

SAM / BAM Tutorial

EMBL Heidelberg

Course Materials

Tobias Rausch

June 2011

Contents

| | | |
|----------|---------------------------------|----------|
| 1 | SAM / BAM | 3 |
| 1.1 | Introduction | 3 |
| 1.2 | Tasks | 3 |
| 1.2.1 | Viewing a BAM file | 3 |
| 1.2.2 | Converting BAM to SAM | 4 |
| 1.2.3 | The flag field | 4 |
| 1.2.4 | Alignment statistics | 5 |
| 2 | SNP Calling | 7 |
| 2.1 | Introduction | 7 |
| 2.2 | Tasks | 7 |
| 2.2.1 | Running Pileup | 7 |

SAM / BAM

1.1 Introduction

The SAM (**S**equencing **A**lignment / **M**apping) format is a flexible alignment format (Li et al., 2009). It is already supported by a number of read mappers and seems to become the de facto standard alignment format. It is also used in the 1000 Genomes project (www.1000genomes.org). The format specification ships with a set of open source tools, called SAM Tools (samtools.sourceforge.net). The SAM Tools provide utilities to manipulate alignments in the SAM format.

1.2 Tasks

1.2.1 Viewing a BAM file

For the first task I provided a tiny bam file called `align.bam`. Please go to your `tmp` directory and create a new directory called `smallsam` there.

```
cd /tmp
mkdir smallsam
```

Now copy the bam file across.

```
cp align.bam /tmp/smallsam/
cd /tmp/smallsam/
ls
```

Please do the same for the artificial reference sequence, called `all.fa`.

```
cp all.fa /tmp/smallsam/
cd /tmp/smallsam/
ls
```

The reference is so small that you can have a look at it by using the `cat` command.

```
cat all.fa
```

Before we can view the bam file we have to index it. The index helps the viewer to quickly find all reads belonging to a certain chromosomal region and thus speeds up the rendering. The command to index the BAM file is:

```
samtools index align.bam
```

| Index | Field Name | Description |
|-------|------------|--|
| 1 | QNAME | Query pair NAME if paired; or Query NAME if unpaired |
| 2 | FLAG | Bitwise FLAG |
| 3 | RNAME | Reference sequence NAME |
| 4 | POS | 1-based leftmost POSition of the clipped sequence |
| 5 | MAPQ | MAPping Quality |
| 6 | CIGAR | Extended CIGAR string |
| 7 | MRNM | Mate Reference sequence NaMe; "=" if the same as RNAME |
| 8 | MPOS | 1-based leftmost Mate POSition of the clipped sequence |
| 9 | ISIZE | Inferred Insert SIZE |
| 10 | SEQ | Query SEQuence |
| 11 | QUAL | Query QUALity |

Table 1.1: Brief summary of the SAM format

Finally, we can start the samtools viewer.

```
samtools tview align.bam all.fa
```

The very first row shows the reference sequence. The underlined row underneath the reference is the consensus sequence of all aligned reads. Usually, it is simply the most frequent letter occurring in each column. When you press the ?-Key you get a small help screen. To leave the help screen you have to press the q-Key for quit. Try out a couple of keys shown in the help screen. For instance, press the .-Key to toggle on or off the dot view. Note that in the dot view a , indicates a read aligned to the reverse strand whereas a . indicates a read aligned to the forward strand.

1.2.2 Converting BAM to SAM

The samtools offer the view command to dump the content of a BAM file in SAM format.

```
samtools view align.bam
```

Using output redirection we write the content to a file, called align.sam.

```
samtools view align.bam > align.sam
```

Use the SAM Format specification from the web (samtools.sourceforge.net) or my brief summary Table 1.1 to familiarize yourself with the SAM format. Note that all columns are tab-delimited. Hence, to show only the CIGAR strings you can use:

```
cut -f 6 align.sam
```

1.2.3 The flag field

The flag field is a decimal number that has to be interpreted as a 16-bit binary number.

$$f_{15}f_{14}f_{13}f_{12}f_{11}f_{10}f_9f_8f_7f_6f_5f_4f_3f_2f_1f_0$$

Linux has a command called bc that offers such a conversion. Using that command we can convert a decimal number to binary by specifying that the output base (obase) is 2.

| Field | Hex Code | Description |
|----------|----------|---|
| f_0 | 0x0001 | the read is paired in sequencing, no matter whether it is mapped in a pair |
| f_1 | 0x0002 | the read is mapped in a proper pair (depends on the protocol, normally inferred during alignment) |
| f_2 | 0x0004 | the query sequence itself is unmapped |
| f_3 | 0x0008 | the mate is unmapped |
| f_4 | 0x0010 | strand of the query (0 for forward; 1 for reverse strand) |
| f_5 | 0x0020 | strand of the mate |
| f_6 | 0x0040 | the read is the first read in a pair |
| f_7 | 0x0080 | the read is the second read in a pair |
| f_8 | 0x0100 | the alignment is not primary (a read having split hits may have multiple primary alignment records) |
| f_9 | 0x0200 | the read fails platform/vendor quality checks |
| f_{10} | 0x0400 | the read is either a PCR duplicate or an optical duplicate |

Table 1.2: Brief summary of the SAM format

```
echo 'obase=2;8' | bc
echo 'obase=2;4' | bc
echo 'obase=2;12' | bc
```

Thus, the last command tells us:

$$f_3 = 1; f_2 = 1; f_1 = 0; f_0 = 0;$$

Obviously, each flag f_i has a meaning. Please look up the flags in the SAM specification or refer to Table 1.2. The Hex Code can be used together with `awk`. For example, if I want to extract all mapped reads I have to check that bit f_2 is NOT set to 1. Remember that the second field (the flag field in SAM) can be accessed using `$2` in `awk`.

```
awk '!and($2, 0x0004)' align.sam
```

1.2.4 Alignment statistics

Using the linux commands and `awk` try to answer the following questions. In brackets you will see a hint to the linux commands you might want to use.

1. How many reads are in the file [wc]?
2. How many reads are mapped [awk, wc]?
3. How many reads have been mapped to the reverse strand [awk, wc]?
4. How many reads have been mapped to the forward strand [awk, wc]?
5. How many reads are paired in sequencing [awk, wc]?
6. How many reads start at position 2 [awk, wc]?
7. Create a frequency table that shows how many reads per chromosome [cut, sort, uniq]?

8. Create a frequency table that shows how many reads start at what genomic position [cut, sort, uniq]?

SNP Calling

2.1 Introduction

Calling single-nucleotide polymorphisms is a non-trivial task. The toughest problem is to distinct true SNP positions from sequencing errors. Let us have a look again on the alignment example from Section 1.

```
samtools tview align.bam all.fa
```

One column is labeled with the R consensus letter indicating that in this column the purines A and G are occurring. Hence, it might be a potential SNP column. Doing SNP calling manually on a 3GB genome is an impossible task. Most SNP callers simply process the alignment in a column-wise fashion and then output all putative SNP columns. They also use all kinds of heuristics to distinguish a true SNP from a false one that simply appeared due to a collapsed repeat or misaligned reads.

2.2 Tasks

2.2.1 Running Pileup

The samtools package offers the so-called pileup command. Pileup computes a consensus letter for each alignment column. Let us execute it on the previously used small BAM file.

```
samtools pileup -cf all.fa align.bam
```

The columns of that file are briefly explained in Table 2.1. Let us focus on the single SNP column with consensus letter R we have seen before.

```
samtools pileup -cf all.fa align.bam | grep "R"
```

A dot or a comma in the read base column stands for a match to the reference, either on the forward (dot) or reverse (comma) strand. Any other DNA nucleotide that appears in the read base column stands for a mismatch. Upper case letters are mismatches on the forward strand whereas lower case letters are mismatches on the reverse strand. There are also a couple of so called meta-characters. A + indicates an insertion, a minus a deletion. A ^ symbol marks the start of a read segment and is followed by the mapping quality of that base. A \$ symbol marks the end of a read segment.

1. Please compare the pileup output with the graphical view of the alignment using tview.

2. SNP Calling

| Index | Field Name | Description |
|-------|------------|------------------------------------|
| 1 | Chr | Chromosome identifier |
| 2 | Pos | 1-based coordinate |
| 3 | RBase | Reference base |
| 4 | CBase | Consensus base |
| 5 | CQual | Consensus quality |
| 6 | SNP | SNP quality |
| 7 | MAPQ | Maximum mapping quality |
| 8 | DEPTH | Number of reads covering this site |
| 9 | BASE | Read bases |
| 10 | QUAL | Base qualities |

Table 2.1: Brief summary of the Pileup consensus format

2. Try to understand the read base column for insertions and deletions.

Bibliography

H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, and . G. P. D. P. Subgroup. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 2009.

Index

1000 Genomes project, 3

Alignment statistics, 5

Awk, 5

BAM, 3

CIGAR, 4

Consensus, 7

Converting BAM/SAM, 4

Flag, 5

Flag field, 4

Pileup, 7

SAM, 3

SAM Fields, 4

SAM Format, 4

Samtools, 3

 flagstat, 5

 index, 4

 pileup, 7

 tview, 4

 view, 4

SNP, 7

SNP Calling, 7

Variant Calling, 7