Supplementary Material

Table of Contents

1 SAM Fields/Columns

A SAM file is structured into the SAM header (HEAD) and an alignment section, made up of SAM records. Each SAM record occupies one line in the SAM file and is made up of 12 fields/columns which are separated by tab stops. The names to identify the columns in Table 1 have been chosen according to the SAM format specification [1].

Table 1: SAM fields/columns

Field/Column	Name	Remarks/Evaluation Category
1	QNAME	Ident
2	FLAG	Aux
3	RNAME	Nuc
4	POS	Nuc

5	MAPQ	Aux
6	CIGAR	Nuc
7	RNEXT	Paired
8	PNEXT	Paired
9	TLEN	Paired
10	SEQ	Nuc
11	QUAL	Qual
12	OPT	Aux
--	CRTL	Tabs (\t) and line feeds (\n)
--	HEAD	SAM header

2 Evaluation Categories

In order to compare different SAM compression tools, SAM fields be associated with categories as shown in Table 2.

Table 2: Evaluation categories

Category	Associated SAM Fields
Aux	FLAG, MAPQ, OPT
Ident	QNAME
Nuc	RNAME, POS, CIGAR, SEQ
Paired	RNEXT, PNEXT, TLEN
Qual	QUAL

3 Tools and Codec Categories

The tools used are summarized in Table 3.

Table 3: Tool comparison

Tool	Available	Reference-based	Random Access	Remark
Tsc v1.0		N	Y	
Quip v1.1.8		N	N	Does not preserve RNEXT and OPT
Quip v1.1.8 –a		N	N	Does not preserve RNEXT and OPT
Quip v1.1.8 –r		Y	N	Does not preserve RNEXT and OPT
DeeZ v1.0		Y	Y	
Sam_comp v0.7		Y/N	N	Does not preserve RNEXT, PNEXT, TLEN, and OPT
CRAM v2.0		Y	Y	

3.1 Tsc v1.0

The software is structured into codec categories that have been assigned to evaluation categories as shown in Table 4.

Table 4: Tsc codec categories

Codec Category	Input	Evaluation Category
Aux	FLAG, MAPQ, OPT	Aux
Ident	QNAME	Ident
Nuc	RNAME, POS, CIGAR, SEQ	Nuc

Paired	RNEXT, PNEXT, TLEN	Paired
Qual	QUAL	Qual

3.2 Quip v1.1.8

Quip [2].

The software is structured into codec categories that have been assigned to evaluation categories as shown in Table 5. Quip does not preserve RNEXT and OPT.

Table 5: Quip codec categories

Codec Category	**Input**	**Evaluation Category**
ID	QNAME	Ident
Aux	FLAG, RNAME, POS, MAPQ, CIGAR, PNEXT, TLEN	Aux/Nuc/Paired
Seq	SEQ	Nuc
Qual	QUAL	Qual
--	RNEXT, OPT	Aux/Paired

3.3 DeeZ v1.0

DeeZ [3].

The software is structured into codec categories that have been assigned to evaluation categories as shown in Table 5.

Table 6: DeeZ codec categories

Codec Category	**Input**	**Evaluation Category**
sequence	POS, CIGAR, SEQ, RNAME	Nuc
editOp	POS, CIGAR, SEQ	Nuc
readName	QNAME	Ident
mapFlag	FLAG	Aux
mapQual	MAPQ	Aux
quality	QUAL	Qual
pairedEnd	RNEXT, PNEXT, TLEN	Paired
optField	OPT	Aux

3.4 Sam_comp v0.7

Sam_comp [4].

The software had to be modified in order to obtain compression statistics. The modifications in the source code have been marked with the string {vogesMod}. To track the modifications, a Git repository has been set up at the Institut fuer Informationsverarbeitung. The program version used for the evaluation lives in the branch “vogesMod”.

```
$ git clone git@git.tnt.uni-hannover.de:voges/sam_comp-0.7.git
$ git checkout vogesMod
```

Sam_comp is structured into different codecs as shown in Table 7. When modifying the sam_comp source code, the codec category “diffcol” has been added for coding operations concerning POS, CIGAR, and SEQ that have not been delegated to a distinct function. The codec category “len” codes the read lengths and is thus assigned to the evaluation category “Nuc”.

However, sam_comp does not preserve the SAM fields RNEXT, PNEXT, TLEN and OPT.

Table 7: Sam_comp codec categories

Codec Category	Input	Evaluation Category
len	--	Nuc
rname	RNAME	Nuc
pos	POS	Nuc
mapQ	MAPQ	Aux
flags	FLAG	Aux
name2	QNAME	Ident
qual	QUAL	Qual
seq8	SEQ	Nuc
cigar	CIGAR	Nuc
consensus	POS, CIGAR, SEQ	Nuc
diffcol	POS, CIGAR, SEQ	Nuc
--	RNEXT, PNEXT, TLEN	Paired
--	OPT	Aux

3.5 CRAM 2.0

4 Tool Invocation Parameters

4.1 Tsc v1.0

The option -s enables printing detailed statistics after compression.

```
$ tsc -s file.sam -o file.sam.tsc
$ tsc –d -s file.sam.tsc –o file.sam.tsc.sam
```

4.2 Quip v1.1.8

The options -lv print detailed statistics for the compressed .qp file.

```
$ quip –lv file.sam.qp
```

4.2.1 Non-reference-based mode

```
$ quip file.sam -c > file.sam.qp
$ quip –d --output=sam file.sam.qp –c > file.sam.qp.sam
```

4.2.2 De-novo assembly mode

```
$ quip –a file.sam -c > file.sam.qp
$ quip –d --output=sam file.sam.qp –c > file.sam.qp.sam
```

4.2.3 Reference-based mode

```
$ quip –r ref.fa file.sam -c > file.sam.qp
$ quip –d –r ref.fa --output=sam file.sam.qp –c > file.sam.qp.sam
```

4.3 DeeZ v1.0

First, we have to build an index for the reference FASTA file using SAMtools. Alternatively, DeeZ can build its own index file.

```
$ samtools faidx ref.fa
```

The option -r indicates the reference to be used; option –S enables the printing of detailed compression statistics.

```
$ deez –S –r ref.fa file.sam –o file.sam.dz
$ deez –d –r ref.fa file.sam.dz –o file.sam.dz.sam
```

4.4 Sam_comp v0.7

Sam_comp has been run in the non-reference-based mode. The option -v enables the {vogesMod} modifications during compression.

```
$ sam_comp -v < file.sam > file.sam.zam
$ sam_comp –d < file.sam.zam > file.sam.zam.sam
```

4.5 CRAM v2.0

First, we have to build an index for the reference FASTA file using SAMtools.

```
$ samtools faidx ref.fa
```

Explain options "-Q", "-n", and "—capture-all-tags".

```
$ java –jar cramtools-2.0.jar cram \
    -I file.sam --input-is-sam     \
    -O file.sam.cram               \
    -R ref.fa                      \
    -Q –n --capture-all.tags
$ java –jar cramtools-2.0.jar bam  \
    -I file.sam.cram               \
    -R ref.fa                      \
    --print-sam-header             \
    > file.sam.cram.sam
```

5 Benchmarking

All measurements have been performed on an Intel® Core™ i7-3770K CPU with 8 cores @ 3.50 GHz and 32 GB RAM. Timing and memory measurements have been performed by prepending `/usr/bin/time –v` to the mentioned commands.

6 Datasets

For the evaluation, already aligned data from the MPEG genome compression database [5] has been used. Before invoking the mentioned compression tools, the data was converted from BAM to SAM using SAMtools. The option –h ensures that the SAM header is preserved during BAM-to-SAM conversion.

```
$ samtools view –h file.bam > file.sam
```

An overview about the dataset is given in Table 8.

Table 8: Data used for evaluation

Identifier	URL	Reference	Size

7 References

[1] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin,

and 1000 Genome Project Data Processing Subgroup, “The Sequence Alignment/Map format and SAMtools,” *Bioinformatics*, vol. 25, no. 16, pp. 2078–2079, 2009.

[2] D. C. Jones, W. L. Ruzzo, X. Peng, and M. G. Katze, “Compression of next-generation sequencing reads aided by highly efficient de novo assembly,” *Nucleic Acids Res.*, vol. 40, no. 22, pp. e171–e171, 2012.

[3] F. Hach, I. Numanagić, and S. C. Sahinalp, “DeeZ: reference-based compression by local assembly.,” *Nat. Methods*, vol. 11, no. 11, pp. 1082–4, Nov. 2014.

[4] J. K. Bonfield and M. V. Mahoney, “Compression of FASTQ and SAM Format Sequencing Data,” *PLoS One*, vol. 8, no. 3, p. e59190, 2013.

[5] C. Alberti, M. Mattavelli, L. Chiariglione, I. Xenarios, N. Guex, H. Stockinger, T. Schuepbach, P. Kahlem, C. Iseli, D. Zerzion, D. Kuznetsov, Y. Thoma, E. Petraglio, C. Sahinalp, I. Numanagic, and J. Delgado, “Database for Evaluation of Genome Compression and Storage,” Geneva, 2015.