WES pipeline

Ref:
https://zhuanlan.zhihu.com/p/137078769
https://www.xinzipanghuang.net/wes-pipeline-3/
https://www.jianshu.com/p/4b677654be15?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation

0 log in to the server

```
ssh -p 17019 zhiwen@111.229.50.198
wanwan0712

ssh -p 17019 jiajun@111.229.50.198
335566cz
```

1 download sra files

```
$ wget https://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/2.9.6/sratoolkit.2.9.6-ubu
 ( not sure which version to install )
```

#环境配置

```
vi ~/.bashrc
i  #切换成插入
export PATH=/home/software/anaconda3/bin:$PATH
export PATH=/home/zhiwen/FastQC/bin:$PATH
export PATH=/home/zhiwen/sratoolkit.2.9.6-ubuntu64/bin:$PATH
(#这里的绝对路径是通过在存放sratoolkit文件夹下输入pwd查看得到的)
Esc  #回到命令
：wq  #保存退出.bashrc文件
source ~/.bashrc  ( #使配置生效 )
```

#下载文件

```
prefetch SRR10059492 (PC1)
prefetch SRR10059491 (PC2)
prefetch SRR10059490 (PC3)
```

2 mkdir

```
mkdir raw
mkdir afterQC
```

```
mkdir aligned
mkdir genome
mkdir hg19_vcf
```

3 Trans sra to fastq

/home/zhiwen/sratoolkit.2.9.6-ubuntu64/bin/fastq-dump --split-3 SRR10059491.1 -O /home/zhiwen/download/

[Rejected 121227 READS because of filtering out non-biological READS
Read 88150293 spots for SRR14700194.sra
Written 88150293 spots for SRR14700194.sra]
Problem came through: fastq-dump也用的是绝对路径，比较之慢，可以考虑后台

4 QC (with FastQC)

./fastqc [-o output dir] [--(no)extract] [-f fastq"bam"sam]
[-c contaminant file] seqfile1 .. seqfileN

```
wget https://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc_v0.11.
unzip fastqc_v0.11.9.zip
cd FastQC/
chmod 755 fastqc
./fastqc -h (test for install)
```

Problem came through: -bash: ./fastqc: 没有那个文件或目录, 在环境里配置了路径也没有解决。最后方案：使用绝对路径/home/zhiwen/FastQC/fastqc

#check QC

```
scp -P 1701 zhiwen@111.229.50.198:/home/zhiwen/download/SRR14700194_1_fastqc
scp -P 1701 zhiwen@111.229.50.198:/home/zhiwen/download/SRR14700194_2_fastqc
Problem came through: scp是在本地主机上使用的
scp -P 1701  jiajun@111.229.50.198:/home/jiajun/raw/*.html ./desktop
```

FastQC结果解读：https://www.jianshu.com/p/134c45339805
QC结果好，质量高，没必要cut

5 Reference Sequence Download

GRCh37 = hg19

```
for i in $(seq 1 22) X Y M;
do
nohup wget http://hgdownload.cse.ucsc.edu/goldenPath/hg19/chromosomes/chr${i
done
```

#uncompress

```
gunzip *.gz
```

##以染色体序号排序，这是GATK分析的要求

```
for i in $(seq 1 22) X Y M;
do cat chr${i}.fa >> hg19.fa;
done
```

#download bwa

```
wget https://sourceforge.net/projects/bio-bwa/files/bwa-0.7.17.tar.bz2
tar -jxvf bwa-0.7.17.tar.bz2
cd bwa-0.7.17/
make
```

6 Index

BWA是一款基于BWT的快速比对工具，其由三个算法组成。这三个算法分别是：BWA backtrack, BWA SW and BWA MEM。其中，BWA MEM是最新的，其更快更准确，更适合用于人重数据分析。对于上述三种算法，首先需要使用索引命令构建参考基因组的索引，用于后面的比对。所以，使用BWA整个比对过程主要分为两步，第一步建索引，第二步使用BWA MEM进行比对。BWA命令中参数众多，这里不一一讲解，只讲解最常用的几个，具体命令如下：

```
BWA-backtrack：是用来比对 Illumina 的序列的，reads 长度最长能到 100bp。
BWA-SW：用于比对 long-read ，支持的长度为 70bp-1Mbp；同时支持剪接性比对。
BWA-MEM：推荐使用的算法，支持较长的read长度，同时支持剪接性比对（split alignments）
```

建立索引：bwa index [–p prefix] [–a algoType] <ref.fa>

参数详解：ref.fa——参加基因组文件，作为输入文件；
p——输出文件前缀；

a——构建索引的算法；包括两个算法，分别是is和bwtsw。对于参考基因组文件大于2G的使用bwtsw算法，使用bwtsw算法必须保证参考基因组文件大小大于10M。
#原文链接：https://blog.csdn.net/u013553061/article/details/53120973
BWA使用详解: https://www.jianshu.com/p/19f58a07e6f4

```
cd genome
nohup ../bwa-0.7.17/bwa index -p hg19 -a bwtsw ../hg19/hg19.fa &
```

7 Alignment

7.1 bwa

https://www.jianshu.com/p/19f58a07e6f4
–t，线程数；

–M , –M 将 shorter split hits 标记为次优，以兼容 Picard markDuplicates 软件;
–R 接的是 Read Group的字符串信息，它是用来将比对的read进行分组的，这个信息对于我们后续对比对数据进行错误率分析和Mark duplicate时非常重要。
（1) ID，这是Read Group的分组ID，一般设置为测序的lane ID
（2) PL，指的是所用的测序平台
（3) SM，样本ID
（4) LB，测序文库的名字
这些信息设置好之后，在RG字符串中要用制表符（\t）将它们分开

```
cd afterQC

nohup ../bwa-0.7.17/bwa mem -t 4 -M -R "@RG\tID:lane1\tPL:illumina\tLB:libra

nohup ../bwa-0.7.17/bwa mem -t 4 -M -R "@RG\tID:lane1\tPL:illumina\tLB:libra

nohup ../bwa-0.7.17/bwa mem -t 4 -M -R "@RG\tID:lane1\tPL:illumina\tLB:libra
```

7.2 安装miniconda

https://www.jianshu.com/p/fab0068a32b4

```
nohup wget -c https://mirrors.tuna.tsinghua.edu.cn/anaconda/miniconda/Minico

nohup wget -c https://mirrors.tuna.tsinghua.edu.cn/anaconda/miniconda/Minico
```

```
bash Miniconda-3.0.0-Linux-x86_64.sh

source ~/.bashrc
```

7.3 samtools

#download:
https://www.jianshu.com/p/6b7a442d293f

```
Conda install samtools
```

Error: –bash: /home/software/anaconda3/bin/conda:
/home/ruiqi/anaconda3/bin/python: bad interpreter: Permission denied

```
nohup wget -c https://github.com/samtools/samtools/releases/download/1.9/sam

tar jxvf samtools-1.9.tar.bz2
./configure --prefix=/home/zhiwen/samtools-1.9（需要用绝对路径）
make
make install
./samtools --help
```

#error
samtools: error while loading shared libraries: libcrypto.so.1.0.0: cannot open shared object file: No such file or directory
#solution: https://www.cnblogs.com/huanping/p/13786701.html
ln –s /usr/lib64/libcrypto.so.1.0.2k ~/miniconda3/lib/libcrypto.so.1.0.0

#使用Samtools将比对后的文件进行排序，统计

```
cd aligned/
samtools view -b -S wes.sam > wes.bam  #sam to bam,便于存储
samtools sort wes.bam -o wes.sorted.bam    #有顺序的排序，便于后面的操作
samtools flagstat wes.sorted.bam > wes.sorted.bam.flagstat    #统计比对信息
```

#results
#SRR10059490

```
cat SRR10059490_wes.sorted.bam.flagstat
68818602 + 0 in total (QC-passed reads + QC-failed reads)
111592 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
68145279 + 0 mapped (99.02% : N/A)
68707010 + 0 paired in sequencing
34353505 + 0 read1
34353505 + 0 read2
67126046 + 0 properly paired (97.70% : N/A)
67731174 + 0 with itself and mate mapped
302513 + 0 singletons (0.44% : N/A)
461146 + 0 with mate mapped to a different chr
372476 + 0 with mate mapped to a different chr (mapQ>=5)
```

#SRR10059491

```
cat SRR10059491_wes.sorted.bam.flagstat
83879323 + 0 in total (QC-passed reads + QC-failed reads)
121639 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
82897448 + 0 mapped (98.83% : N/A)
83757684 + 0 paired in sequencing
41878842 + 0 read1
41878842 + 0 read2
81797668 + 0 properly paired (97.66% : N/A)
82457088 + 0 with itself and mate mapped
318721 + 0 singletons (0.38% : N/A)
484562 + 0 with mate mapped to a different chr
383931 + 0 with mate mapped to a different chr (mapQ>=5)
```

#SRR10059492

```
cat SRR10059492_wes.sorted.bam.flagstat
```

```
84818591 + 0 in total (QC-passed reads + QC-failed reads)
152473 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
83761877 + 0 mapped (98.75% : N/A)
84666118 + 0 paired in sequencing
42333059 + 0 read1
42333059 + 0 read2
78523788 + 0 properly paired (92.75% : N/A)
83287486 + 0 with itself and mate mapped
321918 + 0 singletons (0.38% : N/A)
4376674 + 0 with mate mapped to a different chr
4089256 + 0 with mate mapped to a different chr (mapQ>=5)
```

#pretty good

8 GATK

8.1 download:

```
wget https://github.com/broadinstitute/gatk/releases/download/4.2.0.0/gatk-4

unzip gatk-4.2.0.0.zip

#配置环境
export PATH=/home/zhiwen/gatk-4.2.0.0:$PATH

#检测运行
gatk --help
gatk --list
```

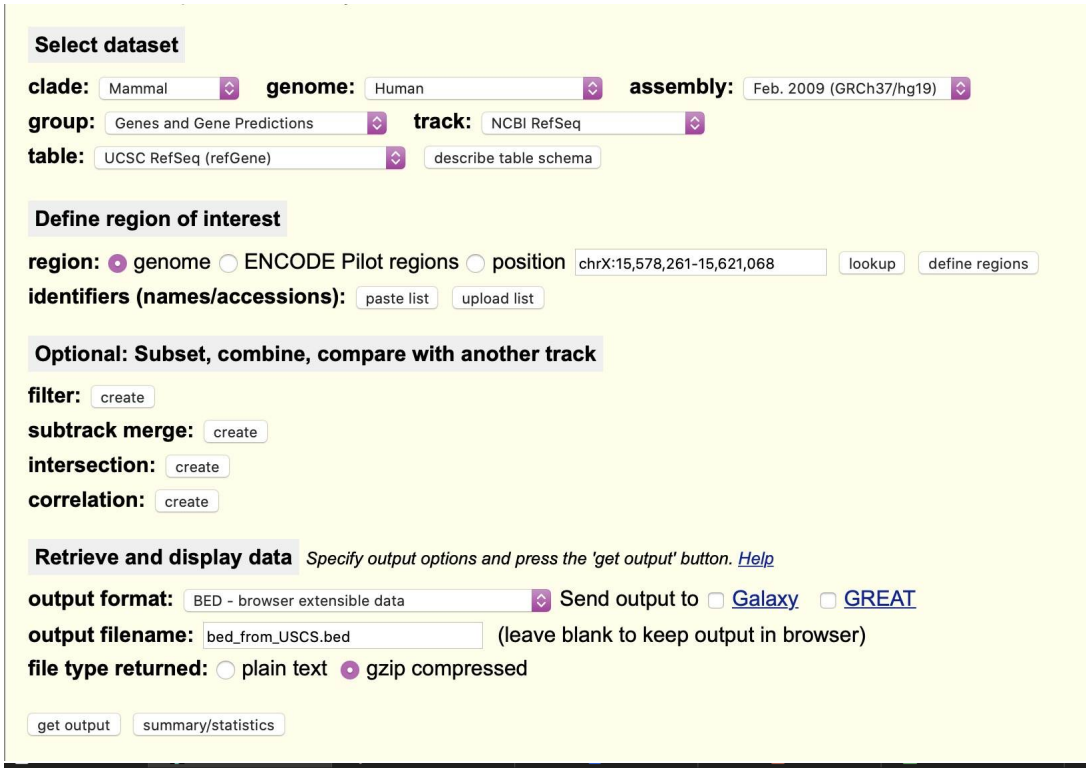
#Or download to laptop first and then

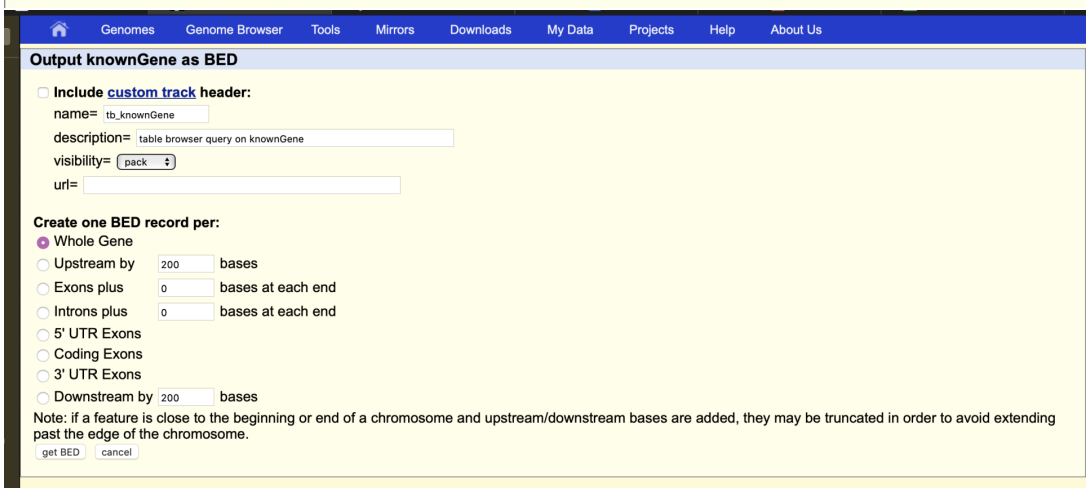
```
scp -r -P 1701 /Downloads/gatk-4.2.0.0  zhiwen@111.229.50.198:/home/zhiwen/
```

8.2 interval list

#1 first download bed file
#外显子区域覆盖度 需要生成外显子interval文件，生成这个文件的前提又需要dict文件和外显子bed文件(也可以去UCSC下载)
#方法一：first download bed file from USCS and then scp to Tencent cloud

```
http://genome.ucsc.edu/cgi-bin/hgTables?
hgsid=1131938085_hzQmQ24tR7FzoxKW0tC5VkkoqEvQ&clade=mammal&org=&db=hg19&hgta
```





#方法二

#ssh –p 1701 jiajun@111.229.50.198

#scp –P 1701 bed_from_UCSC_simplied.bed zhiwen@111.229.50.198:/home/zhiwen/aligned/

#2 Make dictionary

```
cd genome
../gatk-4.2.0.0/gatk CreateSequenceDictionary -R hg19.fa -O hg19.dict
```

#3 bed to interval list (wrong)

```
cd aligned
../gatk-4.2.0.0/gatk BedToIntervalList -I bed_from_UCSC.bed -O Exon.interval
```

###error: picard.PicardException: Sequence 'chr6_apd_hap1' was not found in the sequence dictionary ## (还有一个chrMT，我把bed里面的chrMT都换成chrM了，因为hg19.dict里面是chrM)

```
    at picard.util.BedToIntervalList.doWork(BedToIntervalList.java:156)
```

```
    at picard.cmdline.CommandLineProgram.instanceMain(CommandLineProgram.jav
    at org.broadinstitute.hellbender.cmdline.PicardCommandLineProgramExecutc
    at org.broadinstitute.hellbender.Main.runCommandLineProgram(Main.java:16
    at org.broadinstitute.hellbender.Main.mainEntry(Main.java:203)
    at org.broadinstitute.hellbender.Main.main(Main.java:289)
###solution (temporal)
#Simplify the bed file to which just contains chr1-21 and chrM. Without chr6
```

#3 bed to interval list (correct)

```
../gatk-4.2.0.0/gatk BedToIntervalList -I bed_from_UCSC_simplied.bed -O Ex
on.interval.bed -SD ../genome/hg19.dict
```

#4 CollectHsMetrics (wrong)

```
cd aligned
list="SRR10059490 SRR10059491 SRR10059492"
for line in $list;
do
echo "../gatk-4.2.0.0/gatk CollectHsMetrics -BI Exon.interval.bed -TI Exon.i
```

#error: Exception in thread ”main” java.lang.OutOfMemoryError: Java heap space
#solution: ––java–options ”–Xmx50g”

#new one CollectHsMetrics

```
cd aligned
list="SRR10059490 SRR10059491 SRR10059492"
../gatk-4.2.0.0/gatk --java-options "-Xmx50g" CollectHsMetrics -BI Exon.inte

../gatk-4.2.0.0/gatk --java-options "-Xmx50g" CollectHsMetrics -BI Exon.inte

../gatk-4.2.0.0/gatk --java-options "-Xmx50g" CollectHsMetrics -BI Exon.inte
```

#outcome of CollectHsMetrics 有点像质控

#5 MarkDuplicates (done)
#标记PCR重复序列并建立索引 http://www.360doc.com/content/19/1224/14/68068867_881793271.shtml
#作用是用于下游snp分析作准备

```
cd aligned
list="SRR10059490 SRR10059491 SRR10059492"

for line in $list;
do
echo "../gatk-4.2.0.0/gatk --java-options "-Xmx10G -Djava.io.tmpdir=../tmp/"
-O ${line}_wes.sorted.MarkDuplicates.bam
-M ${line}_wes.sorted.bam.metrics && samtools index ${line}_wes.sorted.MarkD

```

```
#new version1
list="father mother"
for line in $list;
do
echo "gatk --java-options "-Xmx10G" MarkDuplicates -I ${line}.sorted.bam -O

```

for loop没成功，不知道为什么，改为逐行了

#wes.sorted.bam.metrics有统计信息 wes.sorted.MarkDuplicates.bam创建索引文件，他的作用能够让我们可以随机访问这个文件中的任意位置，而且后面的步骤也要求这个bam文件一定要有索引。

```
../gatk-4.2.0.0/gatk --java-options "-Xmx10g" MarkDuplicates -I SRR10059490_
samtools view -f 1024 SRR10059490_wes.sorted.MarkDuplicates.bam | less
samtools index SRR10059490_wes.sorted.MarkDuplicates.bam &

../gatk-4.2.0.0/gatk --java-options "-Xmx10g" MarkDuplicates -I SRR10059491_
samtools index SRR10059491_wes.sorted.MarkDuplicates.bam &

../gatk-4.2.0.0/gatk --java-options "-Xmx10g" MarkDuplicates -I SRR10059492_
samtools index SRR10059492_wes.sorted.MarkDuplicates.bam &

```

"-Xmx50g" 有概率报错：内存不够， 改为"-Xmx10g"

8.2 变异检测

#重新校正碱基质量值（BQSR）

变异检测是一个极度依赖测序碱基质量值，因为这个质量值是衡量我们测序出来的这个碱基到底有多正确的重要指标。它来自于测序图像数据的base calling，因此，基本上是由测序仪和测序系统来决定的，计算出来的碱基质量值未必与真实结果统一。 BQSR（Base Quality Score Recalibration）这个步骤主要是通过机器学习的方法构建测序碱基的错误率模型，然后对这些碱基的质量值进行相应的调整。 这里包含了两个步骤： 第一步，BaseRecalibrator，这里计算出了所有需要进行重校正的read和特征值，然后把这些信息输出为一份校准表文件（wes.recal_data.table） 第二步，ApplyBQSR，这一步利用第一步得到的校准表文件（wes.recal_data.table）重新调整原来BAM文件中的碱基质量值，并使用这个新的质量值重新输出一份新的BAM文件。

使用的vcf(存疑？)

```
1000G_phase1.indels.hg19.sites.vcf.gz
1000G_phase1.snps.high_confidence.hg19.sites.vcf.gz
dbsnp_138.hg19.vcf.gz
```

文件都在aligned目录下：

```
#baserecalibrator
gatk --java-options "-Xmx20G" BaseRecalibrator \
   -I SRR10059490_wes.sorted.MarkDuplicates.bam \
   -R /home/zhiwen/genome/hg19.fa \
   --known-sites 1000G_phase1.indels.hg19.sites.vcf \
   --known-sites dbsnp_138.hg19.vcf \
   --known-sites 1000G_phase1.snps.high_confidence.hg19.sites.vcf \
   -O SRR10059490_recal.table &

#ApplyBQSR
gatk --java-options "-Xmx10G" ApplyBQSR \
  -R /home/zhiwen/genome/hg19.fa \
  -I SRR10059490_wes.sorted.MarkDuplicates.bam \
  -bqsr SRR10059490_recal.table \
  -O ../BQSR/SRR10059490_wes.sorted.MarkDuplicates.BQSR.bam &

same for SRR10059491 SRR10059492

#new version1
#baserecalibrator
cd 5bqsr
gatk --java-options "-Xmx20G" BaseRecalibrator \
-I ../4markduplicates/father.sorted.MarkDuplicates.bam \
-R /home/jiajun/genome/hg19.fa \
--known-sites /home/jiajun/putaotai/aligned/hg19_vcf/Mills_and_1000G_gold_st
--known-sites /home/jiajun/putaotai/aligned/hg19_vcf/dbsnp_138.hg19.vcf \
--known-sites /home/jiajun/putaotai/aligned/hg19_vcf/1000G_phase1.snps.high_
-O father_recal.table &

#ApplyBQSR

gatk --java-options "-Xmx10G" ApplyBQSR \
  -R /home/jiajun/genome/hg19.fa \
  -I ../4markduplicates/father.sorted.MarkDuplicates.bam \
  -bqsr father_recal.table \
  -O father.sorted.MarkDuplicates.BQSR.bam &


```

A USER ERROR has occurred: Fasta index file file:///home/zhiwen/genome/hg19.fa.fai for reference file:///home/zhiwen/genome/hg19.fa does not exist.

solution: 没有生成fa.fai文件，需要先执行faidx

```
samtools faidx hg19.fa
```

A USER ERROR has occurred: Input 1000G_phase1.indels.hg19.sites.vcf must support random access to enable queries by interval. If it's a file, please index it using the bundled tool IndexFeatureFile

solution: 对每个vcf文件index 或者 解压对应的gz文件

```
$ gatk IndexFeatureFile -I *.vcf
or
gzip -d *.gz
```

result

BaseRecalibrator – 1286243 read(s) filtered by: MappingQualityNotZeroReadFilter
0 read(s) filtered by: MappingQualityAvailableReadFilter
0 read(s) filtered by: MappedReadFilter
38376 read(s) filtered by: NotSecondaryAlignmentReadFilter
5999545 read(s) filtered by: NotDuplicateReadFilter
0 read(s) filtered by: PassesVendorQualityCheckReadFilter
0 read(s) filtered by: WellformedReadFilter
7324164 total reads filtered

BaseRecalibrator – 1718801 read(s) filtered by: MappingQualityNotZeroReadFilter
0 read(s) filtered by: MappingQualityAvailableReadFilter
0 read(s) filtered by: MappedReadFilter
41143 read(s) filtered by: NotSecondaryAlignmentReadFilter
5886438 read(s) filtered by: NotDuplicateReadFilter
0 read(s) filtered by: PassesVendorQualityCheckReadFilter
0 read(s) filtered by: WellformedReadFilter
7646382 total reads filtered

BaseRecalibrator – 1970343 read(s) filtered by: MappingQualityNotZeroReadFilter
0 read(s) filtered by: MappingQualityAvailableReadFilter
0 read(s) filtered by: MappedReadFilter
57800 read(s) filtered by: NotSecondaryAlignmentReadFilter
4295149 read(s) filtered by: NotDuplicateReadFilter
0 read(s) filtered by: PassesVendorQualityCheckReadFilter
0 read(s) filtered by: WellformedReadFilter
6323292 total reads filtered

#变异检测

HaplotypeCaller命令

HaplotypeCaller的应用有两种做法，差别只在于要不要在中间生成一个gVCF：（1）直接进行HaplotypeCaller，这适合于单样本，或者那种固定样本数量的情况，也就是只执行一次HaplotypeCaller。如果增加一个样本就得重新运行这个HaplotypeCaller，而这个时候算法需要重新去读取所有人的BAM文件，浪费大量时间；（2）每个样本先各自生成gVCF，然后再进行群体joint-genotype。gVCF全称是genome VCF，是每个样本用于变异检测的中间文件，格式类似于VCF，它把joint-genotype过程中所需的所有信息都记录在这里面，文件无论是大小还是数据量都远远小于原来的BAM文件。这样一旦新增加样本也不需要再重新去读取所有人的BAM文件了，只需为新样本生成一份gVCF，然后重新执行这个joint-genotype就行了。这里用第一种。

```
cd aligned
gatk --java-options "-Xmx8G -Djava.io.tmpdir=../tmp/" HaplotypeCaller -R ../
```

```
gatk --java-options "-Xmx8G -Djava.io.tmpdir=../tmp/" HaplotypeCaller -R ../

gatk --java-options "-Xmx8G -Djava.io.tmpdir=../tmp/" HaplotypeCaller -R ../

#new version1
gatk --java-options "-Xmx8G -Djava.io.tmpdir=../tmp/" HaplotypeCaller -R /hd
```

8.3 变异质控与过滤

质控的含义和目的是指通过一定的标准，最大可能地剔除假阳性的结果，并尽可能地保留最多的正确数据。 第一种方法 GATK VQSR，它通过机器学习的方法利用多个不同的数据特征训练一个模型（高斯混合模型）对变异数据进行质控，使用VQSR需要具备以下两个条件： 第一，需要一个精心准备的已知变异集，它将作为训练质控模型的真集。比如，Hapmap、OMNI，1000G和dbsnp等这些国际性项目的数据，这些可以作为高质量的已知变异集。 第二，要求新检测的结果中有足够多的变异，不然VQSR在进行模型训练的时候会因为可用的变异位点数目不足而无法进行。
此方法要求新检测的结果中有足够多的变异，不然VQSR在进行模型训练的时候会因为可用的变异位点数目不足而无法进行。可能很多非人的物种在完成变异检测之后没法使用GATK VQSR的方法进行质控，一些小panel、外显子测序，由于最后的变异位点不够，也无法使用VQSR。全基因组分析或多个样本的全外显子组分析适合用此方法。

我们样本较少，因此选用第二种

第二种方法通过过滤指标过滤。 QualByDepth（QD）：QD是变异质量值（Quality）除以覆盖深度（Depth）得到的比值。 FisherStrand (FS)：FS是一个通过Fisher检验的p-value转换而来的值，它要描述的是测序或者比对时对于只含有变异的read以及只含有参考序列碱基的read是否存在着明显的正负链特异性（Strand bias，或者说是差异性） StrandOddsRatio (SOR)：对链特异（Strand bias）的一种描述.
RMSMappingQuality (MQ)：MQ这个值是所有比对至该位点上的read的比对质量值的均方根. MappingQualityRankSumTest (MQRankSum) ReadPosRankSumTest (ReadPosRankSum) **通过过滤指标过滤**

VariantFiltration:
Filter variant calls based on INFO and/or FORMAT annotations
This tool is designed for hard-filtering variant calls based on certain criteria. Records are hard-filtered by changing the value in the FILTER field to something other than PASS. Filtered records will be preserved in the output unless their removal is requested in the command line.
Inputs

- A VCF of variant calls to filter.
- One or more filtering expressions and corresponding filter names.

Output
A filtered VCF in which passing variants are annotated as PASS and failing variants are annotated with the name(s) of the filter(s) they failed.

```
1
使用SelectVariants，选出SNP
gatk SelectVariants -select-type SNP -V SRR10059490_wes.raw.vcf -O SRR100594

gatk SelectVariants -select-type SNP -V SRR10059491_wes.raw.vcf -O SRR100594

gatk SelectVariants -select-type SNP -V SRR10059492_wes.raw.vcf -O SRR100594
# 为SNP作过滤
gatk VariantFiltration -V SRR10059490_wes.snp.vcf --filter-expression "QD <

gatk VariantFiltration -V SRR10059491_wes.snp.vcf --filter-expression "QD <

gatk VariantFiltration -V SRR10059492_wes.snp.vcf --filter-expression "QD <

2
使用SelectVariants，选出Indel
gatk SelectVariants -select-type INDEL -V SRR10059490_wes.raw.vcf -O SRR1005

gatk SelectVariants -select-type INDEL -V SRR10059491_wes.raw.vcf -O SRR1005

gatk SelectVariants -select-type INDEL -V SRR10059492_wes.raw.vcf -O SRR1005

# 为Indel作过滤
gatk VariantFiltration -V SRR10059490_wes.indel.vcf --filter-expression "QD

gatk VariantFiltration -V SRR10059491_wes.indel.vcf --filter-expression "QD

gatk VariantFiltration -V SRR10059492_wes.indel.vcf --filter-expression "QD

## new version1
使用SelectVariants，选出SNP
gatk SelectVariants -select-type SNP -V father.raw.vcf -O father.snp.vcf

为SNP作过滤
gatk VariantFiltration -V mother.snp.vcf --filter-expression "QD < 2.0 || MQ

2
使用SelectVariants，选出Indel
gatk SelectVariants -select-type INDEL -V mother.raw.vcf -O mother.indel.vcf

为Indel作过滤
gatk VariantFiltration -V father.indel.vcf --filter-expression "QD < 2.0 ||

```

#q: filter的具体数值意义是否要明确
#warning:
15:55:13.094 WARN JexlEngine – ![0,14]: 'ReadPosRankSum < -8.0;' undefined variable ReadPosRankSum

15:55:13.094 WARN JexlEngine – ![0,9]: 'MQRankSum < –12.5;' undefined variable MQRankSum

你的vcf文件中，有的行INFO那一列没有“MQRankSum” or ”ReadPosRankSum”信息，所以才会出现这样的警告，
可以用“grep –v MQRankSum your_vcf”查找出来，这不是运行错误，可以不管继续进行后续分析；
若想要去除这个警告可以这样做: https://www.jianshu.com/p/1014f65cde3f

9 ANNOVAR

突变注释

ANNOVAR是一个perl编写的命令行工具，能在安装了perl解释器的多种操作系统上执行。允许多种输入文件格式，包括最常被使用的VCF格式。输出文件也有多种格式，包括注释过的VCF文件、用tab或者逗号分隔的txt文件，ANNOVAR能快速注释遗传变异并预测其功能。这个软件需要edu邮箱注册才能下载。 ANNOVAR website

#1 download ANNOVAR
官网注册下载（edu邮箱）http://www.openbioinformatics.org/annovar/annovar_download_form.php
#2 download database
https://annovar.openbioinformatics.org/en/latest/user-guide/download/#:~:text=Many%20of%20the%20databases%20that%20ANNOVAR%20uses%20can,-webfrom%20annovarin%20ANNOVAR%20directly%20to%20download%20these%20databases.

```
perl annotate_variation.pl -buildver hg19 -downdb -webfrom annovar refGene h

 # -buildver 表示version
 # -downdb 下载数据库的指令
 # -webfrom annovar 从annovar提供的镜像下载，不加此参数将寻找数据库本身的源
 # humandb/ 存放于humandb/目录下

perl annotate_variation.pl --buildver hg19 --downdb gwascatalog humandb/ (do
perl annotate_variation.pl --buildver hg19 --downdb ljb26_all --webfrom annc
perl annotate_variation.pl --buildver hg19 --downdb esp6500siv2_ea --webfrom
perl annotate_variation.pl --buildver hg19 --downdb esp6500siv2_all --webfrc
perl annotate_variation.pl --buildver hg19 --downdb --webfrom annovar 1000g2
perl annotate_variation.pl --buildver hg19 --downdb --webfrom annovar 1000g2

perl annotate_variation.pl --buildver hg19 --downdb cytoBand humandb/ (done)
perl annotate_variation.pl --buildver hg19 --downdb avsift -webfrom annovar
perl annotate_variation.pl --buildver hg19 --downdb snp138 humandb/ (done)
perl annotate_variation.pl --buildver hg19 --downdb genomicSuperDups humandb
perl annotate_variation.pl --buildver hg19 --downdb phastConsElements46way h
perl annotate_variation.pl --buildver hg19 --downdb tfbs humandb/ (done)
```

三种注释方式

Gene-based Annotation(基于基因的注释)：基于基因的注释（gene-based annotation）揭示variant与已知基因直接的关系以及对其产生的功能性影响。 **Region-based Annotation（基于区域的注释）**：基于过滤的注释精确匹配查询变异与数据库中的记录：如果它们有相同的染色体，起始位置，结束位置，REF的等位基因和ALT的等位基因，才能认为匹配。基于区域的注释看起来更像一个区域的查询（这个区域也可以是一个单一的位点），在一个数据库中，它不在乎位置的精确匹配，它不在乎核苷酸的识别。基于区域的注释（region-based annotation）揭示variant与不同基因组特定段的关系。 **Filter-based Annotation（基于过滤的注释）**：filter-based和region-based主要的区别是，filter-based针对mutation（核苷酸的变化）而region-based针对染色体上的位置。如在全基因组数据中的变异频率，可使用1000g2015aug、kaviar_20150923等数据库；在全外显组数据中的变异频率，可使用exac03、esp6500siv2等；在孤立的或者低代表人群中的变异频率，可使用ajews等数据库。 **用table_annovar.pl进行注释，可一次性完成三种类型的注释。**

avinput文件由tab分割，最重要的地方为前5列，分别是: 1. 染色体(Chromosome) 2. 起始位置(Start) 3. 结束位置(End) 4. 参考等位基因(Reference Allele) 5. 替代等位基因(Alternative Allele) ANNOVAR主要也是利用前五列信息对数据库进行比对，注释变异。

SNP注释

https://annovar.openbioinformatics.org/en/latest/user-guide/input/#annovar-input-file

```
nohup annovar/convert2annovar.pl -format vcf4 aligned/10filtered/SRR10059490
nohup annovar/table_annovar.pl aligned/11snp_annotation/SRR10059490_snp.avir

The same for 91, 92

```

-buildver hg19 表示使用hg19版本
-out snpanno 表示输出文件的前缀为snpanno
-remove 表示删除注释过程中的临时文件
-protocol 表示注释使用的数据库，用逗号隔开，且要**注意顺序**
-operation 表示对应顺序的数据库的类型（g代表gene-based、r代表region-based、f代表filter-based），用逗号隔开，**注意顺序**
-nastring . 表示用点号替代缺省的值
-csvout 表示最后输出.csv文件。

#加入两个database（重做）

```
nohup annovar/convert2annovar.pl -format vcf4 --includeinfo -withzyg align
ed/10filtered/SRR10059490_wes.snp.filter.vcf > aligned/11_2_snp_annotatio
n/SRR10059490_snp.avinput &

nohup annovar/table_annovar.pl aligned/11_2_snp_annotation/SRR10059490_sn
p.avinput annovar/humandb -buildver hg19 -out aligned/11_2_snp_annotation/
SRR10059490_snpanno -remove -protocol refGene,cytoBand,genomicSuperDups,sn
p138,1000g2015aug_all,esp6500siv2_all -operation g,r,r,f,f,f -nastring . -
csvout &

The same for 91, 92

```

```
#new version1
/home/jiajun/putaotai/annovar/convert2annovar.pl -format vcf4 mother.snp.filter.vcf > ../6snp_annotation/mother_snp.avinput
/home/jiajun/putaotai/annovar/table_annovar.pl ../6snp_annotation/mother_snp.avinput /home/jiajun/putaotai/annovar/humandb -buildver hg19 -out ../6snp_annotation/mother_snpanno -remove -protocol refGene,cytoBand,genomicSuperDups,snp138,1000g2015aug_all,esp6500siv2_all -operation g,r,r,f,f,f -nastring . -csvout
```

- If you need the zygosity, quality and read coverage information in the output line as well, add the `-withzyg` argument (convert2annovar.pl)

Indel注释

```
nohup annovar/convert2annovar.pl -format vcf4 aligned/10filtered/SRR10059492
nohup annovar/table_annovar.pl aligned/11indel_annotation/SRR10059492_indel.

The same for 91, 92
```

#加入两个database（重做）

```
nohup annovar/convert2annovar.pl -format vcf4 --includeinfo -withzyg aligned/10filtered/SRR10059490_wes.indel.filter.vcf > aligned/11_2_indel_annotation/SRR10059490_indel.avinput &

nohup annovar/table_annovar.pl aligned/11_2_indel_annotation/SRR10059490_indel.avinput annovar/humandb -buildver hg19 -out aligned/11_2_indel_annotation/SRR10059490_indelanno -remove -protocol refGene,cytoBand,genomicSuperDups,snp138,1000g2015aug_all,esp6500siv2_all -operation g,r,r,f,f,f -nastring . -csvout &

The same for 91, 92

nohup annovar/convert2annovar.pl -format vcf4 --includeinfo -withzyg aligned/10filtered/SRR10059492_wes.indel.filter.vcf > aligned/11_2_indel_annotation/SRR10059492_indel.avinput &

nohup annovar/table_annovar.pl aligned/11_2_indel_annotation/SRR10059492_indel.avinput annovar/humandb -buildver hg19 -out aligned/11_2_indel_annotation/SRR10059492_indelanno -remove -protocol refGene,cytoBand,genomicSuperDups,snp138,1000g2015aug_all,esp6500siv2_all -operation g,r,r,f,f,f -nastring . -csvout &

#new version1

/home/jiajun/putaotai/annovar/convert2annovar.pl -format vcf4 mother.indel.filter.vcf > ../7indel_annotation/mother_indel.avinput
/home/jiajun/putaotai/annovar/table_annovar.pl ../7indel_annotation/mother_indel.avinput /home/jiajun/putaotai/annovar/humandb -buildver hg19 -out ../7indel_annotation/mother_indelanno -remove -protocol refGene,cytoBand,
```

```
genomicSuperDups,snp138,1000g2015aug_all,esp6500siv2_all -operation g,r,r,
f,f,f -nastring . -csvout
```

review：https://www.bilibili.com/video/BV15s411P7ay?p=17&spm_id_from=pageDriver
样本间变异分析比较
ANNOVAR结果说明-SNP/INDEL：https://www.jianshu.com/p/6c11fe689bac

10 CNV analysis (using gatk)

ref:
https://zhuanlan.zhihu.com/p/66034477
https://blog.csdn.net/yuqiuwang929/article/details/105770989

10.1 PreprocessIntervals

前期准备：目标区域文件格式 & 计算reads count

10.1.1 PreprocessIntervals 窗口划分

对bins进行前期处理以用来计算reads coverage，首先检查输入的interval是否有overlap，有则合并；然后根据指定参数扩充interval，分成bins,按指定bin长切割bins，最后过滤掉都是N的bins

For exome data To pad intervals by 250 bases and disable binning (e.g., for targeted analyses):

```
 cd aligned/
 mkdir 12cnv
../gatk-4.2.0.0/gatk BedToIntervalList \
          -I bed_from_UCSC_simplied.bed \
          -O Exon.interval.interval_list \
          -SD ../genome/hg19.dict

 gatk --java-options "-Xmx20G" PreprocessIntervals \
          -R /home/zhiwen/genome/hg19.fa \
          -L /home/zhiwen/aligned/Exon.interval.interval_list \
          --bin-length 0 \
          --padding 250 \
          -imr OVERLAPPING_ONLY \
          -O preprocessed_intervals.interval_list

#cjj
gatk=/home/jiajun/gatk-4.2.0.0/gatk
ref=/home/jiajun/genome/hg19.fa
ref_dict=/home/jiajun/genome/hg19.dict
interval=/home/jiajun/putaotai/aligned/Exon.interval.interval_list

# --bin-length 为你的窗口大小，-imr OVERLAPPING_ONLY 意思为捕获区间有重叠就会合并
```

```
$gatk PreprocessIntervals -R $ref -L $interval --bin-length 5000 -imr OVERLA
```

error: java.lang.IllegalArgumentException: Interval merging rule must be set to OVERLAPPING_ONLY.

solution: add -imr OVERLAPPING_ONLY \

ref: https://gatk.broadinstitute.org/hc/en-us/articles/360035531152--How-to-Call-common-and-rare-germline-copy-number-variants

这里有个问题--bin-length的取值是0还是5000?

--bin-length 为你的窗口大小

应该是5000吧

10.1.2 计算样本每个窗口的reads (见10.1.5)

```
gatk --java-options "-Xmx20G" CollectReadCounts \
     -L preprocessed_intervals.interval_list \
     -R /home/zhiwen/genome/hg19.fa \
     -imr OVERLAPPING_ONLY \
     -I /home/zhiwen/aligned/3sorted_bam/SRR10059490_wes.sorted.bam \
     --format TSV \
     -O SRR10059490_wes.tsv

#cjj
input_bam=/home/jiajun/putaotai/aligned/version1/3sortedbam/mole_v.sorted.bam
$gatk CollectReadCounts -L targets.preprocessed.5000.interval_list -R $ref -imr OVERLAPPING_ONLY -I $input_bam --format TSV -O ./mole_v.sorted_reads.tsv

input_bam=/home/jiajun/putaotai/aligned/version1/3sortedbam/normal_v.sorted.bam
$gatk CollectReadCounts -L targets.preprocessed.5000.interval_list -R $ref -imr OVERLAPPING_ONLY -I $input_bam --format TSV -O ./normal_v.sorted_reads.tsv

input_bam=/home/jiajun/putaotai/aligned/version1/3sortedbam/normal_b.sorted.bam
$gatk CollectReadCounts -L targets.preprocessed.5000.interval_list -R $ref -imr OVERLAPPING_ONLY -I $input_bam --format TSV -O ./normal_b.sorted_reads.tsv


```

error: A USER ERROR has occurred: Traversal by intervals was requested but some input files are not indexed. Please index all input files:

solution: 对原始bam文件进行index

10.1.3 窗口文件添加GC信息

```
gatk AnnotateIntervals \
```

```
 -R /home/jiajun/genome/hg19.fa \
 -L preprocessed_intervals.interval_list \
 --interval-merging-rule OVERLAPPING_ONLY \
 -O annotated_intervals.tsv &

#cjj
$gatk AnnotateIntervals -L targets.preprocessed.5000.interval_list -R $ref -
```

java.lang.IllegalArgumentException: Interval padding must be set to 0.
不知道为什么ip 不能设置为40或者其他

*分析外显子组时，有时需分析目标位点franking区域，可通过--interval-padding/-ip来实现
比如-L chr1:100 -ip 20 将把目标区域扩充到chr1:80-120

*分析外显子组时，有时需排除目标位点franking区域，可通过--interval-exclusion-padding/-ixp来实现
比如-XL chr1:100 -ixp 20 则排除区域chr1:80-120(原来只排除位点100)

10.1.4 剔除GC异常以及reads数偏多/偏少的窗口

不明白这两种有什么区别？也可以和在一起用？
https://gatk.broadinstitute.org/hc/en-us/articles/360036725951-FilterIntervals

```
$gatk FilterIntervals
-L targets.preprocessed.5000.interval_list
--annotated-intervals targets.preprocessed.5000.annotated.tsv $sample_rc
-imr OVERLAPPING_ONLY
-O gc.filtered.bin5000.interval_list

（目前用的是这版本）
gatk FilterIntervals \
-L preprocessed_intervals.interval_list \
-I normal_b.counts.hdf5 \
-I normal_v.counts.hdf5 \
-I mole_v.counts.hdf5 \
-imr OVERLAPPING_ONLY \
--annotated-intervals annotated_intervals.tsv \
-O filtered_intervals.interval_list &

#cjj
$gatk FilterIntervals -L targets.preprocessed.5000.interval_list --annotated
```

10.1.5 CollectReadCounts

计算指定的intervals的reads数，即计算起始位点落入intervals的reads数。

```
gatk CollectReadCounts \
-I ./3sortedbam/normal_b.sorted.bam \
```

```
-L ./9cnv/preprocessed_intervals.interval_list \
--interval-merging-rule OVERLAPPING_ONLY \
-O ./9cnv/normal_b.counts.hdf5 &

```

(Optional) CollectAllelicCounts 目前没做
计算指定参考基因组和指定位点的allele depth，只计算满足参数要求的reads depth(通过reads filter条件并大于指定的minimum-base-quality)

```
gatk CollectAllelicCounts \
-I sample.bam \
-R reference.fa \
-L sites.interval_list \
-O sample.allelicCounts.tsv
```

10.1.6 `DetermineGermlineContigPloidy`

根据先验值确定倍性

```
gatk --java-options "-Xmx20G" DetermineGermlineContigPloidy \
-L preprocessed_intervals.interval_list \
--interval-merging-rule OVERLAPPING_ONLY \
--input normal_b.counts.hdf5 \
--input normal_v.counts.hdf5 \
--contig-ploidy-priors ploidy_priors_table.tsv \
--output ploidy-calls \
--output-prefix ploidy &

#cjj
$gatk DetermineGermlineContigPloidy -L molev.filtered.bin5000.interval_list

```

error: A required Python package ("gcnvkernel") could not be imported into the Python environment. This tool requires that the GATK Python environment is properly established and activated. Please refer to GATK README.md file for instructions on setting up the GATK Python environment.
重新安装了gatk环境

```
cd gatk-4.2.0.0/
conda env create -n gatk -f gatkcondaenc.yml
conda activate gatk
conda install gatk
```

ref:
https://gatk.broadinstitute.org/hc/en-us/articles/360035889851
https://gatk.broadinstitute.org/hc/en-us/articles/360036349252-DetermineGermlineContigPloidy
https://bioconda.github.io/recipes/gatk4/README.html

重新跑，无法理解的error就出现了
'Sample "{0}" already has coverage metadata annotations'.format(sample_name))

gcnvkernel.structs.metadata.SampleAlreadyInCollectionException: Sample ”wes” already has coverage metadata annotations

我也

10.2 germline CNV calling

COHORT mode

```
gatk GermlineCNVCaller \
   --run-mode COHORT \
   -L filtered_intervals.interval_list \
   --interval-merging-rule OVERLAPPING_ONLY \
   --contig-ploidy-calls ./ploidy-calls/ \
   --input normal_b.counts.hdf5 \
   --input normal_v.counts.hdf5 \
   --output output_dir \
   --output-prefix normal_cohort_run
```

Stderr: usage: cohort_denoising_calling.7550592232649395138.py [-h]

最后处理，输入hdf.5文件，输出vcf文件，最后用igv看

```
gatk PostprocessGermlineCNVCalls \
```