的阈值
18416 + 0 with itself and mate mapped# paired reads中两条都比对到参考序列上的reads数
579 + 0 singletons (2.90% : N/A)# 单独一条匹配到参考序列上的reads数，和上一个相加，则是总的匹配上的reads数。
0 + 0 with mate mapped to a different chr# paired reads中两条分别比对到两条不同的参考序列的reads数
0 + 0 with mate mapped to a different chr (mapQ>=5)# 同上一个，只是其中比对质量>=5的reads的数量
```

3.idxstats

统计一个表格，4列，分别为”序列名，序列长度，比对上的reads数，unmapped reads number”。第4列应该是paired reads中有一端能匹配到该scaffold上，而另外一端不匹配到任何scaffolds上的reads数。

```
samtools idxstats aln.bam  > aln.stat
```