**Week 8, Lecture 15**

István Albert

Biochemistry and Molecular Biology and Bioinformatics Consulting Center

Penn State

---

**Beyond SAM**

SAM files are great but

- are a bit too large – a compressed format would be better
- cannot be easily searched – would be great to filter or query them

SAM $\rightarrow$ BAM (Binary Alignment Format)

---

**Binary SAM (BAM) files**

**SAM file:**
- information on the alignment of each read
- optimized for readability and sequential access

**BAM (binary SAM):**
- compression $\rightarrow$ saves space (optimized for size)
- may be sorted + indexed $\rightarrow$ location query (optimized for random access)
- the file is not readable by eye

Your default format should be BAM – only turn it into SAM when viewing the file

---

**SAM/BAM hierarchy**

- **SAM file**
  - transform (view)
- **BAM file**
  - sorting
  - Transform (view)
  - Sorted BAM file
  - index
  - Indexed BAM file
  - Sorted SAM file

Some tools have certain requirements of what type of SAM/BAM they take.

Your default data format should be a sorted, indexed BAM file!
Beyond BAM → CRAM

A new push is underway to move from BAM to CRAM.

European Bioinformatics Institute will start using it right away.

It offers better compression.

It is going to be a long way to adoption (years perhaps). For now BAM is what your will most likely use.

Samtools: is suite of commands

Samtools is a suite of programs for processing high-throughput sequencing data. It contains three separate programs, which communicate with each other via stdin, stdout, and status codes.

Different commands:

- index
- index
- query
- sort
- merge
- merge

Most actions will provide help on their usage

Download and ‘make’ SAMTOOLS

http://samtools.sourceforge.net/
Default Operation

- By default, `samtools` expects a **BAM** file as input and will produce a **SAM** file as output.

- Alignment results are typically stored as a sorted and **indexed** **BAM** file.

- Aligners produce **SAM** files so our first job is usually to convert those to **BAM** formats.

Transform SAM to BAM

**transform to bam**

```
samtools view -Sb input.sam > tempfile.bam
```

**sort bam file**

```
samtools sort -f tempfile.bam output.bam
```

**Index bam file**

```
samtools index output.bam
```

Automate the process with a shell script

```
# Parameters from the command line.
READ1=-1
READ2=-2
SAM=-3

# This is the reference file.
REFS=/refs/HS7/bola-1999.fq

# Run the alignment
rm -f temp.sam temp.sra
samtools view >S temp.sra > temp.sra
# Start the conversion to a BAM file.
samtools view -Sb temp.sra > temp.bam
# Sort the alignment.
samtools sort -f temp.bam $SAM
# Index the alignment.
samtools index $SAM
# Get rid of intermediate files.
rm -f temp.sra temp.sra

# Parameters from the command line.
READ1=-1
READ2=-2
SAM=-3

# This is the reference file.
REFS=/refs/HS7/bola-1999.fq

# Run the alignment
rm -f temp.sam temp.sra
samtools view >S temp.sra > temp.sra
# Start the conversion to a BAM file.
samtools view -Sb temp.sra > temp.bam
# Sort the alignment.
samtools sort -f temp.bam $SAM
# Index the alignment.
samtools index $SAM
# Get rid of intermediate files.
rm -f temp.sra temp.sra
```

Filtering SAM/BAM files

**Required flag (keep if matches)**

```
samtools view -f
```

**Filtering flag (remove if matches)**

```
samtools view -F
```
Flags are using a bitwise representation

1 = 00000001 \rightarrow \text{paired end read}
2 = 00000010 \rightarrow \text{mapped as proper pair}
4 = 00000100 \rightarrow \text{unmapped read}
8 = 00001000 \rightarrow \text{read mate unmapped}
16 = 00010000 \rightarrow \text{read mapped on reverse strand}

\text{samtools filtering}

\begin{verbatim}
# how many reads in total
samtools view -c results.bam

# reads that cannot be mapped
samtools view -c -f 4 results.bam

# reads that can be mapped
samtools view -c -f 0 results.bam

# reads that map to reverse strand
samtools view -c -f 16 results.bam

# reads that map to forward strand
samtools view -c -f 15 results.bam

# reads that have a minimum mapping quality of 1
samtools view -c -q 1 results.bam

# note that for BWA this also means unique alignment!
samtools view -c -q 1 results.bam
\end{verbatim}

A sorted file will stay sorted during transformation

- Once sorted all output will stay sorted regardless of the output type (SAM, BAM)
- You can creating a second, smaller and filtered file that does not need to be sorted again.
- You do need to index the new file though!

\text{Explore other commands}

\begin{verbatim}
flag statistics
samtools flagstat data.bam

index stats
samtools idxstats data.bam

depth of coverage
samtools depth data.bam | head
\end{verbatim}
Querying a BAM file **name:start-end**

Samtools allows querying by location:

```bash
# The reference name is a bit complex.
# let’s create a shortcut to it
CHR='gi|10313991|ref|NC_002549.1,'

# Slice into the datafile.
samtools view results.bam $CHR:1-100
```

---

**Homework 16**

Simulate paired end reads with **wgsmi** from the Ebola genome of 1999. Use a simulation with an error rate of at least **10%**. Align these reads back to the genome.

1. Find the number of uniquely mapped reads (MAPQ>0)
2. Find the number of high quality alignments (MAPQ>30) for each strand separately.
3. A genomic feature has its start site on the forward strand on chromosome I at position 10,000.
   - How many reads fall within 500b upstream of this location?
   - Report the number of reads in this region for each strand separately.