**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC 1/SC 29/WG 11 N17755**

**July 2018, Ljubljana, SI**

|  |  |
| --- | --- |
| **Source:** | **Requirements** |
| **Status:** | **Approved** |
| **Title:** | **Samtools support in MPEG-G** |

# Introduction

Samtools is an open source tool [1] widely used by the genomic community. This document aims at documenting how to map common uses of Samtools on the MPEG-G API. It would be beneficial to provide users with the possibility to perform the same functions on MPEG-G files as they do on SAM and BAM files. Additional functionality which is not directly available in Samtools and which is provided by the MPEG-G format and API, is also described in this document.

# Analysis of common Samtools functions

## view

BAM (Binary Alignment Map) files are the lossless compressed binary equivalent of SAM.

To convert SAM file to BAM, Samtools uses the **view** command.

Example:

*samtools* ***view*** *-S -b –o E.coli\_alignedreads.sam E.coli\_alignedreads.sorted.bam*

*-S indicates that the input is SAM*

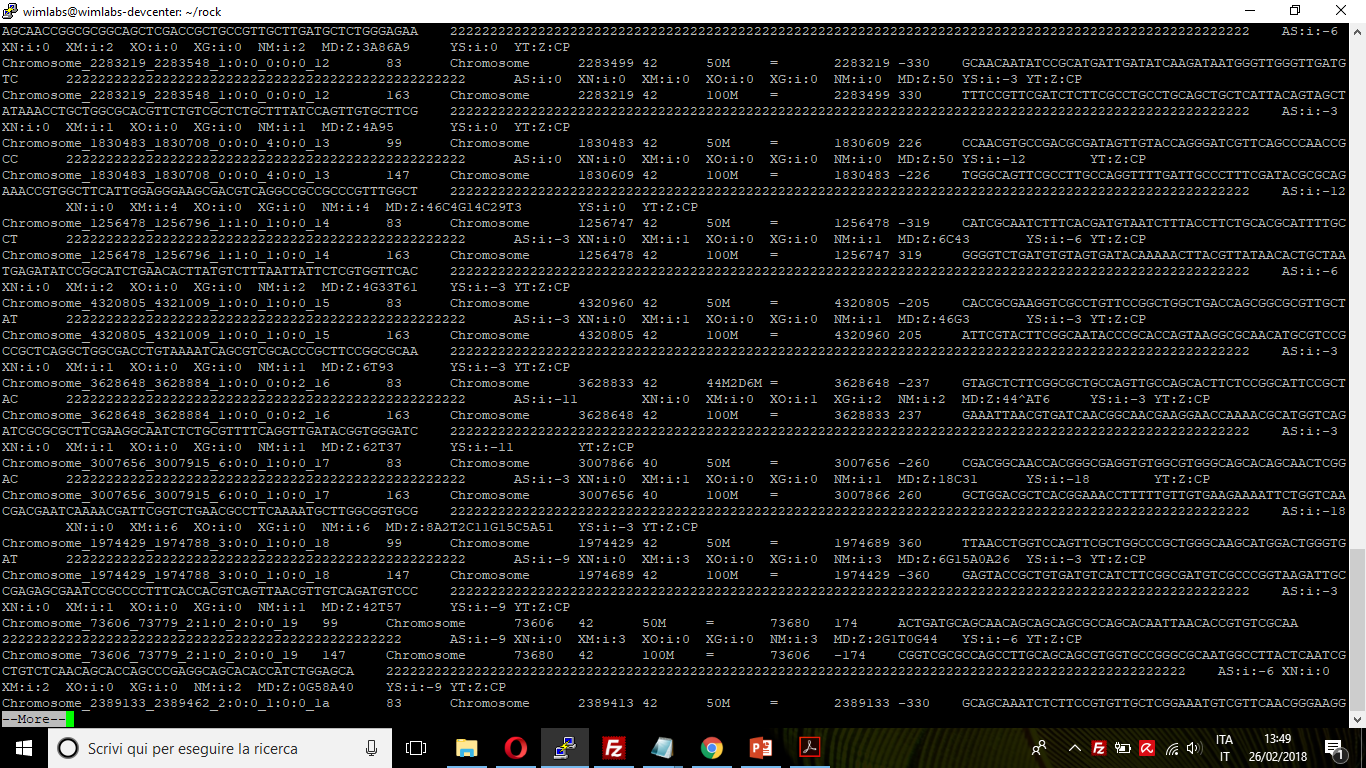
*-b indicates that the output is BAM*

*-o specifies the name of the output file*

Samtools also uses the **view** command to display SAM content.

Example:

*samtools* ***view*** *E.coli\_alignedreads.bam*



Samtools uses **view** command to only filter reads with a specific Flag field:

-f only output alignments with the *INT* flag; viceversa -F do not output reads with the *INT* flag.

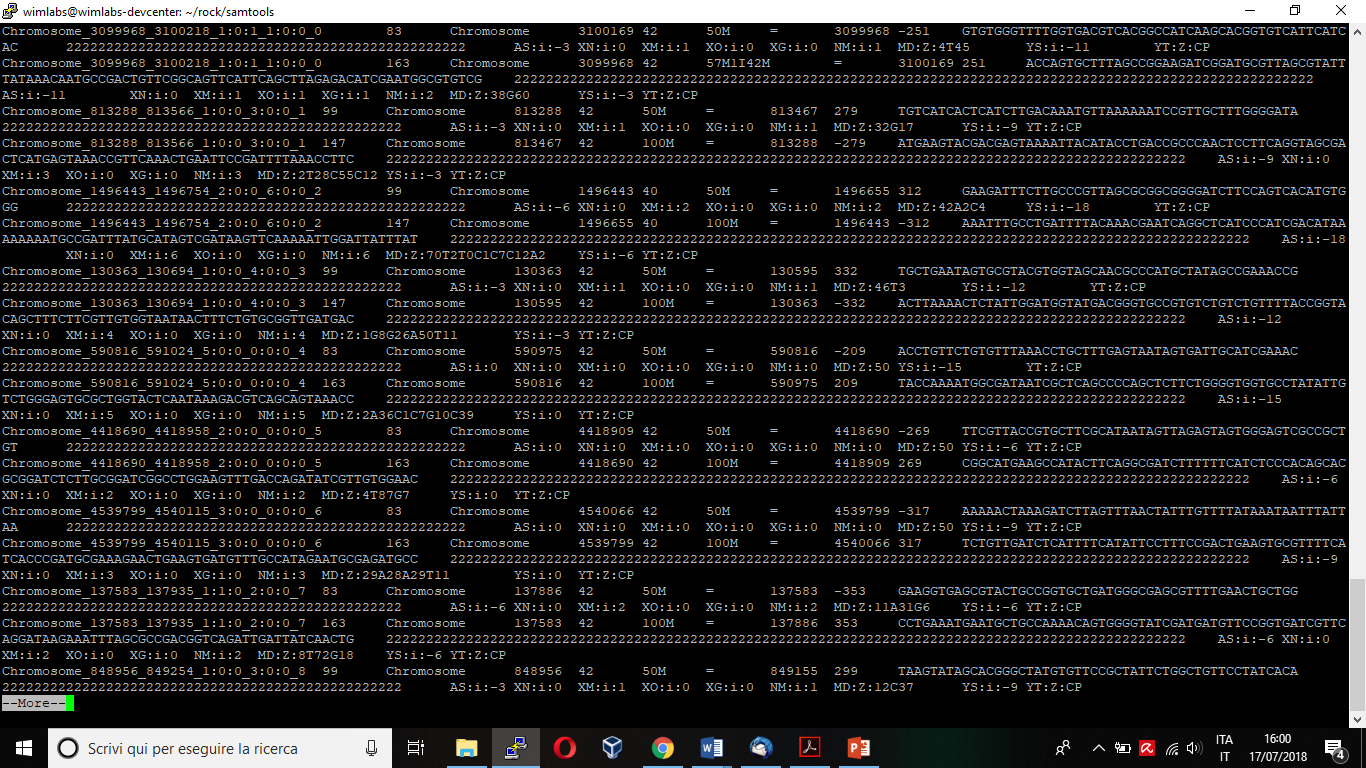
Table 1: SAM flag fields

|  |  |  |
| --- | --- | --- |
| **Int value** | **Hex value** | **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 filters, such as platform/vendor quality controls |
| 1024 | 0x400 | PCR or optical duplicate |
| 2048 | 0x800 | supplementary alignment |

Example:

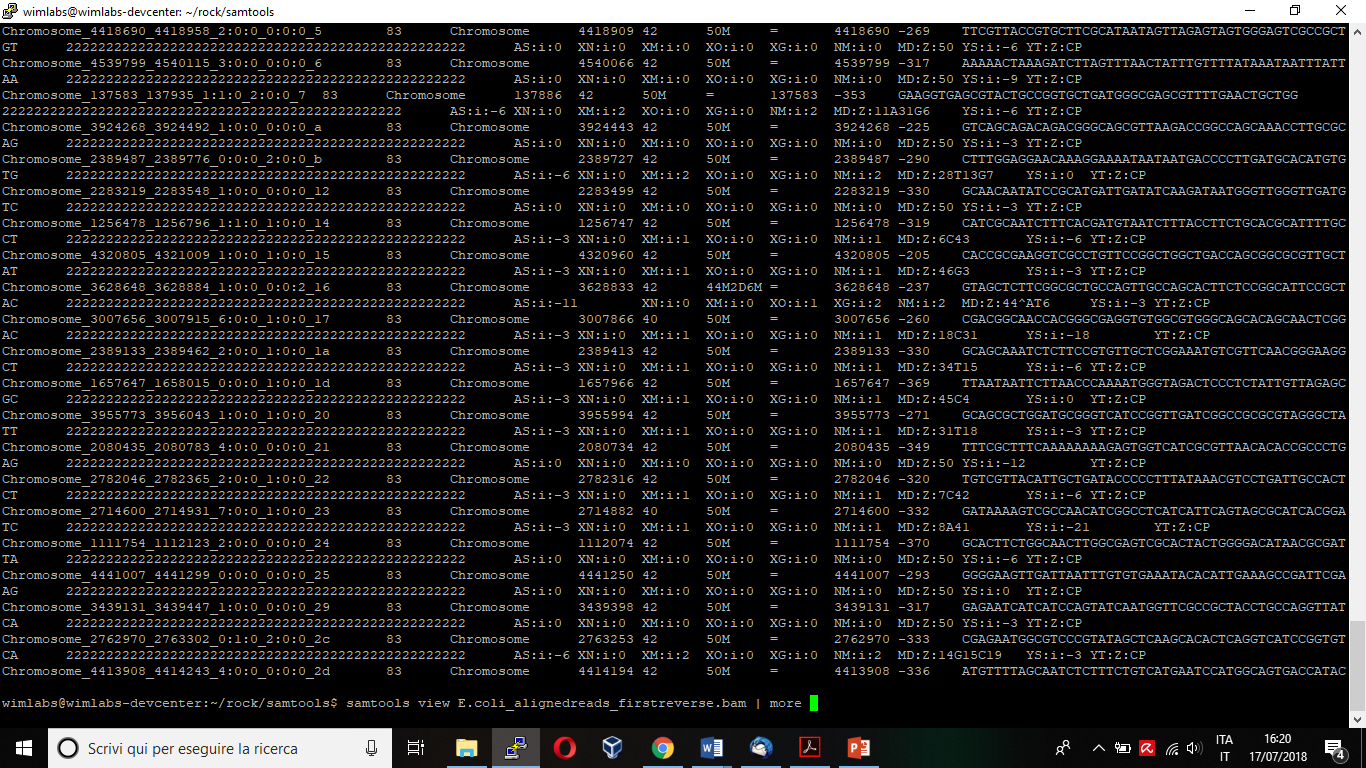
1 - Filtering out unmapped reads in BAM files:

*samtools* ***view*** *-h -F 4 E.coli\_alignedreads.bam > E.coli\_alignedreads\_onlymapped.bam*



2 - Filtering in only read paired (flag = 1), read mapped in proper pair (flag = 2), SEQ being reverse complemented (flag = 16), the first segment in the template (flag =64)

*samtools* ***view*** *-h -f 83 E.coli\_alignedreads.bam > E.coli\_alignedreads\_onlyfirstreverse.bam*

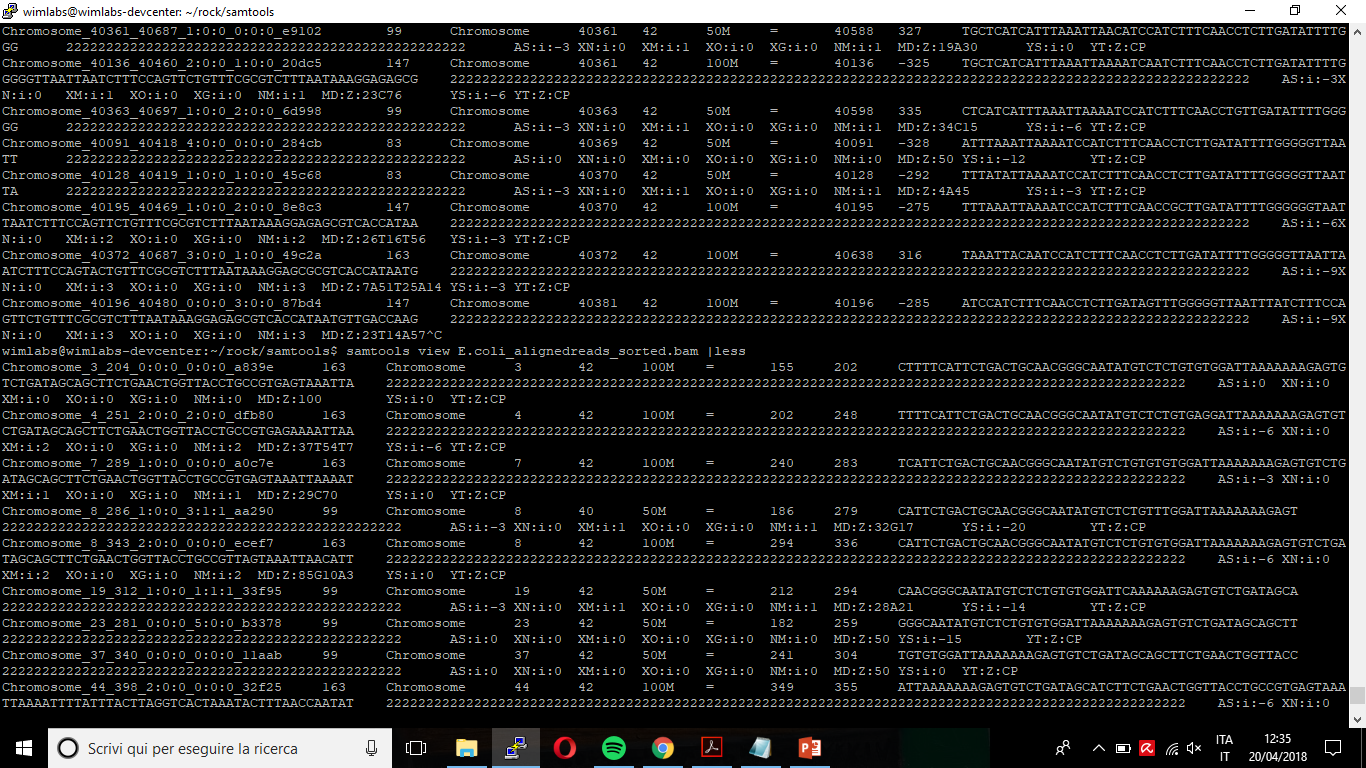


## sort

Samtools can sort reads according to their reference sequence position, using the s**ort** command.

Example:

*samtools* ***sort*** *–o E.coli\_alignedreads\_sorted.bam E.coli\_alignedreads.bam*



## index

Samtools uses the command **index** to produce an index file (.bai) which stores the BAM file offsets of genome positions.

Example:

samtools **index** *E.coli\_alignedreads.sorted.bam E.coli\_alignedreads.sorted.bai*

## view\_random access

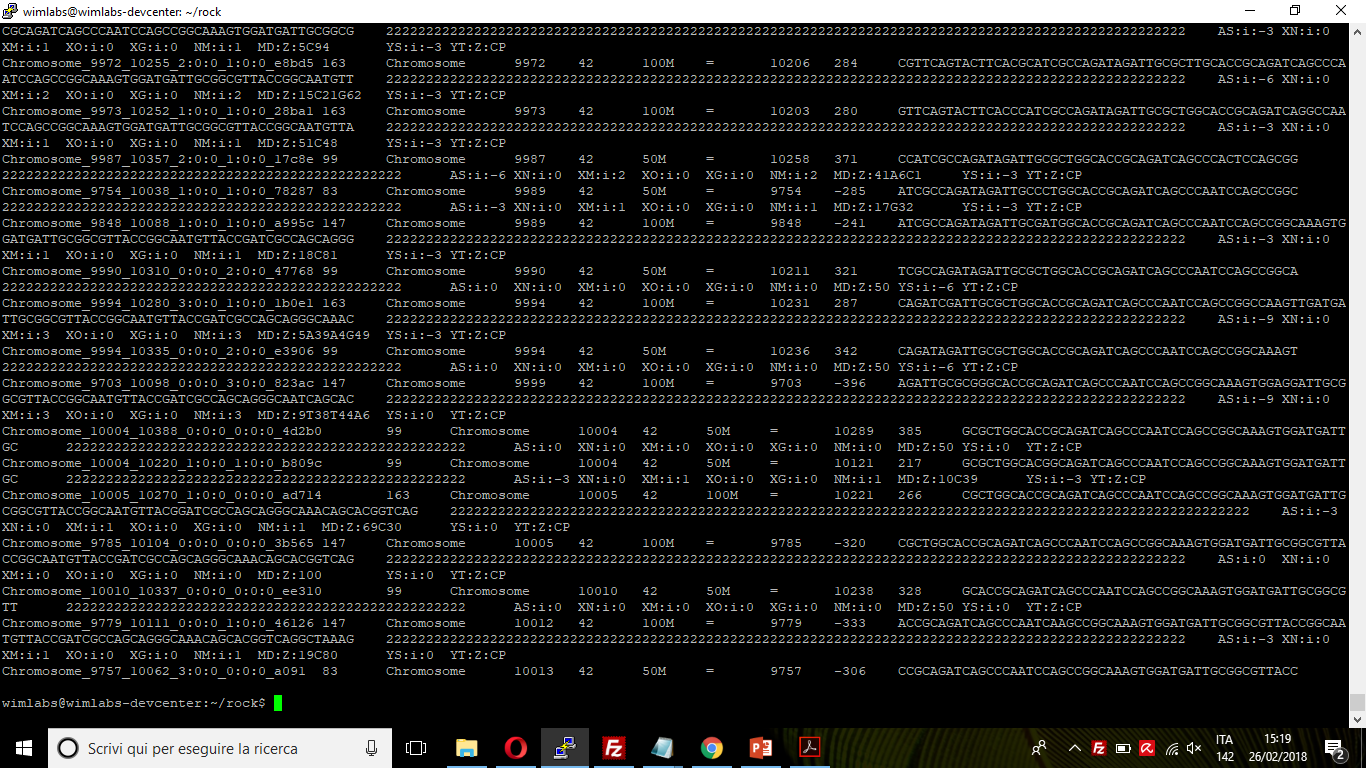
Samtools uses **view** command to make random access to a genomic region, printing on screen all reads having at least one base within the defined region.

The input file is the sorted bam file sorted (*E.coli\_alignedreads.sorted.bam* in previous examples); the genomic region must be specified by the reference sequence name and the start and/or end positions (e.g. ch3: 1000-2000).

Moreover, in the .bam file’s char must be present the equivalent .bai file created with **index** command (*E.coli\_alignedreads.sorted.bai* in previous example), that contains the index used by Samtools to jump directly to specific parts of the bam file without reading through all of the sequences.

Example:

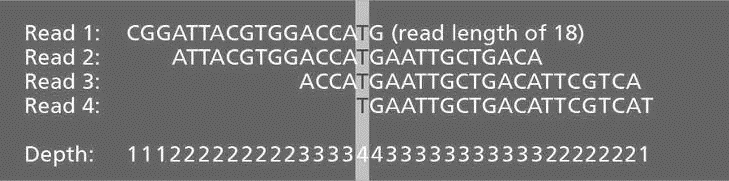
*Samtools* ***view*** *E.coli\_alignedreads\_sorted.bam Chromosome: 10000-10005*



Note: in this first line is also printed the read with 9999 as left-most base position because, even if the 9999 base is not included in the defined region, others are.

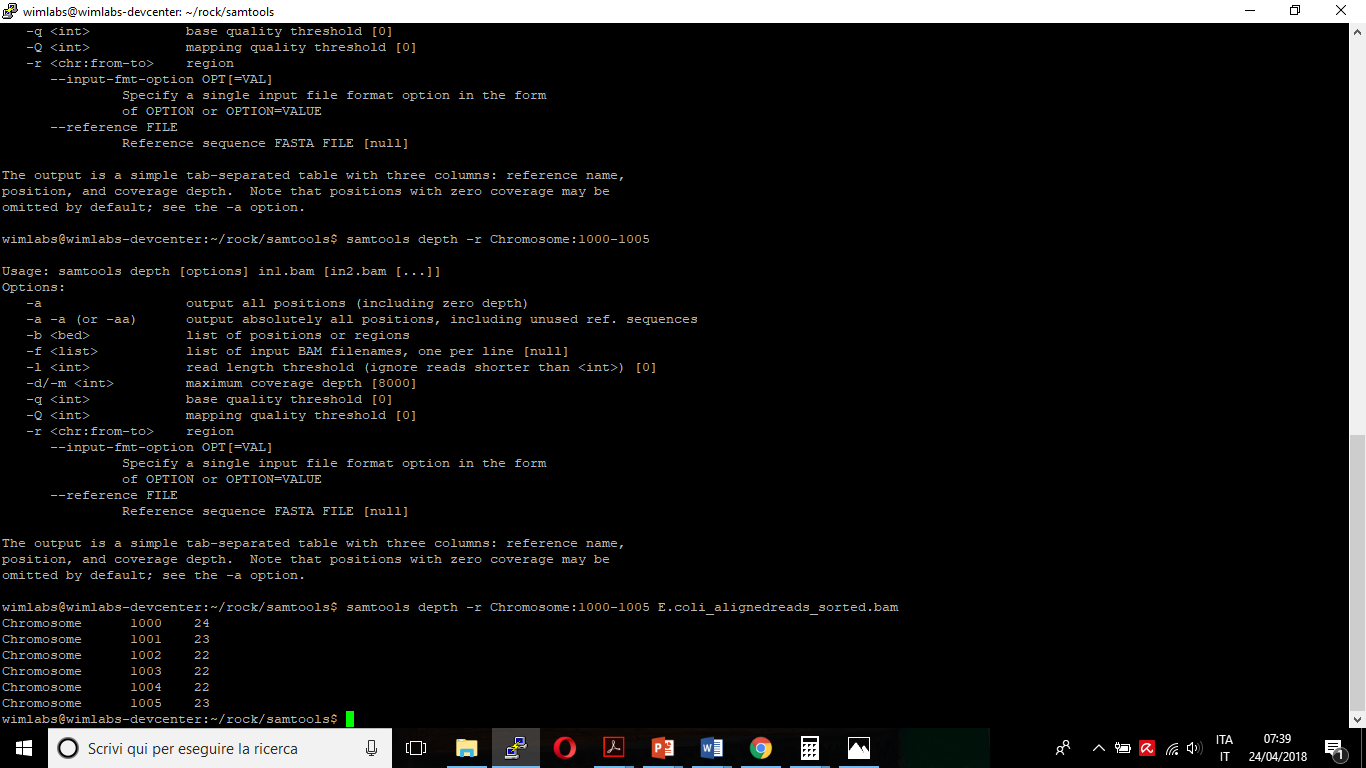
## depth

Samtools uses the **depth** command to compute the coverage (i.e. the number) of reads that include a given nucleotide in the reconstructed sequence (the input file must be sorted).



Example:

*Samtools* ***depth*** *-r Chromosome:1000-1005 E.coli\_alignedreads\_sorted.bam*



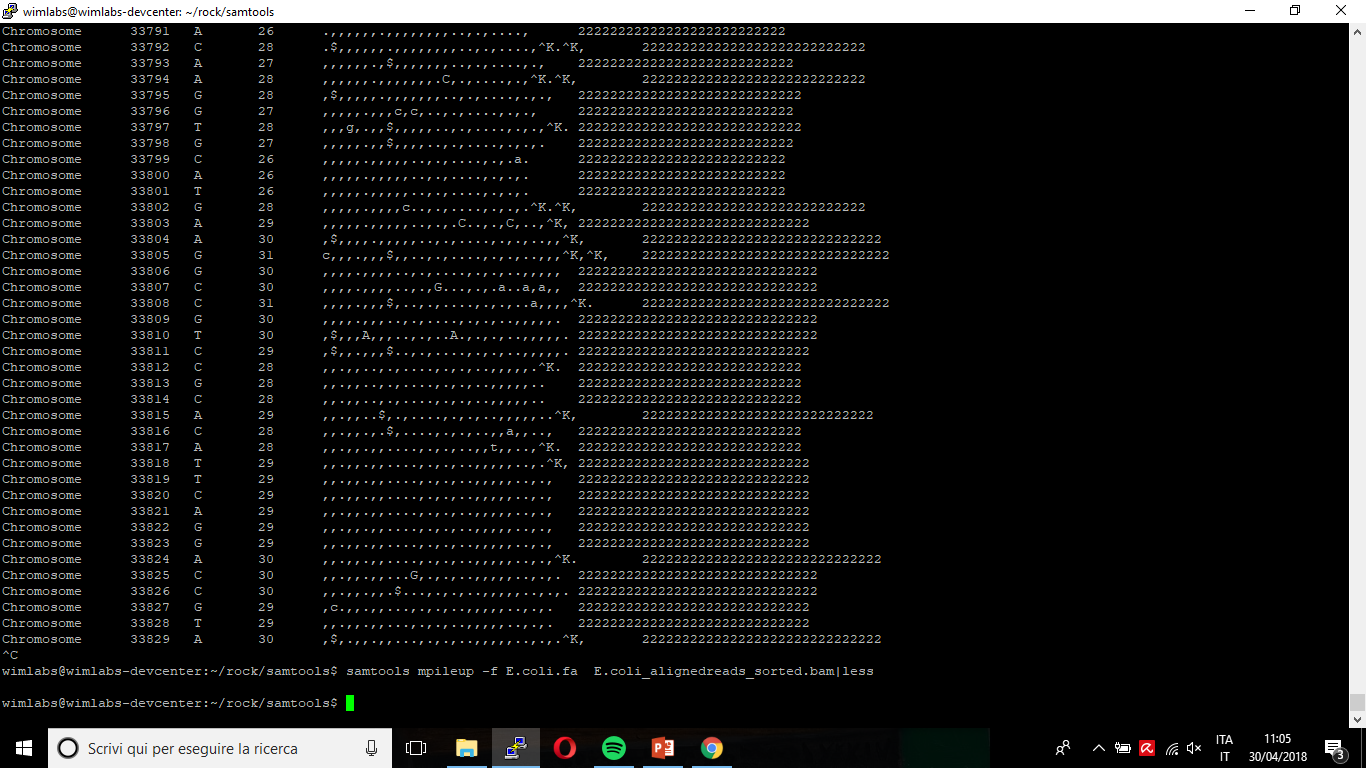
## mpileup

Samtools uses **mpileup** command to generate a pileup for one or multiple BAM files.

In the pileup format each line represents a genomic position, with the following information:

1. chromosome name
2. 1-based coordinate
3. reference base
4. the number of reads covering the site
5. read bases
6. base qualities

Example:



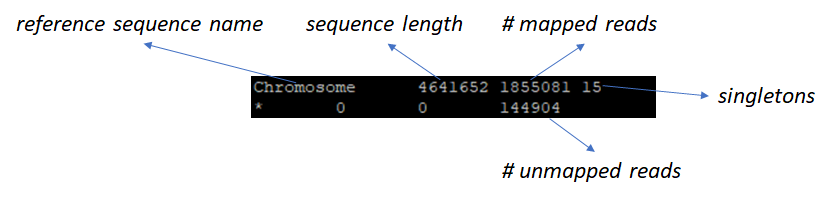
## idxstats

Samtools uses **idxstats** command to retrieve statistics such as:

* reference sequence name
* sequence length
* number of mapped reads
* number of singletons
* number of unmapped reads.

Example:

*Samtools* ***idxstats*** *E.coli\_alignedreads\_sorted.bam*



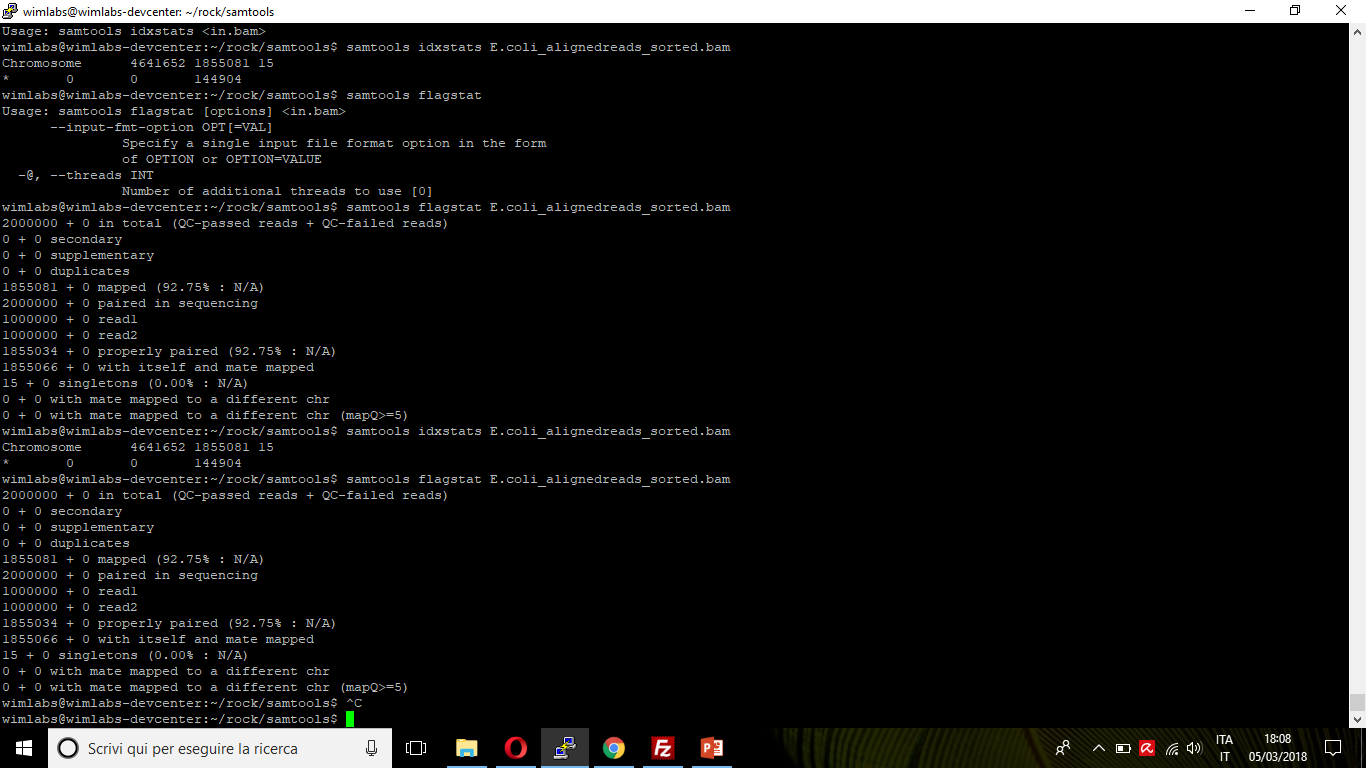
## flagstat

Samtools uses **flagstat** to perform a full pass through the input file to calculate and print statistics to stdout.

This operation is used to provide counts for each of 13 categories based primarily on bit flags in the FLAG field of the SAM format.

Example:

*samtools* ***flagstat*** *E.coli\_alignedreads\_sorted.bam*



## merge

Samtools uses the merge command to concatenate BAM files.

*samtools* ***cat*** *[-o out.bam] <in1.bam> <in2.bam> [ ... ]*

## faidx

Samtools uses the **faidx** command to create .fai file containing reference sequences indexed of the equivalent FASTA format.

Example:

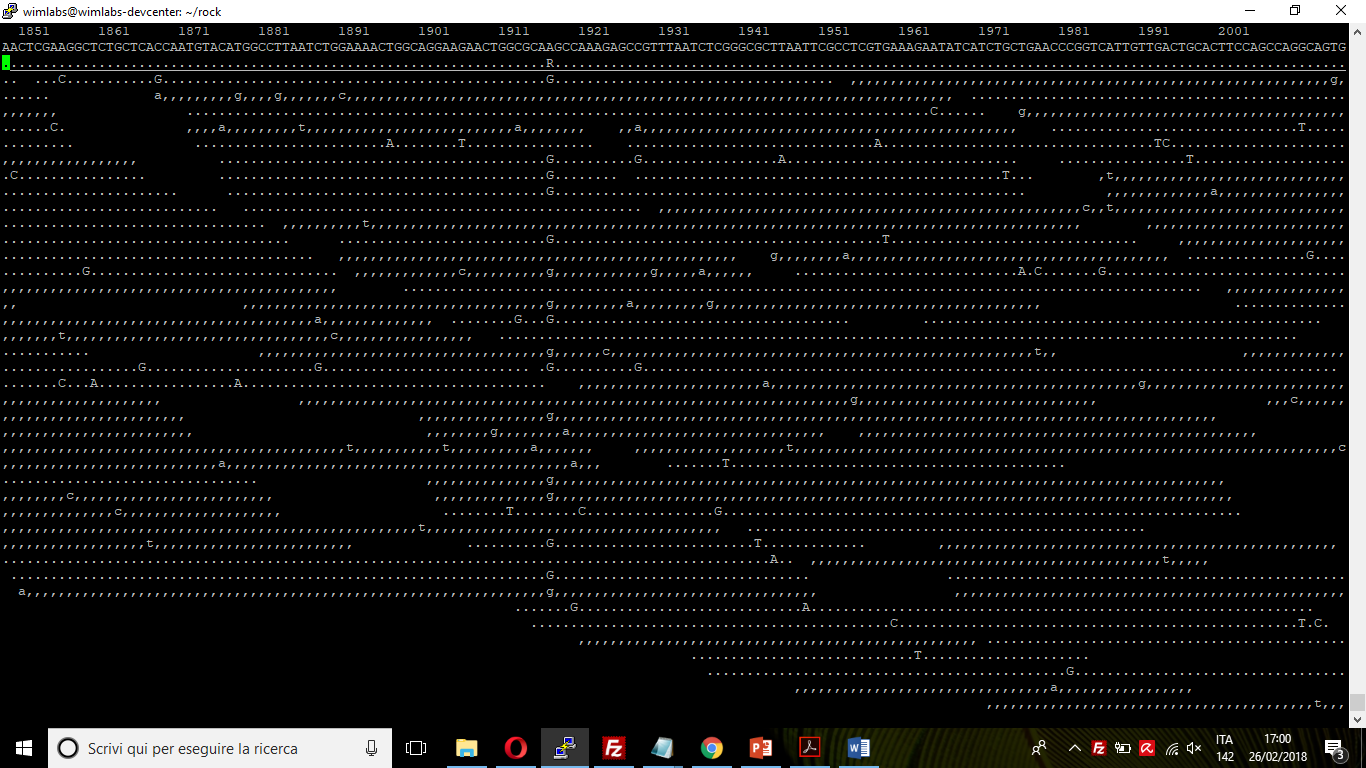
*samtools* ***faidx*** *<ref.fasta> [region1 [...]]*

## tview

Samtools uses the **tview** command to compare .bam file to the equivalent reference genome.

Example:

*samtools* ***tview*** *alignments/E.coli\_alignedreads\_sorted.bam Samtools/Escherichia\_coli.fa*



Single Nucleotide Polymorphism (SNP): G 🡪 A

## calmd

Samtools uses **calmd** to generate the MD tag.

The MD field aims to achieve mismatches and indels without looking at the reference.

Example:10A5^AC6

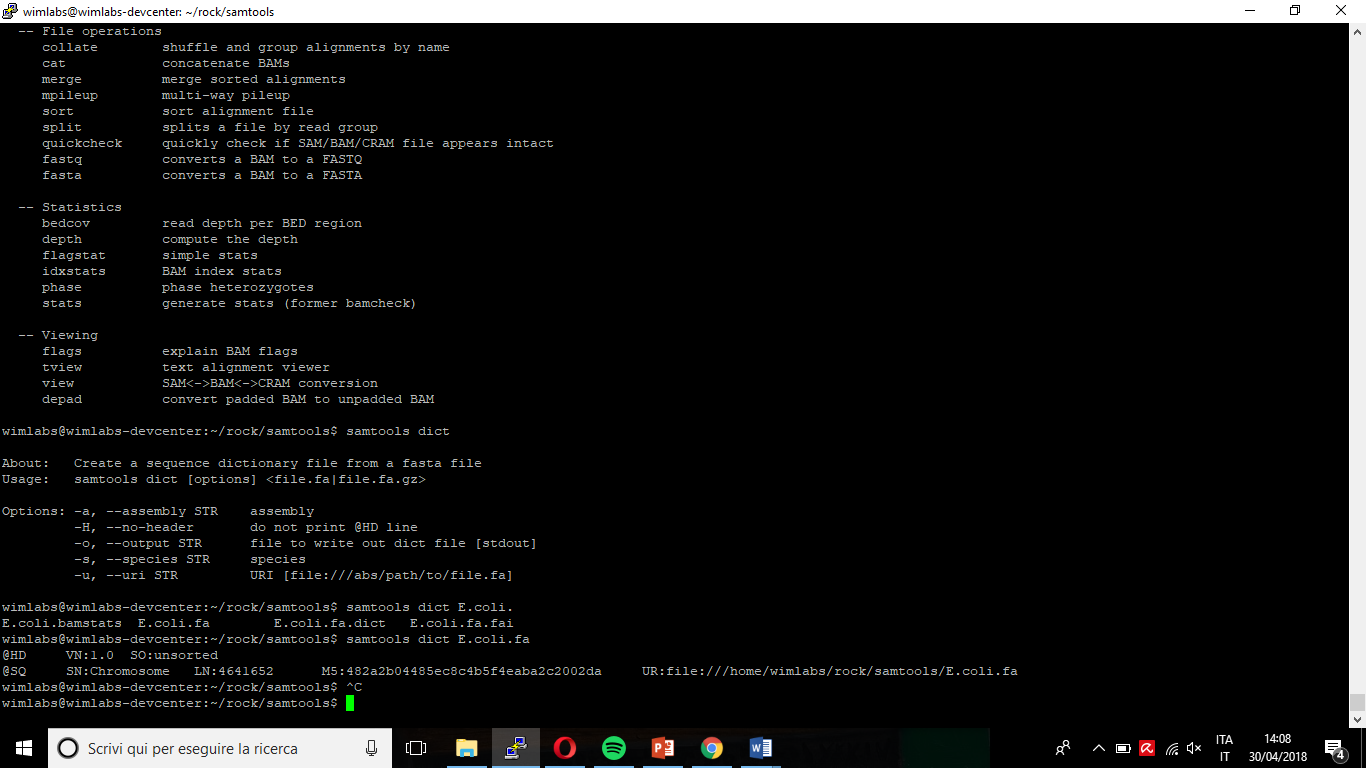
* 10 bases matching to the reference
* A mismatch to the reference
* 5 bases matching to the reference
* ^AC two base deletions from the reference where the deleted sequence is AC
* 6 bases matching to the reference

## dict

Samtools uses the **dict** command to generate the M5 and the URI of the reference file.

Example:

*samtools* ***dict*** *E.coli.fa*



# MPEG-G API

In order to facilitate access to and manipulation of MPEG-G compliant genomic content and the fields it contains, an **Application** **Programming** **Interface** (**API**) is specified in ISO/IEC 23092-3.

Table shows a classification of the different kind of operations defined in the API.

|  |  |
| --- | --- |
| **Operation name** | **Description** |
| Access | Operations that return content to the requester. |
| Modification | Operations that change content as indicated by the requester. |
| Authorization | Operations that check that the user has permission to perform an operation. |
| Verification | Operations that check the integrity of some content indicated by the requester. |
| Conversion | Operations that convert some content from/to MPEG-G to/from other formats. |

# Mapping of Samtools functions to MPEG-G API

Access (core) operations allow getting content from one or more fields, listing content and searching for content.

## view (random access) - sort – index – depth – mpileup - tview

Mapping of SAM fields to MPEG-G can be performed based on the mapping described in clause “SAM Interoperability” of ISO/IEC 23092-3.

In order to selectively retrieve MPEG-G content: GetDataDatasetGroup, GetDataDataset, GetRegionDatasetGroup, GetRegionDataset, GetByLabel.

In order to compute selective access of MPEG-G content: GetRegion and GetBySignature operations, defining as input parameters either the parameters seq\_id, start\_position and end\_position, in case of aligned content, or a signature, in case of unmapped content.

In order to compute data filtering operations: GetData operations, defining as input parameters class\_type and/or mismatch\_threshold.

An equivalent of the index command in MPEG-G is not needed as the indexing information is natively present in the Dataset Container (dtcn) as Master Index Table (mitb).

Samtools commands such as depth, mpileup and tview can be easily implemented in MPEG-G by parsing the MPEG-G record fields and analysing them in the uncompressed domain.

## Idxstats - Flagstat

Statistics that in SAM can be retrieved using the Idxstats and Flagstat commands can be retrieved in MPEG-G as follows:

Multi alignments mmap descriptor

Duplicates flag descriptor

Mapped Class P\_N\_M\_I\_HM

Properly paired flag descriptor

With itself and mate mapped rcomp descriptor

Singletons Class\_HM

With mate mapped to a different chr pair descriptor

In MPEG-G the equivalent statistics can be retrieved first by filtering data based on the class type and then by decoding the relevant descriptors, such as mmap, flags, rcomp, pair.

The same information can be easily retrieved in the uncompressed domain (i.e., after decoding) using the corresponding MPEG-G Record fields.

## merge

The merge operations can be easily executed using the MPEG-G format and particularly the features of Part 1 enabling aggregation, such as Dataset Group and Dataset, which do not require the content to be decoded and re-encoded.

## calmd

calmd generate the MD tag that contains equivalent information of mmpos and mmtype descriptors of the MPEG-G file and of the MPEG-G record field named ecigar\_string.

## faidx

The faidx command allows indexing the reference in the FASTA format or extract subsequences from the indexed reference.

The MPEG-G GetReference function specified in ISO/IEC 23092-3 allows retrieving the reference (or part thereof, such as a single reference sequence), either in MPEG-G (compressed or uncompressed) format, or in FAST-A format, which has been used to align and compress the relevant genomic content. The reference is converted into the rawReference format, as specified in part 2, to be made available to either a part 2 decoder or to an analysis application.  
The indexing information of an MPEG-G compressed reference is provided by the Master Index Table as specified in ISO/IEC 23092-1, using its ref\_sequence\_id, ref\_start\_position and ref\_end\_position fields.

## dict

General data integrity of an MPEG-G file or part thereof should be verified at the application level.

# MPEG-G additional functionality

## Labels

MPEG-G provides a labelling system, based on the label box as defined in ISO/IEC 23092-1, which allows associating a human readable string identifier to one or more Datasets, genomic regions and/or classes. Such a feature provides a user friendly approach to selective access and a way of marking possibly scattered parts of the MPEG-G content with a unique identifier meaningful to the user (such as a gene name for example).

## Data aggregation

MPEG-G inherently provides the capability of aggregating multiple genomic studies and/or datasets, by means of the box system defined in ISO/IEC 23092-1.

Particularly, the concepts of Dataset and DatasetGroup have been defined in order to aggregate and easily identify sets of genomic data representing the same entity at different levels.

## Data streaming

The MPEG-G transport format, as specified in ISO/IEC 23092-1, allows transmitting compressed data from a datacenter for analysis with real-time access. Transmission is possible as soon as data are available, with no need to create a monolithic file first. This feature is not available in SAM/BAM, as the SAM/BAM file is typically seen as a monolithic entity which can be retrieved remotely only by a progressive download approach, which implies that the content has to be fully available before starting the remote retrieval.

# Conclusions

The main Samtools operations have been studied and mapped to the API functions described in ISO/IEC 23092-3.

As a conclusion, MPEG-G can easily cover all the main operations that Samtools can perform on a BAM file.

Some of the operations can be implemented in a more efficient way using the MPEG-G file format and API functions.

Also, MPEG-G covers an extra set of functionality which is not supported by Samtools.

# References

1. http://www.htslib.org/doc/Samtools.html