Week 6, Lecture 12

István Albert

Biochemistry and Molecular Biology
and Bioinformatics Consulting Center
Penn State

Binary SAM (BAM) files

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

BAM (binary SAM):
- compression → saves space (optimized for size)
- may be sorted + indexed → 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 necessary!

SAM/BAM hierarchy

SAM file
transform (view)

BAM file
sorting

Sorted BAM file
Transform (view)

Sorted SAM file

index

Indexed BAM 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!

Download and ‘make’ SAMTOOLS

SAMtools

Some tools have certain requirements of what type of SAM/BAM they take.
Your default data format should be a sorted, indexed BAM file!
Samtools: is suite of programs

Default operation

- By default **samtools** expects to see a **BAM** file and wants to produce a **SAM** file as output

- To process other inputs/outputs we need to pass certain flags to **samtools**.

Transform SAM to BAM!

**samtools view -Sb input.sam > output.bam**
Sort/Index BAM file

```
ailbert@portos $ /work/lecl12
$ ./bin/samtools sort results.bam sorted-results
ailbert@portos $ /work/lecl12
$ ls -l
total 24
-rw-r--r--  1 ailbert  staff  598B Oct  4 09:19 results.bam
-rw-r--r--  1 ailbert  staff  1364 Oct  4 09:18 results.sam
-rw-r--r--  1 ailbert  staff  598B Oct  4 09:21 sorted-results.bam
ailbert@portos $ /work/lecl12
$ ./bin/samtools index sorted-results.bam
ailbert@portos $ /work/lecl12
$ ls -l
total 32
-rw-r--r--  1 ailbert  staff  598B Oct  4 09:19 results.bam
-rw-r--r--  1 ailbert  staff  1364 Oct  4 09:18 results.sam
-rw-r--r--  1 ailbert  staff  2248 Oct  4 09:21 sorted-results.bam
ailbert@portos $ /work/lecl12
```

Filtering SAM/BAM files

```
ailbert@portos $ /work/lecl12
$ ./bin/samtools view
Usage: samtools view [options] <in.bam>|<in.sam> [region: [...]]
Options:
  -b output BAM
  -h print header for the BAM output
  -s print header only (no alignments)
  -S output in SAM
  -u unsorted BAM output (force -b)
  -1 fast compression (force -b)
  -a output FLAG in SAM (samtools-c specific)
  -g output FLAG in string (samtools-c specific)
  -d print only the count of matching records
  -l FILE output alignments overlapping the input BED FILE [null]
  -t FILE list of reference names and lengths (force -s) [null]
  -f FILE reference sequence file (force -s) [null]
  -o FILE output file name [stdin]
  -k FILE list of read groups to be outputted [null]
  -e INT required flag, 0 for unset [0]
  -f INT filtering flag, 0 for unset [0]
  -p INT minimum mapping quality [0]
  -r INT only output reads in library [null]
  -m INT only output reads in read group [null]
  -k INT minimum fraction of tags in subsample, integer part as seed [-1]
  -? longer help
```

Write a script that does the work in one step

```
1
2 # producing a BAM file from the query
3 #
4 # this script output samtools view -b <query.fa>
5 # convert to bam
6 # sort bam
7 # index bam
8 # end
9
10 # this script output samtools view -b -F 4 <results.fa>
11 # convert to bam
12 # sort bam
13 # index bam
14 # end
```

Flags are using a bitwise representation

- 1 \( \rightarrow \) paired end read
- 2 \( \rightarrow \) mapped as proper pair
- 4 \( \rightarrow \) unmappable read
- 8 \( \rightarrow \) read mate unmapped
- 16 \( \rightarrow \) read mapped on reverse strand

```bash
ailbert@portos $ /work/lecl12
$ ./bin/samtools view -c -F 4 results.bam
1
ailbert@portos $ /work/lecl12
$ ./bin/samtools view -c -F 4 results.bam
3
```

<means to count the lines
\(<\text{number}>\) - keep reads that match
\(<\text{number}>\) - remove reads that match
Filtering examples

The output is a SAM file (unless overridden)

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!

Other interesting features
Homework 12

Generate a **sorted** and **indexed** BAM file based on the data `lecture-11.fq` Show the commands and report the results for:

1. Computing the number of uniquely mapped reads
2. Computing 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 111000:
   - How many reads fall within 500 bp upstream of this location?
   - Print the position of each read (hint: there are not that many)
   - Report the number of reads for each strand separately.