5.1.1 常用生物信息学格式介绍

rabbit gao's blog 2015-07-10

前言

在各个行业都是有行业标准的，这样才能统一规范而方便后面的分析，在生物信息学领域中主要是各种大量序列数据、注释数据等，这些都是有特定的格式去表示，下面列举几种常见的格式。了解这些是进行后续生物信息学分析的必备知识，有些人虽说是在做生物信息学分析，但是到现在可能还不知道什么是GFF3格式等。

---

fasta

fasta格式是最基本的表示序列信息（核苷酸或者蛋白质）的格式。
http://genetics.bwh.harvard.edu/pph/FASTA.html ， https://en.wikipedia.org/wiki/FASTA_format 。这里简单介绍下，fasta格式的文件通常后缀名为.fasta 或者.fa， 其实这都无所谓，因为都是文本文件。fasta格式文件（可以包含多条序列）中的一条序列的通常表示方法如下：

```
>gi|129295|sp|P01013|OVAX_CHICK GENE X PROTEIN (OVALBUMIN-RELATED)
QIKDLLVSSSTDLDTTLVLVNAIYFKGMWKTAFNAEDTREMPFHVTKQESKPVQMMCMNNSFNVATLPAE
KMKILELPFASGDLSMLVLLPDEVSDLERIEKTINFEKLTEWTNPNTMEKRRVKVYLPQMKIEEKYNLTS
VLMALGMTDLFIPSANLTGISSAESLKISQAVHGAFMELSEDGIEMAGSTGVIEDIKHSPESEQFRADHP
FLFLIKHNPTNTIVYFGRYWSP
```

其中主要分为两个部分：

- 第一部分是序列的定义行（单行），该行的开头是>符号，紧跟着后面的就是该条序列的名称（具有唯一性，即不能和其它序列同名称），即>号和后面的名称的第一字符间是没有任何空白的。一般第一个空格后面的内容即为可选的描述信息。如上面， gi|129295|sp|P01013|OVAX_CHICK为序列名称， 而GENE X PROTEIN (OVALBUMIN-RELATED)则为描述信息。注意：有点软件是把一整行当做名称的，所以在出现错误的时候可以查看下格式是否正确。
- 第二部分就是序列，所有的序列碱基或者氨基酸可以都放在一行存储，也可以多行存储，但是建议大家多行存储且单行长度不超过80个字符，因为这样容易阅读。且序列的多行之间不能有空行，序列信息描述的第一行与序列数据的第一行之间不能有空行。其中序列数据主要是按照密码表来表示的，*表示是蛋白质翻译的结束。

多行序列举例如下：

```
>SEQUENCE_1
MTEITAAMVKELRESTGAGMMDCKNALSETNGDFDKAVQLLREKGLGKAAKKADRLAAEG
LVSVKVSDDFTIAAMRPSYLSYEDLDMTFVENEYKALVAELEKENEERRRLKDPNKPEHK
IPQFASRKQLSDAILKEAEEKIKEELKAQGKPEKIWDNIIPGKMNSFIADNSQLDSKLTL
MGQFYVMDDKKTVEQVIAEKEKEFGGKIKIVEFICFEVGEGLEKKTEDFAAEVAAQL
>SEQUENCE_2
SATVSEINSETDFVAKNDQFIALTKDTTAHIQSNSLQSVEELHSSTINGVKFEEYLKSQI
ATIGENLVVRRFATLKAGANGVVNGYIHTNGRVGVVIAAACDSAEVASKSRDLLRQICMH
```

---

fastq

fastq（ https://en.wikipedia.org/wiki/FASTQ_format ）同样是以文本形式来存储序列信息的格式，后缀名通常为.fastq 或者.fq，但是与fasta不相同的是，它除了存储序列本身外还存储了序列中每个单元所对应的质量分数，所以fastq格式通常用于高通量测试数据的存储。早期是有Sanger机构开发的，但是现在已经演变成一个高通量测序的标准了。

fastq格式文件中一个完整的单元分为四行，每行的含义如下：

第一行： 以@开头，内容同fasta的描述行类似

第二行：具体的碱基序列

第三行：以+开头，后面的内容可以和第一行类似，也什么都没有只留+

第四行：以ASCII字符集（分数）编码来表示对应碱基的测序质量

比如下面的这个例子：

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

下面以Illumina和NCBI SRA两个测序数据来源来讲讲它们之间的区别：

通常我们获取测序数据有两种途径，一种是自己通过仪器测定，一种是在公共数据库中(比如之前说到的NCBI中的SRA数据库)获取，这两种方式主要是在序列名称的命名上和测序质量表示方式上有所不同。

Illumina 序列名称：

```
@HWUSI-EAS100R:6:73:941:1973#0/1
```

上述以：隔开的每个字段的含义如下：

HWUSI-EAS100R	the unique instrument name
6	flowcell lane
73	tile number within the flowcell lane
941	‘x’-coordinate of the cluster within the tile
1973	‘y’-coordinate of the cluster within the tile
#0	index number for a multiplexed sample (0 for no indexing)

/1	the member of a pair, /1 or /2 *(paired-end or mate-pair reads only)*

NCBI SRA数据库：

将测序数据提交到NCBI的SRA数据库时，SRA数据库会为每一个样本提供一个编号，一般是SRRxxxxx，所以从SRA数据库上下载公共的测试数据（原始格式为

.sra， 需特定工具转换为fastq），其fastq格式文件中每个单元的名称是以SRA编号接数字加以区分的。比如下面的这个示例：

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
length=36
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345
length=36
IIIIIIIIIIIIIIIIIIIIIIIIIIIIII9IG9IC
```

- 需要注意的是：当把测序数据上传到SRA数据库时，它通常会将表示质量的分数 **转换为标准的Sanger格式** 。

质量分数表示法：

由于测序仪器的不同等因素所以对碱基测序质量的表示方式也不相同，在Fastq格式文件中，用ASCII码表来表示每个碱基的测序质量，下面介绍几种不同的方案：

```
  SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
  ..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....................
  ...............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII.......................
  ...............................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ.......................
  LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL.....................................................
  !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
  |                         |    |        |                              |                     |
 33                        59   64       73                            104                   126
  0........................26...31.......40
                           -5....0........9.............................40
                                 0........9.............................40
                                    3.....9.............................40
  0.2......................26...31.......41

 S - Sanger        Phred+33,  raw reads typically (0, 40)
 X - Solexa        Solexa+64, raw reads typically (-5, 40)
 I - Illumina 1.3+ Phred+64,  raw reads typically (0, 40)
 J - Illumina 1.5+ Phred+64,  raw reads typically (3, 40)
     with 0=unused, 1=unused, 2=Read Segment Quality Control Indicator (bold)
     (Note: See discussion above).
 L - Illumina 1.8+ Phred+33,  raw reads typically (0, 41)
```

其中有五种表示方法，Sanger的码表范围为!至I，其对应的数值为33-73，如果减去33（即Phred+33表示法）这个基数则范围转换为0-40，即如果某一个碱基的测序质量为!则对应的测序质量分数为0，表示测序质量低。其它几种表示法类似（X,I,J,L）。这里介绍测序质量的表示方法是因为后面有的软件是要指定测序数据的质量表示方法。

---

gff2

http://gmod.org/wiki/GFF2

GFF(General Feature Format)是一种用于描述基因或者其它序列元素的文件格式，GFF有几个版本，早期的第Version 2和现在的Version 3. Version 2 是由Sanger机构所制定的，而Version 3是由Sequence Ontology Project制定。正是由于有统一的格式来表示基因等元素，使得GFF格式的文件被广泛的使用与mapping与基因组数据可视化方面。

GFF2文件格式是由tab隔开的九列值，每一行的九个字段的含义如下：

```
Chr1  curated  CDS 365647  365963  .  +  1  Transcript
"R119.7"
```

第一列： reference sequence， 该列表示的是特征元素所在的染色体（或者scaffold，或者contig），也就是在基因组中的坐标系统，后续一切的注释信息都是基于此列。

第二列：source，该列表示改行注释信息的来源，比如上述的一行表示该行的CDS注释信息来自名为“curated”的注释。

第三列：feature，或者说是method，type， 表示的是该注释的类型，比如上述表示改行注释为CDS信息，可以将source和feature结合起来描述的更加详细。

第四列：start position，在reference sequence上的开始位置（坐标），通常是从1为起点而不是0。

第五列：end position， 在reference sequence上的结束位置（坐标），一般是大于start position的。

第六列：score， 表示该行feature的分数，比如序列相似性等，如果没有对应的分数可以用.代替。

第七列：strand，feature所在链，+表示正链，-表示负链，.表示不确定或者与链无关。

第八列：phase，与蛋白质编码相关，一般是用于CDS，值的范围为0-2，表示编码时阅读框的移动相位。

下面这段描述很详细:

‘0’ indicates that the specified region is in frame, i.e. that its first base corresponds to the first base of a codon. ‘1’ indicates that there is one extra base, i.e. that the second base of the region corresponds to the first base of a codon, and ‘2’ means that the third base of the region is the first base of a codon. If the strand is ‘-‘, then the first base of the region is value of <end>, because the corresponding coding region will run from <end> to <start> on the reverse strand.

第九列：group，或者称为attributes，是用于对改行注释更多的描述，以键值对的形式，比如上面的例子表示该CDS是属于名为R119.7的transcript。该列中可以存在多个属性，属性之间是用;隔开的。

对于GFF格式的理解主要是集中在最后一列，有以下集中情况：

1. 对于单个feature

```
Chr3   giemsa heterochromatin  4500000 6000000 . . .   Band
3q12.1
```

2. 对于属于同一集合的多个feature

```
IV     curated exon    5506900 5506996 . + .   Transcript
B0273.1
IV     curated exon    5506026 5506382 . + .   Transcript
B0273.1
IV     curated exon    5506558 5506660 . + .   Transcript
B0273.1
IV     curated exon    5506738 5506852 . + .   Transcript
B0273.1
```

比如上面这个例子就表示这四个exon都是属于同一个名为B0273.1的transcript，这是表示一个完整transcript结构的最基本要求。

GFF2还可用于序列比对结果表示等其它方面，这里不做介绍了。

---

gtf(gff2.5)

http://mblab.wustl.edu/GTF2.html

GTF（Gene Transfer Format）格式是借鉴于GFF2格式，也被称为GFF2.5，大部分字段的定义是和GFF2相同的，只是每行的第九列必须带有如下四个域，具体为gene_id value; transcript_id value; 这样的设计是为了适应一个基因的多个转录本这种情况。比如下面的这个例子：

```
AB000123    Twinscan    CDS    193817    194022    .    -    2    gene_id
"AB000123.1"; transcript_id "AB00123.1.2";
AB000123    Twinscan    CDS    199645    199752    .    -    2    gene_id
"AB000123.1"; transcript_id "AB00123.1.2";
AB000123    Twinscan    CDS    200369    200508    .    -    1    gene_id
"AB000123.1"; transcript_id "AB00123.1.2";
AB000123    Twinscan    CDS    215991    216028    .    -    0    gene_id
"AB000123.1"; transcript_id "AB00123.1.2";
AB000123    Twinscan    start_codon    216026    216028    .    -    .
gene_id    "AB000123.1"; transcript_id "AB00123.1.2";
AB000123    Twinscan    stop_codon    193814    193816    .    -    .
gene_id    "AB000123.1"; transcript_id "AB00123.1.2";
```

---

gff3

http://gmod.org/wiki/GFF3 http://www.sequenceontology.org/gff3.shtml

GFF2格式早期用的比较多，但是现在用的多的是GFF3格式，这也是好多软件所支持的，比如Gbrowse，Jbrowse等基因组数据可视化工具。

先看下面这个简单的例子：

```
##gff-version 3
ctg123  .  exon  1300  1500  .  +  .
ID=exon00001
ctg123  .  exon  1050  1500  .  +  .
ID=exon00002
ctg123  .  exon  3000  3902  .  +  .
ID=exon00003
ctg123  .  exon  5000  5500  .  +  .
ID=exon00004
ctg123  .  exon  7000  9000  .  +  .
ID=exon00005
```

第一行的##gff-version 3通常是需要的，而且必须是在文件的第一行。

前八列和GFF2、GFF2.5类似，但是有几点是要特别注意的，主要是将GFF3注释数据用于基因组浏览器时，字段中的一些特殊字符比如空格，> %等都需要使用URL编码进行转换才能准确的在web中进行展示。

第九列同样是表示attributes，采用的同样是键值对的形式（tag=value），只是这里有几个特定的键，具体如下：

ID，feature在整个GFF3文件中唯一的标识符；

Name，feature的名字，不同于ID，Name不要求唯一，只是方便用户浏览；

Alias， 相当于feature的别名；

Parent，表明该feature所属的上一级feature 的ID，这种关系可用于exons-transcripts，transcripts-genes，可以看出一个feature可以拥有多个子feature；

Target， 主要是用于序列比对结果的展示，value的格式为target_id start end [strand], 其中如果target_id中含有空格则需转换为%20；

后面还有些其它属性比如Note等，这里不再做详细描述。

下面再来看下典型的例子：

- **蛋白质编码基因结构**

```
ctg123 example gene            1050 9000 . + . ID=EDEN;Name=EDEN;Note=protein
kinase

ctg123 example mRNA            1050 9000 . + .
ID=EDEN.1;Parent=EDEN;Name=EDEN.1;Index=1
ctg123 example five_prime_UTR  1050 1200 . + . Parent=EDEN.1
ctg123 example CDS             1201 1500 . + 0 Parent=EDEN.1
ctg123 example CDS             3000 3902 . + 0 Parent=EDEN.1
ctg123 example CDS             5000 5500 . + 0 Parent=EDEN.1
ctg123 example CDS             7000 7608 . + 0 Parent=EDEN.1
ctg123 example three_prime_UTR 7609 9000 . + . Parent=EDEN.1

ctg123 example mRNA            1050 9000 . + .
ID=EDEN.2;Parent=EDEN;Name=EDEN.2;Index=1
ctg123 example five_prime_UTR  1050 1200 . + . Parent=EDEN.2
ctg123 example CDS             1201 1500 . + 0 Parent=EDEN.2
ctg123 example CDS             5000 5500 . + 0 Parent=EDEN.2
ctg123 example CDS             7000 7608 . + 0 Parent=EDEN.2
ctg123 example three_prime_UTR 7609 9000 . + . Parent=EDEN.2

ctg123 example mRNA            1300 9000 . + .
ID=EDEN.3;Parent=EDEN;Name=EDEN.3;Index=1
ctg123 example five_prime_UTR  1300 1500 . + . Parent=EDEN.3
ctg123 example five_prime_UTR  3000 3300 . + . Parent=EDEN.3
ctg123 example CDS             3301 3902 . + 0 Parent=EDEN.3
ctg123 example CDS             5000 5500 . + 1 Parent=EDEN.3
ctg123 example CDS             7000 7600 . + 1 Parent=EDEN.3
ctg123 example three_prime_UTR 7601 9000 . + . Parent=EDEN.3
```

一个名为EDEN的基因拥有三个转录本，分别名为EDEN.1 EDEN.2 EDEN.3， 每个转录本又有UTR和CDS等信息。

- **序列比对**

```
ctg123 est EST_match 1050 1500 . + . ID=Match1;Name=agt830.5;Target=agt830.5 1 451
ctg123 est EST_match 3000 3202 . + . ID=Match1;Name=agt830.5;Target=agt830.5 452
654

ctg123 est EST_match 5410 5500 . - . ID=Match2;Name=agt830.3;Target=agt830.3 505
595
ctg123 est EST_match 7000 7503 . - . ID=Match2;Name=agt830.3;Target=agt830.3 1 504

ctg123 est EST_match 1050 1500 . + . ID=Match3;Name=agt221.5;Target=agt221.5 1 451
ctg123 est EST_match 5000 5500 . + . ID=Match3;Name=agt221.5;Target=agt221.5 452
952
ctg123 est EST_match 7000 7300 . + . ID=Match3;Name=agt221.5;Target=agt221.5 953
1253
```

- **定量数据**

```
ctg123 affy microarray_oligo   1 100 281 . .
Name=Expt1
ctg123 affy microarray_oligo 101 200 183 . .
Name=Expt1
ctg123 affy microarray_oligo 201 300 213 . .
Name=Expt1
ctg123 affy microarray_oligo 301 400 191 . .
Name=Expt1
ctg123 affy microarray_oligo 401 500 288 . .
Name=Expt1
ctg123 affy microarray_oligo 501 600 184 . .
Name=Expt1
```

- **含Fasta格式的GFF3格式文件**

```
##gff-version 3
ctg123 . exon            1300  1500  .  +  .
ID=exon00001
ctg123 . exon            1050  1500  .  +  .
ID=exon00002
ctg123 . exon            3000  3902  .  +  .
ID=exon00003
ctg123 . exon            5000  5500  .  +  .
ID=exon00004
ctg123 . exon            7000  9000  .  +  .
ID=exon00005
##FASTA
>ctg123
cttctgggcgtacccgattctcggagaacttgccgcaccattccgccttg
tgttcattgctgcctgcatgttcattgtctacctcggctacgtgtggcta
tctttcctcggtgccctcgtgcacggagtcgagaaaccaaagaacaaaaa
aagaaattaaaatatttatttgctgtggtttttgatgtgtgttttttat
aatgatttttgatgtgaccaattgtacttttcctttaaatgaaatgtaat
cttaaatgtatttccgacgaattcgaggcctgaaaagtgtgacgccattc
...
```

该GFF3文件中含有对应的序列，以##FASTA作为标示。

---

bed

bed格式同样是用于展示序列注释信息，有相应的软件来处理这类格式的文件，如bedtools。可以用在类似GBrowse这样的基因组数据可视化工具中。 以tab隔开，它必须的三个字段为 chrom、chromStart、chromEnd，还有9个可选字段。

注意：用于在GBrowse上展示相关注释的bed格式通常第一行有一个关于track的描述信息。

比如下面的例子：

```
track name=pairedReads description="Clone Paired Reads"
useScore=1
chr22 1000 5000 cloneA 960 + 1000 5000 0 2 567,488, 0,3512
chr22 2000 6000 cloneB 900 - 2000 6000 0 2 433,399, 0,3601
```

---

sam/bam

下面主要讲解下详细比对部分字段的具体含义：

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	`[!-?A-~]{1,255}`	Query template NAME
2	FLAG	Int	`[0,2^16-1]`	bitwise FLAG
3	RNAME	String	`\*|[!-()+-<>-~][!-~]*`	Reference sequence NAME
4	POS	Int	`[0,2^31-1]`	1-based leftmost mapping POSition
5	MAPQ	Int	`[0,2^8-1]`	MAPping Quality
6	CIGAR	String	`\*|([0-9]+[MIDNSHPX=])+`	CIGAR string
7	RNEXT	String	`\*|=|[!-()+-<>-~][!-~]*`	Ref. name of the mate/next read
8	PNEXT	Int	`[0,2^31-1]`	Position of the mate/next read
9	TLEN	Int	`[-2^31+1,2^31-1]`	observed Template LENgth
10	SEQ	String	`\*|[A-Za-z=.]+`	segment SEQuence
11	QUAL	String	`[!-~]+`	ASCII of Phred-scaled base QUALity+33

其中：

1. QNAME 表示的是查询序列的名称即短片段（reads）的名称；

2. FLAG 以整数来表示比对的结果，不同数值有不同的意义，数值也可以是下列数的组合；

Bit		Description
1	0x1	template having multiple segments in sequencing
2	0x2	each segment properly aligned according to the aligner
4	0x4	segment unmapped
8	0x8	next segment in the template unmapped
16	0x10	SEQ being reverse complemented
32	0x20	SEQ of the next segment in the template being reverse complemented
64	0x40	the first segment in the template
128	0x80	the last segment in the template
256	0x100	secondary alignment
512	0x200	not passing quality controls
1024	0x400	PCR or optical duplicate
2048	0x800	supplementary alignment

比如如果FLAG是4的话则表示该reads没有比对到参考序列上，flag为16表示single-end reads比对到参考序列的反链上，

flag为83（64+16+2+1）表示paired-end reads中的第一个reads比对到参考序列上了。

3. RNAME 表示参考序列的名称，比如基因组的染色体编号等，如果没有比对上则显示为*；

4. POS 表示比对的起始位置，以1开始计数，如果没有比对上则显示为0；

5. MAPQ 比对质量；

6. CIGAR CIGAR 字符串，即比对的详细情况，简要比对信息表达式（Compact Idiosyncratic Gapped Alignment Report），其以参考序列为基础，使用数字加字母表示比对结果，比如3S6M1P1I4M，前三个碱基被剪切去除了，然后6个比对上了，然后打开了一 个缺口，有一个碱基插入，最后是4个比对上了，是按照顺序的；

7. RNEXT 双末端测序中下一个reads比对的参考系列的名称，如果没有则用*表示，如果和前一个reads比对到同一个参考序列则用=表示；

8. PNEXT 下一个reads比对到参考序列上的位置，如果没有则用0表示；

9. ISIZE/TLEN query序列的模板长度或者插入长度，Template的长度，最左边得为正，最右边的为负，中间的不用定义正负，不分区段（single-segment)的比对上，或者不可用时，此处为0；

10. reads的序列信息；

11. reads的序列质量信息，同FASTQ。

后面还有些可选字段，比如：

可选字段（optional fields)，格式如：TAG:TYPE:VALUE，其中TAG有两个大写字母组成，每个TAG代表一类信息，每一行一个TAG只能出现一次，TYPE表示TAG对应值的类型，可以是字符串、整数、字节、数组等。

示例：

```
:497:R:-272+13M17D24M 113 1 497 37 37M 15 100338662 0
CGGGTCTGACCTGAGGAGAACTGTGCTCCGCCTTCAG 0;==-==9;>>>>>=>>>>>>>>>>>>=>>>>>>>>>> XT:A:U
NM:i:0 SM:i:37 AM:i:0 X0:i:1 X1:i:0 XM:i:0 XO:i:0 XG:i:0 MD:Z:37
:20389:F:275+18M2D19M 99 1 17644 0 37M = 17919 314
TATGACTGCTAATAATACCTACACATGTTAGAACCAT >>>>>>>>>>>>>>>>>>>><<>>><<>>4:::>>:<9
RG:Z:UM0098:1 XT:A:R NM:i:0 SM:i:0 AM:i:0 X0:i:4 X1:i:0 XM:i:0 XO:i:0 XG:i:0 MD:Z:37
:20389:F:275+18M2D19M 147 1 17919 0 18M2D19M = 17644 -314
GTAGTACCAACTGTAAGTCCTTATCTTCATACTTTGT ;44999;499<8<8<<<8<<><<<<><7<;<<<>><< XT:A:R
NM:i:2 SM:i:0 AM:i:0 X0:i:4 X1:i:0 XM:i:0 XO:i:1 XG:i:2 MD:Z:18^CA19
:21597+10M2I25M:R:-209 83 1 21678 0 8M2I27M = 21469 -244
CACCACATCACATATACCAAGCCTGGCTGTGTCTTCT <;9<<5><<<<><<<><>><<><><>><9><><>>9>><><> XT:A:R
NM:i:2 SM:i:0 AM:i:0 X0:i:5 X1:i:0 XM:i:0 XO:i:1 XG:i:2 MD:Z:35
```

解释：

Field	Alignment 1	Alignment 2	Alignment 3	Alignment 4
QNAME	1:497:R:-272+13M17D24M	19:20389:F:275+18M2D19M	19:20389:F:275+18M2D19M	9:21597+10M2I25M:R:-209
FLAG	113	99	147	83
RNAME	1	1	1	1
POS	497	17644	17919	21678
MAPQ	37	0	0	0
CIGAR	37M	37M	18M2D19M	8M2I27M
MRNM/RNEXT	15	=	=	=
MPOS/PNEXT	100338662	17919	17644	21469
ISIZE/TLEN	0	314	-314	-244
SEQ	CGGGTCTGACCTGAGGAGAACTGTGCTCCGCCTTCAG	TATGACTGCTAATAATACCTACACATGTTAGAACCAT	GTAGTACCAACTGTAAGTCCTTATCTTCATACTTTGT	CACCACATCACATATACCAAGCCTGGCTGTGTCTTCT
QUAL	0;==-==9;>>>>>=>>>>>>>>>>>=>>>>>>>>>>	>>>>>>>>>>>>>>>>>>>><<>>><<>>4:::>>:<9	;44999;499<8<8<<<8<<><<<<><7<;<<<>><<	<;9<<5><<<<><<<><>><<><><>><9><><>>9>><><>
TAGs	XT:A:U NM:i:0 SM:i:37 AM:i:0 X0:i:1 X1:i:0 XM:i:0 XO:i:0 XG:i:0 MD:Z:37	RG:Z:UM0098:1 XT:A:R NM:i:0 SM:i:0 AM:i:0 X0:i:4 X1:i:0 XM:i:0 XO:i:0 XG:i:0 MD:Z:37	XT:A:R NM:i:2 SM:i:0 AM:i:0 X0:i:4 X1:i:0 XM:i:0 XO:i:1 XG:i:2 MD:Z:18^CA19	XT:A:R NM:i:2 SM:i:0 AM:i:0 X0:i:5 X1:i:0 XM:i:0 XO:i:1 XG:i:2 MD:Z:35

其中可以看出Aligenment 2 和 Alignment 3是成对的reads，其插入长度为314。

bam格式中的b是binary的意思，是sam格式的二进制表示方式，为什么要用二进制表示呢？ 因为sam格式文件大小通常是十分大的，一般是以G为单位，所以为了减少存储量等因素而将sam转换为二进制格式以便于分析。

sam/bam格式是由特定的一些软件（比如samtools）来处理的，包括格式互转、排序、建立索引、搜寻突变等操作，后续分析中会详细讲解samtools工具的使用方法。

---

vcf

http://samtools.github.io/hts-specs/VCFv4.2.pdf

vcf（Variant Call Format）格式是用于表示突变信息的文本格式，可以用来表示single nucleotide variants, insertions/deletions, copy number variants and structural variants等。VCF格式同样是分为两大部分，一部分是注释描述信息，一部分是具体的突变信息，其中注释信息是以##开头的，我们来看下面这个例子：

```
##fileformat=VCFv4.2
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Homo sapiens",taxonomy=x>
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF    ALT     QUAL FILTER INFO                              FORMAT      NA00001        NA00002        NA00003
20     14370   rs6054257 G      A       29   PASS   NS=3;DP=14;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51 1/1:43:5:.,.
20     17330   .         T      A       3    q10    NS=3;DP=11;AF=0.017               GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3   0/0:41:3
20     1110696 rs6040355 A      G,T     67   PASS   NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2   2/2:35:4
20     1230237 .         T      .       47   PASS   NS=3;DP=13;AA=T                   GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51 0/0:61:2
20     1234567 microsat1 GTC    G,GTCT  50   PASS   NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

我们着重来关注第二部分的每列字段是什么含义：

CHROM 即chromosome， 染色体名称；

POS 即position， 发生突变的参考序列的位置（从1开始计数）；

ID 突变的名称；

REF 参考序列POS上的碱基；

ALT 发生突变的碱基，多个的话以,连接， 可选符号为ATCGN*，大小写敏感；

QUAL 基于Phred格式的表示ALT的质量，也可以理解为可靠性；

FILTER 过滤后的状态，即按照可靠性进行筛选；

INFO 额外信息，可结合注释描述信息进行理解

- AA : ancestral allele
- AC : allele count in genotypes, for each ALT allele, in the same order as listed
- AF : allele frequency for each ALT allele in the same order as listed: use this when estimated from primary data, not called genotypes
- AN : total number of alleles in called genotypes
- BQ : RMS base quality at this position
- CIGAR : cigar string describing how to align an alternate allele to the reference allele
- DB : dbSNP membership
- DP : combined depth across samples, e.g. DP=154
- END : end position of the variant described in this record (for use with symbolic alleles)
- H2 : membership in hapmap2
- H3 : membership in hapmap3
- MQ : RMS mapping quality, e.g. MQ=52
- MQ0 : Number of MAPQ == 0 reads covering this record
- NS : Number of samples with data
- SB : strand bias at this position
- SOMATIC : indicates that the record is a somatic mutation, for cancer genomics
- VALIDATED : validated by follow-up experiment
- 1000G : membership in 1000 Genomes

针对vcf格式有如bcftools等软件进行处理。