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 → 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
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.
Download and ‘make’ SAMTOOLS

http://samtools.sourceforge.net/
Samtools: is suite of commands

$ samtools

Program: samtools (Tools for alignments in the SAM format)
Version: 1.1 (using htslib 1.1)

Usage: samtools <command> [options]

Commands:
  -- indexing
    faiidx  index/extract FASTA
    index   index alignment
  -- editing
    calmd   recalculate MD/NM tags and '=' bases
    fixmate fix mate information
    reheader replace BAM header
    rmdup   remove PCR duplicates
    targetcut cut fosmid regions (for fosmid pool only)
  -- file operations
    bamshuf shuffle and group alignments by name
    cat     concatenate BAMs
    merge   merge sorted alignments
    mpileup multi-way pileup
    sort    sort alignment file
    split   splits a file by read group
    bam2fq  converts a BAM to a FASTQ

different commands
Most actions will provide help on their usage.

```bash
samtools view -b -C -l -u -h -H -c -o FILE -U FILE -t FILE <in.bam>|<in.sam>|<in.cram> [region ...]
```

Options:
- `b`: output BAM
- `C`: output CRAM (requires -T)
- `l`: use fast BAM compression (implies -b)
- `u`: uncompressed BAM output (implies -b)
- `h`: include header in SAM output
- `H`: print SAM header only (no alignments)
- `c`: print only the count of matching records
- `o FILE`: output file name [stdout]
- `U FILE`: output reads not selected by filters to FILE [null]
- `t FILE`: FILE listing reference names and lengths (see long help)
- `[null]`
- `T FILE`: reference sequence FASTA FILE [null]
- `L FILE`: only include reads overlapping this BED FILE [null]
- `r STR`: only include reads in read group STR [null]
- `R FILE`: only include reads with read group listed in FILE [null]
- `q INT`: only include reads with mapping quality >= INT [0]
- `l STR`: only include reads in library STR [null]
- `m INT`: only include reads with number of CIGAR operations consuming query sequence >= INT [0]
- `f INT`: only include reads with all bits set in INT set in FLAG [null]
• 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

```bash
# Parameters from the command line.
READ1=$1
READ2=$2
BAM=$3

# This is the reference file.
REFS=~/.refs/852/ebola-1999.fa

# Run the alignment
bwa mem $REFS $READ1 $READ2 > temp.sam

# Start the conversion to a BAM file.
samtools view -Sb temp.sam > temp.bam

# Sort the alignment.
samtools sort -f temp.bam $BAM

# Index the alignment.
samtools index $BAM

# Get rid of intermediate files.
rm -f temp.sam temp.bam

echo "*** finished, alignments in: $BAM"
```
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 → paired end read
2 = 00000010 → mapped as proper pair
4 = 00000100 → unmapped read
8 = 00001000 → read mate unmapped
16 = 00010000 → read mapped on reverse strand

```
ialbert@porthos ~/work/lec12
$ ~/bin/samtools view -c -f 4 results.bam
1

ialbert@porthos ~/work/lec12
$ ~/bin/samtools view -c -F 4 results.bam
3
```

-c means to count the lines
-f <number> - keep reads that match
-F <number> - remove reads that match
Samtools filtering

```bash
# 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 4 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 16 results.bam

# reads that have a minimum mapping quality of 1
# note that for BWA this also means unique alignment!
samtools view -c -q 1 results.bam
```
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!
Explore other commands

Flag statistics

```bash
samtools flagstat data.bam
```

Index stats

```bash
samtools idxstats data.bam
```

Depth of coverage

```bash
samtools depth data.bam | head
```
Samtools allows querying by location:

```shell
# 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 wgsim 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.