# SNP-calling

## FASTQ文件

### 1来源：

产生自测序仪的原始测序数据，文件大小依照不同的测序量(或测序深度)而有很大差异，文件后缀通常都是.fastq，.fq或者.fq.gz(gz压缩)。

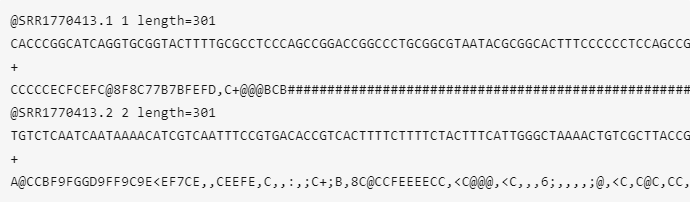
### 2构成：

每四行成为一个独立的单元，我们称之为read。

**第一行：**以‘@’开头，是这一条read的名字，这个字符串是根据测序时的状态信息转换过来的，中间不会有空格，它是每一条read的唯一标识符。

**第二行：**测序read的序列，N代表的是测序时那些无法被识别出来的碱基。

**第三行：**以“+”开始，可以储存一些附加信息，一般是空的。

**第四行：**测序read的质量值，它描述的是每个测序碱基的可靠程度，用ASCII码表示，越大说明测序的质量越好。*公式：*Q = -10log(p\_error)。即，**质量值是测序错误率的对数(10为底数)乘以-10(并取整)**。比如，如果该碱基的测序错误率是0.01，那么质量值就是20(俗称Q20)。为什么要用ASCII码来代表，直接用数字不行吗？行！但很难看 。并非所有的ASCII码都是可见的字符，为了能够让碱基的质量值表达出来，必须避开所有这些不可见字符。最简单的做法就是加上一个固定的整数(33)！我们把加33的的质量值体系称之为Phred33。

## 1原始数据质量控制

**质控的意义**：以illumina为首的二代测序基本都是运用边合成边测序的技术。碱基的合成依靠的是化学反应，这使得碱基链可以不断地从5'端一直往3'端合成并延伸下去。但在这个合成的过程中随着合成链的增长，DNA聚合酶的效率会不断下降，特异性也开始变差，这就会带来一个问题——越到后面碱基合成的错误率就会越高。而测序数据的质量好坏会影响我们的下游分析。

一般我们可以从如下几个方面来分析：

**·read各个位置的碱基质量值分布**

好的结果是碱基质量值都基本都在大于30，而且波动很小。

**·碱基的总体质量值分布**

对于二代测序，最好是达到Q20的碱基要在95%以上(最差不低于90%)，Q30要求大于85%(最差也不要低于80%)。

**·read各个位置上碱基分布比例**，目的是为了分析碱基的分离程度

最好平均在1%以内。

**·GC含量分布**

对于人类来说，我们基因组的GC含量一般在40%左右。

**·read各位置的N含量**

N在测序数据中一般是不应该出现的。

**·read是否还包含测序的接头序列**

在测序之前需要构建测序文库，测序接头就是在这个时候加上的，其目的一方面是为了能够结合到flowcell上，另一方面是当有多个样本同时测序的时候能够利用接头信息进行区分。一般的WGS测序是不会测到这些接头序列的。

**·read重复率**，这个是实验的扩增过程所引入的

**fastp软件**可以很轻松地实现上述功能。除了质控报告，fastq还兼具过滤，接头处理，滑窗质量剪裁，PE数据的碱基校正，全局剪裁，polyG剪裁，分子标签UMI处理，输出文件切分等功能。

**Trimmomatic**

## 2read比对，排序和去除重复序列

### 序列比对

NGS测序下来的短序列(read)存储于FASTQ文件里面，FASTQ文件中紧挨着的两条read之间没有任何位置关系，它们都是随机来自于原本基因组中某个位置的短序列而已。因此，我们需要先把这一大堆的短序列捋顺，一个个去跟该物种的参考基因组比较，找到每一条read在参考基因组上的位置，然后按顺序排列好，这个过程就称为测序数据的比对。

基因组使用**BWA软件**进行比对，转录组推荐使用**STAR**或者**HISAT2**软件

过程大致是：参考基因组建索引——比对

### 加头排序

FASTQ文件里面这些被测序下来的read是随机分布于基因组上面的，第一步的比对是按照FASTQ文件的顺序把read逐一定位到参考基因组上之后，随即就输出了，它不会也不可能在这一步里面能够自动识别比对位置的先后位置重排比对结果。因此，比对后得到的结果文件中，每一条记录之间位置的先后顺序是乱的，我们后续去重复等步骤都需要在比对记录按照顺序从小到大排序下来才能进行，所以这才是需要进行排序的原因。

**picard.jar AddOrReplaceReadGroup模块加头 samtools软件排序并转为bam文件**

### 标记PCR重复

构建测序文库时细胞量的不足和打断过程中引起的部分降解，会使整体或者局部的DNA浓度过低。如果直接取样很可能会漏掉原本基因组上的一些DNA片段，导致测序不全。PCR扩增原本的目的就是为了增大微弱DNA序列片段的密度，但由于整个反应都在一个试管中进行，因此其他一些密度并不低的DNA片段也会被同步放大，在取样上机时，这些DNA片段就很可能被重复取到相同的几条去进行测序。最直接的后果就是同时增大了变异检测结果的假阴和假阳率。主要原因有：①DNA打断的那一步会发生一些损失，主要表现是会引发一些碱基发生颠换变换(嘌呤变嘧啶或者反之)，PCR扩大了这个信号。②PCR过程中也会带来新的碱基错误。③对于真实的变异，PCR反应可能会对包含某一个碱基的DNA模版扩增更加剧烈(这个现象称为PCR Bias)。因此，如果反应体系是对含有reference allele的模板扩增偏向强烈，那么变异碱基的信息会变小，从而会导致假阴。

PCR对真实的变异检测和个体的基因型判断都有不好的影响。GATK、Samtools、Platpus等这种利用贝叶斯原理的变异检测算法都是认为所用的序列数据都不是重复序列(即将它们和其他序列一视同仁地进行变异的判断，所以带来误导)，因此必须要进行标记(去除)或者使用PCR-Free的测序方案。

**gatk的MarkDuplicates模块**

### 创建比对索引文件

为标记重复后的bam文件创建索引文件，它的作用能够让我们可以随机访问这个文件中的任意位置

**samtools软件**

## 3变异检测

这是目前所有WGS数据分析流程的一个目标——获得样本准确的变异集合。这里变异检测的内容一般会包括：SNP、Indel，CNV和SV等，这个流程中我们只做其中最主要的两个：SNP和Indel。

**GATK HaplotypeCaller模块**

它会先推断群体的单倍体组合情况，计算各个组合的几率，然后根据这些信息再反推每个样本的基因型组合。因此它不但特别适合应用到群体的变异检测中，而且还能够依据群体的信息更好地计算每个个体的变异数据和它们的基因型组合。

一般来说，在实际的WGS流程中对**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，也就是执行一次HaplotypeCaller之后就老死不相往来了。否则你会碰到仅仅只是增加一个样本就得重新运行这个HaplotypeCaller的坑爹情况(即，N+1难题)，而这个时候算法需要重新去读取所有人的BAM文件，这将会是一个很费时间的痛苦过程；

(2) **每个样本先各自生成gVCF**，然后再进行群体joint-genotype。这其实就是GATK团队为了解决(1)中的N+1难题而设计出来的模式。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。

推荐选择第二种方式，变异检测不是一个样本的事情，有越多的同类样本放在一起joint calling结果将会越准确，而如果样本足够多的话，在低测序深度的情况下也同样可以获得完整并且准确的结果，而这样的分步方式是应对多样本的好方法

## 4变异结果质控和过滤

也就是俗称的**硬过滤**

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。首选的质控方案是GATK VQSR(Variant Quality Score Recalibration)，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。

一般来说SNP和INDEL的过滤标准不同，得分开做

SNP："QD < 2.0 || MQ < 40.0 || FS > 60.0 || SOR > 3.0 || MQRankSum < -12.5 || ReadPosRankSum < -8.0"

INDEL："QD < 2.0 || MQ < 40.0 || FS > 200.0 || SOR > 10.0 || MQRankSum < -12.5 || ReadPosRankSum < -8.0"

**- QualByDepth（QD）**

QD是变异质量值(Quality)除以覆盖深度(Depth)得到的比值。这里的变异质量值就是VCF中QUAL的值——用来衡量变异的可靠程度，这里的覆盖深度是这个位点上所有含有变异碱基的样本的覆盖深度之和，通俗一点说，就是这个值可以通过累加每个含变异的样本（GT为非0/0的样本）的覆盖深度（VCF中每个样本里面的DP）而得到。

**- 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）。这个差异反应了测序过程不够随机，或者是比对算法在基因组的某些区域存在一定的选择偏向。如果测序过程是随机的，比对是没问题的，那么不管read是否含有变异，以及是否来自基因组的正链或者负链，只要是真实的它们就都应该是比较均匀的，也就是说，不会出现链特异的比对结果，FS应该接近于零。

**- StrandOddsRatio (SOR)**

SOR原理和FS类似，但是用了不同的统计检验方法计算差异程度，用的是symmetric odds ratio test，它更适合于高覆盖度的数据。

**- 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(先平方、再平均、然后开方)。它和平均值相比更能够准确地描述比对质量值的离散程度。

**- MappingQualityRankSumTest (MQRankSum)**

对杂合位点进行的不同碱基之间比对质量的曼惠特尼秩和检验结果，通过ref和alt碱基的比对质量差异来评估位点的可信度。

**- ReadPosRankSumTest (ReadPosRankSum)**

对杂合位点进行的秩和检验，看不同的碱基是否倾向于出现在reads上的特定位置(例如接近reads的起始或者终止)。

# 基因组组装

## contig和scaffold的含义：

比如一个基因组大小是1M，测序得到若干条reads，这些reads进行拼接，如果完全可以拼接起来，中间没有gap的序列称为contig，即连续的意思。

如果中间有gap,但是可以知道gap的长度，这样的序列就叫做scaffold, 即脚手架（非连续）的意思。

然后把contig 和 scaffold 从长到短进行排列，然后相加，当恰好加到1M的50％，也就是500k的时候 ，那一条 contig 或者scaffold 的长度就叫做Contig N50和Scaffold N50。很明显这个数值越大说明组装的质量越好。

即：从最长的开始倒数，数到长度为总长度一半的片段，最后一个被数到的片段越长，说明长的片段越多，最后组装的质量越好。

N90：把50%改为90%即可。