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 N15092
Geneva, CH – February 2015

Source: **Requirements**

Status: **Approved**

Title: **Database for Evaluation of Genome Compression and Storage**

Authors: Claudio Alberti, Marco Mattavelli (EPFL), Leonardo Chiariglione (CEDEO), Ioannis Xenarios, Nicolas Guex, Heinz Stockinger, Thierry Schuepbach, Pascal Kahlem, Christian Iseli, Daniel Zerzion, Dmitry Kuznetsov (SIB), Yann Thoma, Enrico Petraglio (HEIG-VD), Cenk Sahinalp, Ibrahim Numanagic (Simon Fraser University), James Bonfield (Wellcome Trust Sanger Institute), Vadim Zalunin (EBI), Jaime Delgado (UPC)

Table of Contents

1 Purpose

This document is a first attempt at defining a set of statistically meaningful genomic data to be used as a shared test bed to assess the performance of information compression techniques.

2 Terminology

Term	Definition
Alignment	A sequence read mapped on a reference DNA sequence
BAM	Compressed binary version of SAM
CRAM	GIR that includes SAM + Compression configuration
FastA	GIR that includes header and sequence reads (nucleotides sequence)
FastQ	GIR that includes FastA + Quality Scores
GIR	Genomic Information Representation
Indel	An additional or missing nucleotide in a DNA sequence with respect to a reference DNA sequence.
MAF	Mutation Annotation Format. File format used to mark the genes and other biological features in a DNA sequence
Mate pairs	Two reads from the same (long) DNA strand extracted by sequencing machines. The orientation is the opposite of paired ends.
Paired ends	Couple of reads produced by the same (short) DNA fragment by sequencing both ends. The orientation is the opposite of mate pairs.
Quality score	A quality score is assigned to each nucleotide base call in automated sequencing processes. It expresses the base-call accuracy.
Read header	Each sequence read stored in FastA and FastQ format starts with a textual field called "header" containing a sequence identifier and an optional description
SAM	GIR that is human readable and includes FastQ + Alignment and analysis information
Sequence read	The readout, by a specific technology more or less prone to errors, of a continuous part of a segment of DNA extracted from an organic sample

3 Selection of a file format. FastQ and BAM.

After the 110th MPEG meeting in Strasbourg, the activity of the AhG on genome information compression and storage focused on the selection of the most appropriate file format for information representation among those currently used by the scientific community.

The selected Genomic Information container has to be appropriate for use by compression experts to test compression approaches in a comparable fashion, but this does not imply that MPEG is endorsing it as possible candidate for standardization.

The Genomic Information lifecycle goes through several steps of manipulation from generation to processing and analysis; it is then important to highlight that the Genomic Information which is relevant to this activity is:

1. Unmapped reads as produced by the sequencing machines
2. Metadata related to unmapped reads (essentially Quality Scores and headers)
3. Aligned reads

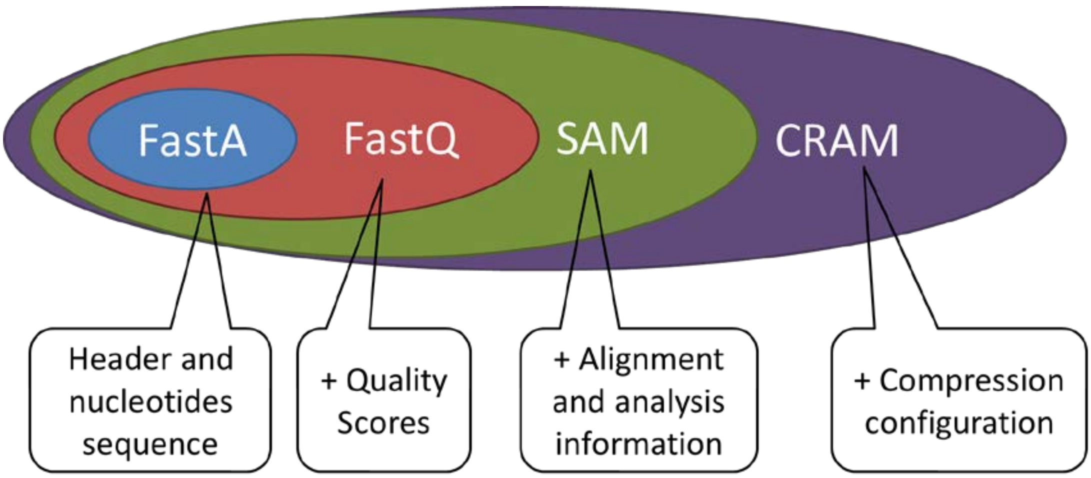
4. Metadata related to aligned reads

The AhG activity focused on evaluating advantages and disadvantages of the most popular file formats used by the scientific community according to their maturity, available tools, storage requirements etc.

The discussions involved both MPEG experts and specialists in computational biology from some of the most active international institutions and research centres such as the European Bioinformatics Institute (UK), the Wellcome Trust Sanger Institute (UK), the Swiss Institute of Bioinformatics (CH) and the Lab for Computational Biology at Simon Fraser University (CA).

Discussion via e-mail and during several teleconferences helped reaching an agreement in selecting SAM (and its compressed equivalent called BAM) as common file format for aligned reads. Unmapped reads will be represented as gzipped FastQ files with the exception of Oxford Nanopore data that are only available as Fast5 files. This format can be transcoded to FastQ using publicly available tools such as fast5tofastq or poretools.

A simplified schematic of the relation among the file formats is depicted in the picture below. SAM is the textual uncompressed equivalent of BAM. Compression in BAM is implemented as a block-based zip, while in CRAM approach to compression is more sophisticated and adopts different compression techniques according to the nature of the compressed information.


FastA
FastQ
SAM
CRAM
Header and nucleotides sequence
+ Quality Scores
+ Alignment and analysis information
+ Compression configuration

3.1 FastA and FastQ

FastA and FastQ are text-based formats organized as sequences of 2 (FastA) or 4 (FastQ) fields describing each read produced by a DNA sequencing machine. An example of these fields with a brief description is provided in the picture below.

FASTQ	Field	FASTA
@HWUSI-EAS100R:6:73:941:1973#0/1	*Header (Unique ID plus other information). Only the first character is standard.*	**>HWUSI-EAS100R:6:73:941:1973#0/1**
GATTTGGGGT…..	*Nucleotides sequence*	**GATTTGGGGT……**
+SRR001666.1 071112_SLXA-EAS1_s_7	*Optional description. Only the first character is standard. This field is*	Not present

	becoming obsolete and only “+” is used to separate the previous and the next field	
!''*((((*+)**	*Quality scores*	Not present

Table 1 - FastQ and FastA are structured as sequences of four and two fields representing each sequence read.

Both file formats start with a header field where only the first character (“@“ for FastQ and “>” for FastA) is standardized to signal the start of a new read. The remaining text in the header usually identifies the originating experiment, the type of sequencing machine or technology adopted and other information aiming at identifying the source of the data.

The second field contains the symbols used to represent nucleotides in both FastA and FastQ. They are usually 5 types of symbols:

- A, C , G, T (T is replaced by U in case of RNA sequencing)
- A fifth symbol “N” used when the sequencing machine cannot take any decision.

FastQ has two additional fields:

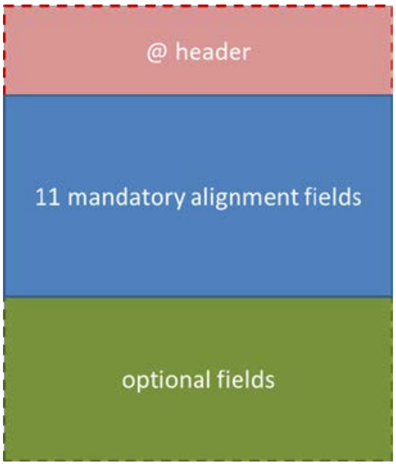
- An optional container of additional metadata starting with “+”
- Quality scores expressing the level of confidence for each nucleotide encoded in the second field. The value and meaning of each symbol vary with the sequencing machine adopted.

3.2 The SAM and BAM file formats

This section provides a summary of the SAM v1 specification, more details can be found in the document available online: http://samtools.github.io/hts-specs. A good summary of SAM feature is available on this SAM wiki entry as well.

SAM is a TAB-delimited text format consisting of

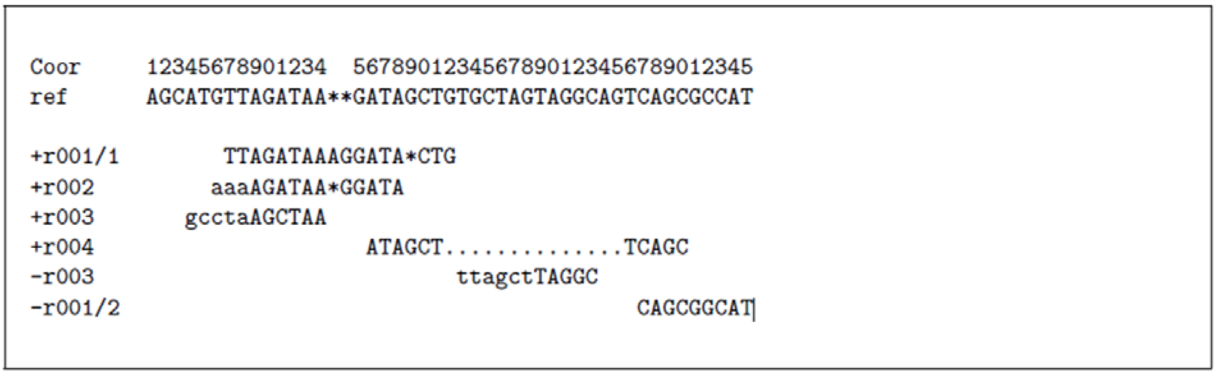
- an optional header section starting with ‘@’
- an alignment section including
  - 11 mandatory fields
  - variable number of optional fields



If present, the header must be prior to the alignments.

Figure 4 and Figure 5 show how some sequence reads are formatted in SAM. The example is taken from the SAM v1 specification and includes:

- read001/1 and read001/2 representing a read pair;
- r002 is a single read;
- r003 is a chimeric read;
- r004 represents a split alignment (a read which needs to be split in order to properly be mapped to the reference genome).

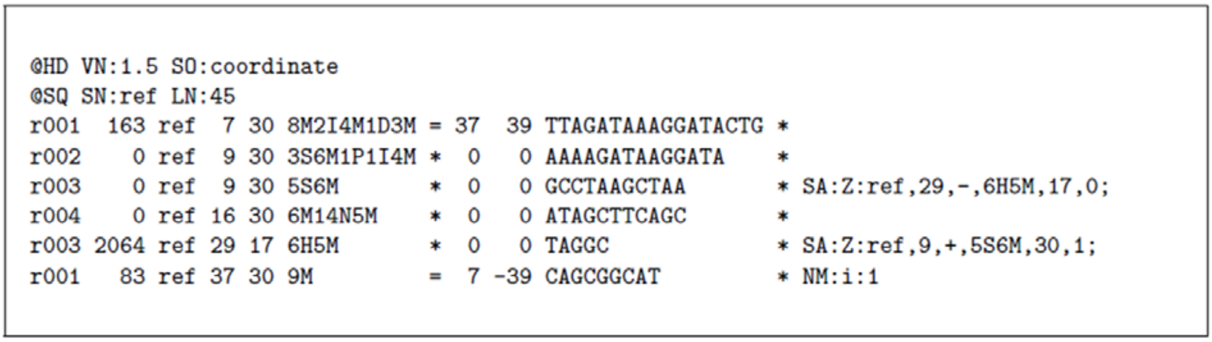


```
Coor     12345678901234  5678901234567890123456789012345
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

+r001/1        TTAGATAAAGGATA*CTG
+r002         aaaAGATAA*GGATA
+r003       gcctaAGCTAA
+r004                     ATAGCT..............TCAGC
-r003                            ttagctTAGGC
-r001/2                                        CAGCGGCAT
```

Figure 1 – The four reads aligned to a reference genome (“ref”)

Figure 4 simply shows the four reads mapped to a reference genome (“ref” on top of the picture). Nucleotides symbols in uppercase identify bases that match to the reference genome while not matching nucleotides are represented in lowercase. Dots represent a gap (unknown sequence) which separates a split alignment.



```
@HD VN:1.5 SO:coordinate
@SQ SN:ref LN:45
r001  163 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001   83 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

Figure 2 - The alignment of Figure 4 formatted as SAM file (only the 11 mandatory columns are used here)

3.2.1 SAM Terminology

Template	A DNA/RNA sequence part of which is sequenced on a sequencing machine or assembled from raw sequences.
Segment	A contiguous sequence or subsequence.
Read	A raw sequence that comes off a sequencing machine. A read may consist of multiple segments. For sequencing data, reads are indexed by the order in which they are sequenced.
Linear alignment	An alignment of a read to a single reference sequence that may include insertions, deletions, skips and clipping, but may not include direction changes (i.e. one portion of the alignment on forward strand and another portion of alignment on reverse strand). A linear alignment can be represented in a single SAM record (e.g. r002 and r004 in the example above).
Chimeric alignment	An alignment of a read that cannot be represented as a linear alignment. A chimeric alignment is represented as a set of linear alignments that do not have large overlaps (e.g. r003 in the example above is composed by two linear alignments). Typically, one of the linear alignments in a chimeric alignment is considered the “representative" alignment and the others are called “supplementary" and are distinguished by the supplementary alignment flag.
Read alignment	A linear alignment (1 SAM record) or a chimeric alignment (several SAM records) that is the complete representation of the alignment of the read.

Multiple mapping	The correct placement of a read may be ambiguous, e.g. due to repeats. In this case, there may be multiple read alignments for the same read. One of these alignments is considered primary. All the other alignments are considered “secondary”. Typically the alignment designated primary is the best alignment, but the decision may be arbitrary.
Phred scale	Given a probability $0 < p \leq 1$, the phred scale of p equals $-10 \log_{10} p$, rounded to the closest integer.

3.2.2 The SAM header

The SAM specification states that “each header line begins with character `@' followed by a two-letter record type code. In the header, each line is TAB-delimited and except the @CO lines, each data field follows a format `TAG:VALUE' where TAG is a two-letter string that denes the content and the format of VALUE.”

The SAM header is optional, but when present it has some mandatory fields that are briefly introduced here. For the complete specification of both mandatory and optional fields please refer to the SAM Format Specification document.

Record	Sub-record	Description
@HD		This is the first header line.
	VN	Format version. Accepted format: `/^[0-9]+\.[0-9]+$/.`
@SQ		Reference sequence dictionary. The order of @SQ lines denes the alignment sorting order.
	SN	Reference sequence name. Each @SQ line must have a unique SN tag. The value of this field is used in the alignment records in RNAME and PNEXT fields. Regular expression: `[!-)+-<>-~][!-~]*`
	LN	Reference sequence length. Range: [1, 2^{31}-1]
@RG		
	ID	Read group identifier. Each @RG line must have a unique ID. The value of ID is used in the RG tags of alignment records. Must be unique among all read groups in header section. Read group IDs may be modified when merging SAM les in order to handle collisions.
@PG		Program (used to manipulate the)
	ID	Program record identifier. Each @PG line must have a unique ID. The value of ID is used in the alignment PG tag and PP tags of other @PG lines. PG IDs may be modified when merging SAM files in order to handle collisions.

Table 2 – SAM header mandatory fields

3.2.3 The alignment section: mandatory fields

In the SAM format, each alignment line typically represents the linear alignment of a segment. Each line has 11 mandatory fields. These fields always appear in the same order and must be present, but their values can be `0' or `*' (depending on the field) if the corresponding information is unavailable. The following table gives an overview of the mandatory fields in the SAM format:

Col	**Field**	**Type**	**Regexp/Range**	**Brief description**
1	QNAME	String	[!-?A-~]{1,255}	Query template name
2	FLAG	Int	[0,2^{16} -1]	bitwise flag

<table>
<tr><td>3</td><td>RNAME</td><td>String</td><td>*|[!-()+-<>-~][!-~]*</td><td>Reference sequence NAME</td></tr>
<tr><td>4</td><td>POS</td><td>Int</td><td>[0, 2^{31}-1]</td><td>1-based leftmost mapping POSition</td></tr>
<tr><td>5</td><td>MAPQ</td><td>Int</td><td>[0, 2^{8}-1]</td><td>MAPping Quality</td></tr>
<tr><td>6</td><td>CIGAR</td><td>String</td><td>*|([0-9]+[MIDNSHPX=])+</td><td>CIGAR string</td></tr>
<tr><td>7</td><td>RNEXT</td><td>String</td><td>*|=|[!-()+-<>-~][!-~]*</td><td>Ref. name of the mate/next read</td></tr>
<tr><td>8</td><td>PNEXT</td><td>Int</td><td>[0,2^{31} -1]</td><td>Position of the mate/next read</td></tr>
<tr><td>9</td><td>TLEN</td><td>Int</td><td>[-2^{31}+1, 2^{31}-1]</td><td>observerd Template LENgth</td></tr>
<tr><td>10</td><td>SEQ</td><td>String</td><td>*|[A-Za-z=.]+</td><td>segment SEQuence</td></tr>
<tr><td>11</td><td>QUAL</td><td>String</td><td>[!-~]+</td><td>ASCII of Phred-scaled base QUALity+33</td></tr>
</table>

Table 3 – SAM alignment section mandatory fields

3.2.4 Compressed SAM: BAM

The BAM file format is the binary equivalent of SAM obtained by compressing SAM using the BGZF (Blocked GNU Zip Format) compression tool.

BGZF implements block compression on top of the standard gzip file format with the goal of both providing good compression and allowing efficient random access to the BAM file.

A BGZF file is a series of concatenated BGZF blocks. Each BGZF block is itself a spec-compliant gzip archive which contains an “extra field" in the format described in RFC1952.

BAM files are essentially composed by a concatenation of BGZF compressed data blocks that can be randomly accessed via a BAM file index that uses *virtual offsets* into the BGZF file. Each virtual file offset is an unsigned 64-bit integer, defined as: `coffset<<16|uoffset`, where `coffset` is an unsigned byte offset into the BGZF file to the beginning of a BGZF block, and `uoffset` is an unsigned byte offset into the uncompressed data stream represented by that BGZF block. More details on the BAM file structure can be found in the SAM/BAM Format specification [8].

4 Public repositories

The International Nucleotide Sequence Database Collaboration consists of a joint effort to collect and disseminate genomic information and it is currently accepting 3 file formats for sequencing data to be submitted by third parties: CRAM, BAM, and FastQ.

Tools are available to transcode data from one file format to another without loss of information, but difference exist among them in terms of supported functionality.

Other important public repositories include the 1000 Genomes Project (the largest contributor to the INSDC initiative mentioned earlier) and the Gene Expression Omnibus (GEO) managed by the US National Center for Biotechnology Information (NCBI).

5 Data classes

In order to make the dataset statistically meaningful the following sequencing data with different characteristics have been considered.

Sequencing technologies

- Illumina Genome Analyzer®,
- Pacific Biosciences SMRT®
- Oxford Nanopore
- Ion Semiconductor (Life Technologies),

Organisms

- Homo Sapiens *(several coverage levels)*
- Bacteria
- Plants

Type of experiment

- Metagenomic
- Cancer cell lines

6 Obtaining the data

After the 111th MPEG meeting in Geneva where a few hard drives provided by interested people were filled with the data described in this document, it will be possible to organize the shipment of a HDD from Lausanne (CH).
Anyone interested in getting the data should contact Claudio Alberti: claudio.alberti@epfl.ch.
The average cost of the HDD + shipment can be estimated at around 250 USD.

Further work on the dataset will be discussed on the AhG email reflector:
genome_compression@listes.epfl.ch

7 Proposed dataset

7.1 Data formats

Unmapped sequences are provided in the form of gzipped FastQ files with the exception of Oxford Nanopore data that are only available as Fast5 files. This format can be transcoded to FastQ using publicly available tools such as fast5tofastq or poretools.
FastQ files are usually manipulated and parsed with custom scripts based on the most popular scripting languages such as bash, python, perl etc.

Mapped sequences are provided in the form of BAM files together with the reference genome used for mapping and in some cases the index file (.BAI) that was made available together with the BAM files. Indexes are used to support random access to BAM files. In case an index is not available with the BAM files, it can be created following the instructions provided in Appendix A.

ID	Sequencing method	Size	File type	Coverage	Origin	Comments
Homo Sapiens						
ERP001775	Illumina HiSeq	~2 TB	FastQ	200x	http://www.ebi.ac.uk/ena/data/view/ERP001775	This is the largest dataset. Can be transcoded from gzip to 7zip if needed
ERP001960	Illumina HiSeq	~120 GB each genome	BAM	30x	http://www.ebi.ac.uk/ena/data/view/ERP001960	3 genomes selected SAMEA1573614 SAMEA1573618 SAMEA1573617
ERP002490	Illumina HiSeq	265 GB	BAM	30-40x	http://www.ebi.ac.uk/ena/data/view/ERP002490	The insert size (distance between the pair of reads) is much larger than usual (about 2K bases instead of 300).
Low coverage ERR317482 WGS	Illumina HiSeq 2000	6.1 GB	BAM	1.9x	http://www.ebi.ac.uk/ena/data/view/ERR317482 ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/reference/phase2_reference_assembly_sequence/hs37d5.fa.gz	Also used as a low coverage test in the Scramble paper.
Low coverage NA21144.chrom11	Illumina HiSeq 2000	1 GB	BAM	7.5x	ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data/NA21144/alignment/NA21144.chrom11.ILLUMINA.bwa.GIH.low_coverage.20130415.bam ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/reference/phase2_reference_assembly_sequence/hs37d5.fa.gz	Used in Scramble paper. Processed through the GATK pipeline, which makes the auxiliary data bulkier. (Can be stripped off easily if desired.)

PacBio_CHM1htert_54x	Pacific Biosciences SMRT	150 GB	FastQ	54x	http://datasets.pacb.com/2014/Human54x/fast.html	Available as zipped FastQ
IonTorrent	Ion Torrent	1.3 GB each file	BAM		http://www.ebi.ac.uk/ena/data/view/ERX276880 http://www.ebi.ac.uk/ena/data/view/ERX276881 http://www.ebi.ac.uk/ena/data/view/ERX276882	http://www.ebi.ac.uk/ena/data/warehouse/search?query=%22instrument_platform=%22ION_TORRENT%22%20%20AND%20submitted_format=%22BAM%22%22&domain=read
RNAseq		16 GB	BAM		http://www.ebi.ac.uk/arrayexpress/files/E-MTAB-1728/K562_ cytosol_LID8465_TopHat_v2.bam	
Bacteria						
ERX593919 (E. Coli)	Oxford Nanopore	60 GB	Gzipped Fast5		http://www.ebi.ac.uk/ena/data/view/ERX593919	To be converted from Fast5
ERX593921 (E. Coli)	Oxford Nanopore	46 GB	Gzipped Fast5		http://www.ebi.ac.uk/ena/data/view/ERX593921	To be converted from Fast5
DH10B (E.Coli)	Illumina	1.3 GB	BAM		ftp://webdata:webdata@ussd-ftp.illumina.com/Data/SequencingRuns/DH10B/MiSeq_Ecoli_DH10B_110721_PF.bam https://raw.githubusercontent.com/allanroscoche/PathTree/master/data/DH10B_WithDup_FinalEdit_validated.fasta	Used in the Deez paper.
ERA269036 9799_7#3.bam (E.Coli)	Illumina	2.3 GB	BAM		ftp://ftp.sra.ebi.ac.uk/vol1/ERA269/ERA269036/bam/ http://www.ncbi.nlm.nih.gov/nuccore/NC_000913.2?report=fasta&format=text	Used in the Scramble paper.
Metagenomic						
Human gut	Illumina Genome Analyzer II	10 GB	FastQ		http://www.ebi.ac.uk/ena/data/view/ERA000116	3 samples picked SAMEA728920 SAMEA728635 SAMEA728854
Cancer cell lines						
Mutation/Variation Calling Benchmark 4 at CGHub		255 GB	BAM	60x	https://cghub.ucsc.edu/datasets/benchmark_download.html	TCGA BENCHMARK CELL LINE: HCC1143 NORMAL 60x f0eaa94b-f622-49b9-8eac-e4eac6762598

Mutation/Variation Calling Benchmark 4 at CGHub		305 GB	BAM	50x	https://cghub.ucsc.edu/datasets/benchmark_download.html	TCGA BENCHMARK CELL LINE: HCC1143 TUMOR 50x ad3d4757-f358-40a3-9d92-742463a95e88
Mutation/Variation Calling Benchmark 4 at CGHub		130 GB	BAM		https://cghub.ucsc.edu/datasets/benchmark_download.html	UCSC ARTIFICIAL MIXED SAMPLE: 80% HCC1954BL 20% HCC1954 360b4736-6c5e-48df-af58-c1cf51609350
Plants						
T. Cacao	Illumina	8.2 GB	FastQ	10x	http://www.ncbi.nlm.nih.gov/sra/SRX288435	

8 References

[1] “SamTools,” [Online]. Available: http://samtools.sourceforge.net/.

[2] “Cram Toolkit,” ENA European Nucleotide Archive, [Online]. Available: http://www.ebi.ac.uk/ena/software/cram-toolkit.

[3] Genome Research Limited, “Samtools,” Genome Research Limited, [Online]. Available: http://www.htslib.org/doc/samtools-1.2.html.

[4] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis and R. Durbin, “The Sequence Alignment/Map format and SAMtools,” *Bioinformatics,* 2009.

9 Appendix A - SAMtools for SAM/BAM manipulation

Once the reads are aligned and expressed as SAM or BAM files, they can be manipulated by the SAMtools toolkit [1].The CRAM toolkit [2] is fully compatible with SAMtools and provides the same functionality together with a more sophisticated support to compression.
The paper describing the SAM format [4] is one of the best starting point to understand the file format and its usage.
This document lists only the basic operations needed to access the information contained in the BAM files, for more advanced features please refer to the SAMtools documentation [3] or other SAMtools tutorials available online.

9.1 Getting the basic tools

The toolset needed for files manipulation includes the **htslib** libraries that can be found here

https://github.com/samtools/htslib/tree/master

HTSlib is an implementation of a unified C library for accessing common file formats, such as SAM, CRAM and VCF, used for high-throughput sequencing data, and is the core library used by SAMtools and bcftools. HTSlib only depends on zlib. It is known to be compatible with gcc, g++ and clang.

SAMtools is then needed to acces the file content

https://github.com/samtools/samtools

After having downloaded and compiled htslib and SAMtools, you will be able to run the basic commands listed below to access the content of Sam and BAM files.

9.2 View SAM/BAM content

SAM is a textual file format and can be view/edited with and text editor. SAMtools provide a command to visualize compressed BAM files as well as plain SAM files.

```
samtools view aligned_reads.sam | more
```

```
samtools view aligned_reads.bam | more
```

In order to search any specific entry in the BAM file usually the `samtools view` command is piped with `awk` or `grep` to find the occurrences of strings.

`samtools view` supports filtering according to specific match criteria. For example this command

```
samtools view -f 4 aligned_reads.bam | more
```

extracts only those reads with flag value of 4, i.e. reads that fail to map to the reference genome.

The exact same command with a –F option would have removed all matching reads.

You can also access a specific segment of a larger BAM file

```
samtools view aligned_reads.bam -region chr9:5000000-5001000
```

9.3 SAM to BAM compression

Once the raw reads are aligned using one of the tools listed in section 5.5.1 and encoded as a SAM file, one common step is to convert the textual SAM to BAM as all the downstream steps of a genomic analysis pipeline require BAM as input.
The syntax of the conversion from SAM to BAM is the following:

```
samtools view -b -S -o aligned_reads.bam aligned_reads.sam
```

```
-b: indicates that the output is BAM.

-S: indicates that the input is SAM.

-o: specifies the name of the output file.
```

9.4 Sorting BAM

Since many programs used in downstream analysis pipelines only accept sorted BAM files, another important BAM manipulation is sorting:

```
samtools sort -m 1000000000 aligned_reads.bam outputPrefix
```

-m specifies the maximum memory to use and the output will be a file named `outputPrefix.bam`

9.5 Indexing BAM

BAM files can also have a companion file, called an index file. This file has the same name as the originating BAM, suffixed with .bai. This file acts like an external table of contents, and allows programs to jump directly to specific parts of the bam file without reading through all of the sequences.
The index is generated using the following command:

```
samtools index aligned_reads.bam
```