Overview of analysis workflows and file types

Beginners session - TLUK Ribosome Profiling Workshop

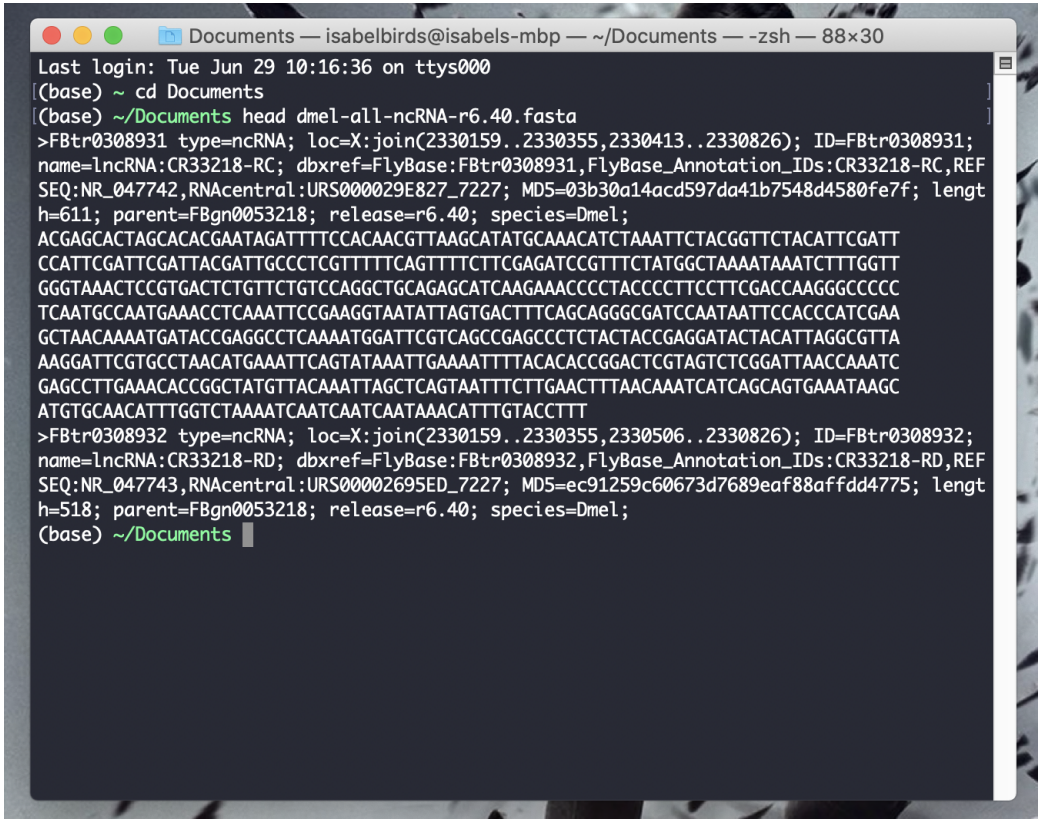
Isabel Birds

02/07/2021

File formats

Text files

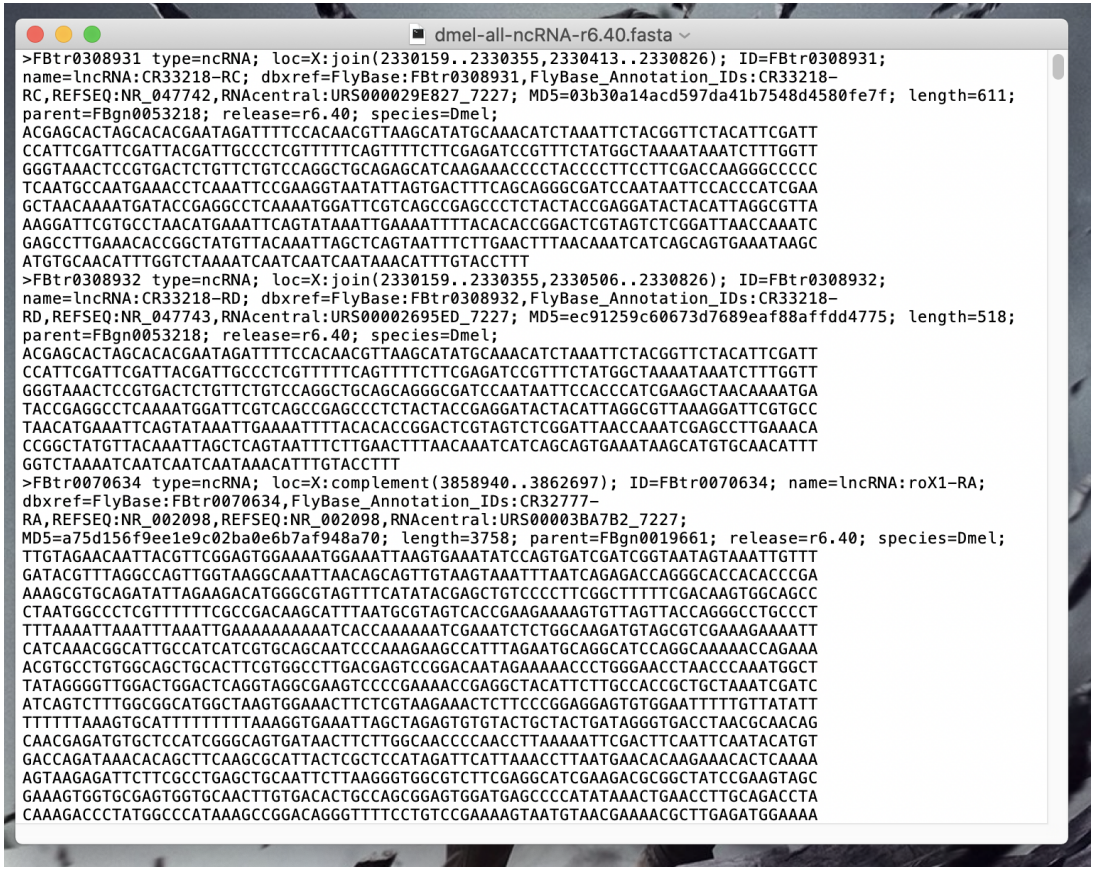
- Plain text - a file containing only text.
- Rich text - a file which includes text formatting (e.g. bold text).



Plain text file in the terminal.

Text files

- Plain text - a file containing only text.
- Rich text - a file which includes text formatting (e.g. bold text)

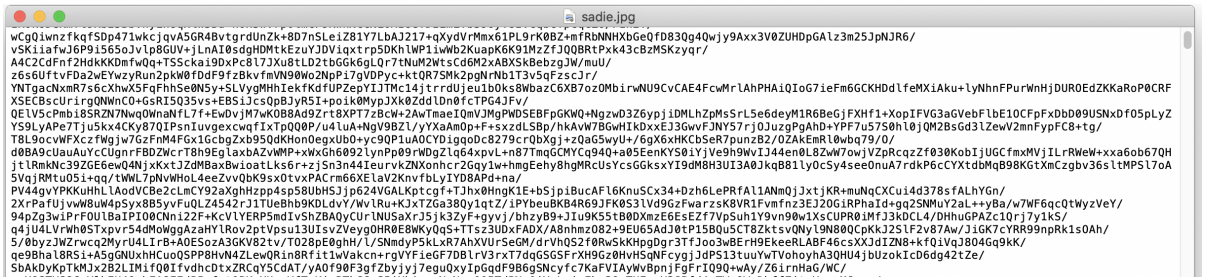


Plain text file viewed in text edit.

File formats

Binary files:

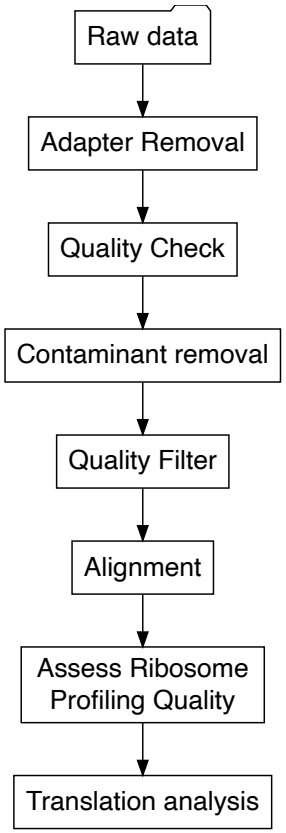
Sadie!



Binary file (Sadie.jpg) viewed in text edit.

Workflow Overview

Overview


Raw data
Adapter Removal
Quality Check
Contaminant removal
Quality Filter
Alignment
Assess Ribosome Profiling Quality
Translation analysis

File types

FASTA

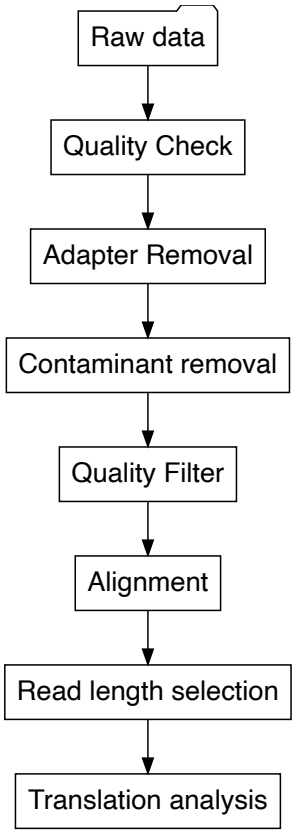
- DNA, RNA, and protein sequences are stored in FASTA format.
- This could be a single transcript, or an entire genome.
- FASTA files generally have the extension .fa or .fasta (eg file1.fa).
- Multiple sequence alignments are also stored in FASTA format, with dashes (-) used to indicate gaps in the alignment.

Example:

```
>ENST00000607096.1|ENSG00000284332.1|-|-|MIR1302-2-201|MIR1302-2|138|miRNA|
GGATGCCCAGCTAGTTTGAATTTTAGATAAACAACGAATAATTTCGTAGCATAAATATGT
CCCAAGCTTAGTTTGGGACATACTTATGCTAAAAAACATTATTGGTTGTTTATCTGAGAT
TCAGAATTAAGCATTTTA
```

The Sequence ID must be unique, and should not contain spaces.

FASTA files in Ribo-seq

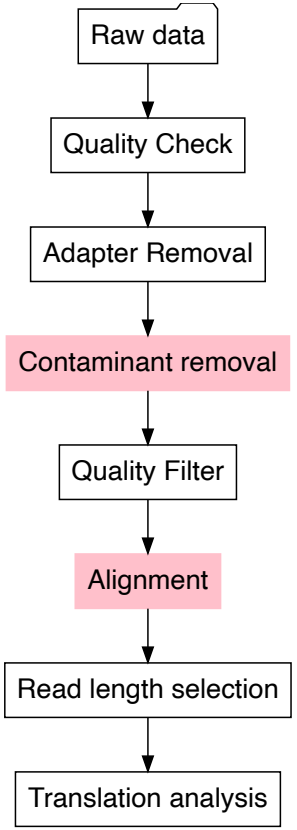


FASTA files in Ribo-seq

Drosophila melanogaster rRNA:

```
>gi|174298|gb|M25016.1|DRORR5SEM_D.melanogaster_5S_rRNA
GCCAACGACCATACCACGCTGAATACATCGGTTCTCGTCCGATCACCGAAATTAAGC
AGCGTCGCGGGCGGTTAGTACTTAGATGGGGGACCGCTTGGGAACACCGCGTGTTGT
TGGCCT
```

(NCBI 2021)



FASTQ

- FASTQ files contain sequencing data, and corresponding quality scores.
- Phred score/Q score - measures the probability a base is called incorrectly.
  - **Q-score**: $Q = -10\log_{10}(e)$
  - where e is the estimated probablitiy of the base call being wrong
  - Q = 20 - error rate 1 in 100.
  - Q = 30 - eror rate 1 in 1000.
- Quality scores are encoded, using ASCII characters to represent numerical scores.

(“NGS Sequencing Technology and File Formats,” n.d.; Peter J. A. Cock 2010)

FASTQ

Each sequence in a FASTQ file contains 4 lines:

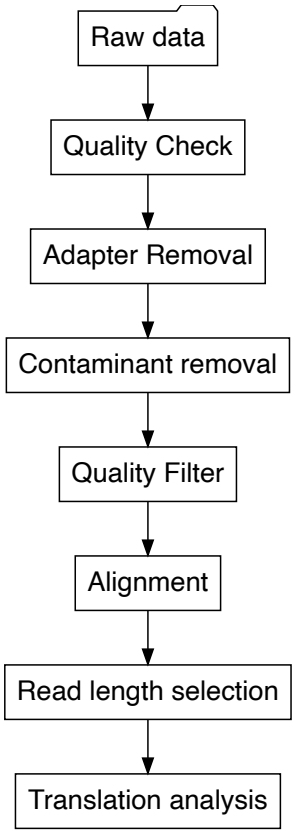
```
LINE 1: @Sequence_ID:optional description of sequencing run
LINE 2: Raw sequence letters (A,C,T,G,N)
LINE 3: + (a separator)
LINE 4: Quality scores of sequence
```

Example:

```
@NB501623:178:HJLC2BGX5:1:11101:5397:1056 1:N:0:AGTCAA
CGGTCNGTGAAGAGTCGAACGTGCTCTGCNGNAGATCGGAAGAGCACACNTCTGANCTCNAGTCACANTNANATNT
+
AAAAA#EEEEEEEEE<E/EEEEEEEE/EE#/#EE/EEEE/EE<AE/EEE#EEAE/#EAE#EEAE/EA#E#E#EE#/
```

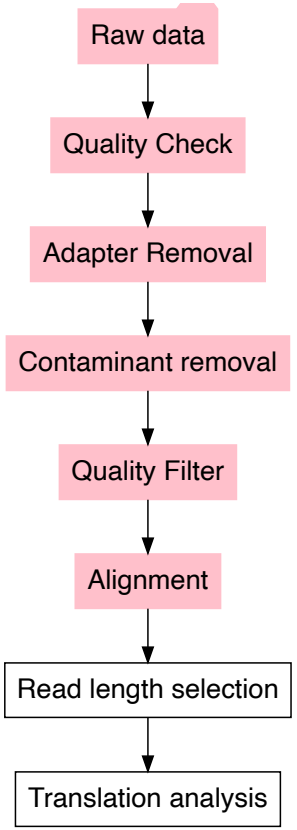
A = 32, E = 36, # = 35, < = 27

FASTQ files in Ribo-seq



FASTQ files in Ribo-seq

- Single-end seq - one fastq per sample (R1).
- Paired-end seq - two fastqs per sample (R1 and R2).



fai

- fai files are indexes for an accompanying fasta or fastq file.
- These allow efficient access to regions within sequences.
- fai files contain five tab-delimited columns for FASTA, and six for FASTQ

```
NAME   LENGTH   OFFSET   LINEBASES   LINEWIDTH    QUALOFFSET
```

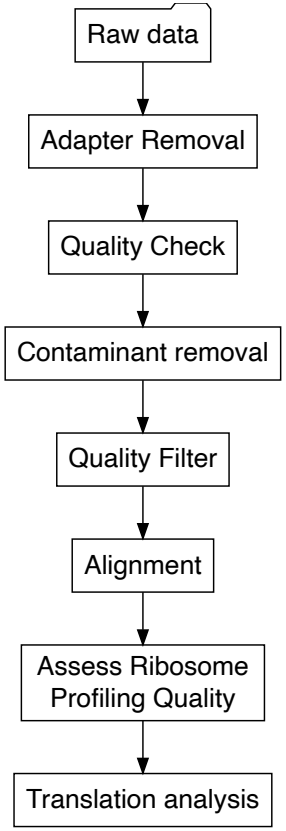
fai

```
NAME   LENGTH   OFFSET   LINEBASES   LINEWIDTH    QUALOFFSET
```

Where:

- **NAME** - sequence name
- **LENGTH** - length of the reference sequence, in bases
- **OFFSET** - Offset from the first base of the sequence
- **LINEBASES** - The number of bases on each line of the sequence
- **LINEWIDTH** - The number of bytes in each line, including the newline
- **QUALOFFSET** - Offset of sequence's first quality score

fai files in Ribo-seq

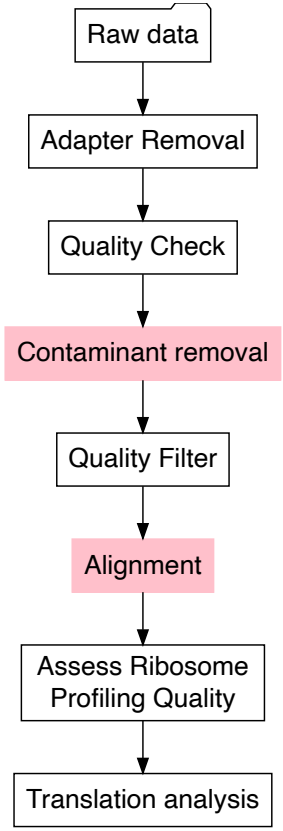


fai files in Ribo-seq

```
NAME   LENGTH   OFFSET   LINEBASES   LINEWIDTH    QUALOFFSET
```

Example: GRCh38.primary_assembly.genome.fa.fai

```
chr1     248956422    8     60   61
chr2     242193529    253105712     60   61
chr3     198295559    499335808     60   61
```



GFF3

Meme credit: @BioMickWatson (Twitter)

GFF3

- General Feature Format (GFF) files are used to describe features of sequences.
- GFF3 is now the standard - GFF2 is depreciated.
- Gene Transfer Format (GTF) files are similar to GFF3
- GFF3 files contain 9 tab-delimited columns.

```
seqid    source    type    start    end    score    strand    phase    attributes
```

GFF3

```
seqid    source    type    start    end    score    strand    phase    attributes
```

Where:

- **seqid** - the name of the sequence containing the feature
- **source** - source of the feature (a program or project)
- **type** - feature type (CDS, gene, exon)
- **start** - start position (1-base offset)
- **end** - end position (1-base offset)
- **score** - varies
- **strand** - +, -, ?, or .
- **phase** - 0,1,2, or . - where the first codon starts
- **attributes** - extra information.

GFF3

```
seqid    source    type    start    end    score    strand    phase    attributes
```

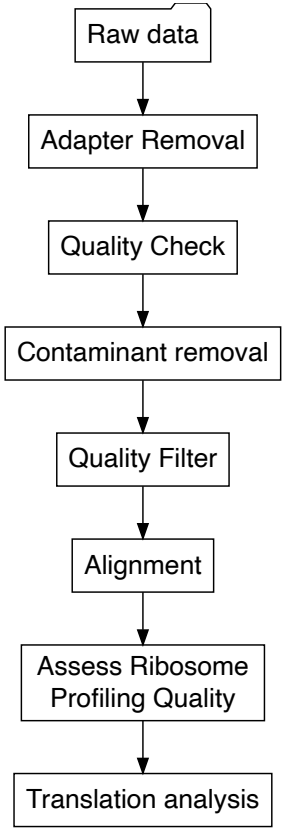
Example:

```
##gff-version 3.1.26
##sequence-region ctg123 1 1497228
ctg123 . gene            1000  9000  .  +  .  ID=gene00001;Name=EDEN
ctg123 . TF_binding_site 1000  1012  .  +  .  ID=tfbs00001;Parent=gene00001
ctg123 . mRNA            1050  9000  .  +  .  ID=mRNA00001;Parent=gene00001;Name=EDEN.1
```

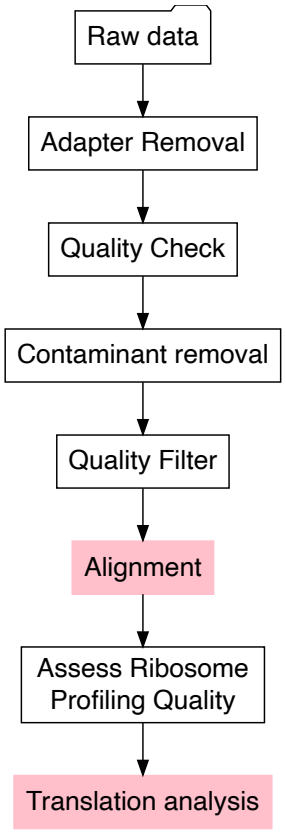
The first line is a comment that defines the version.

(Stein 2020)

GFF3 files in Ribo-seq

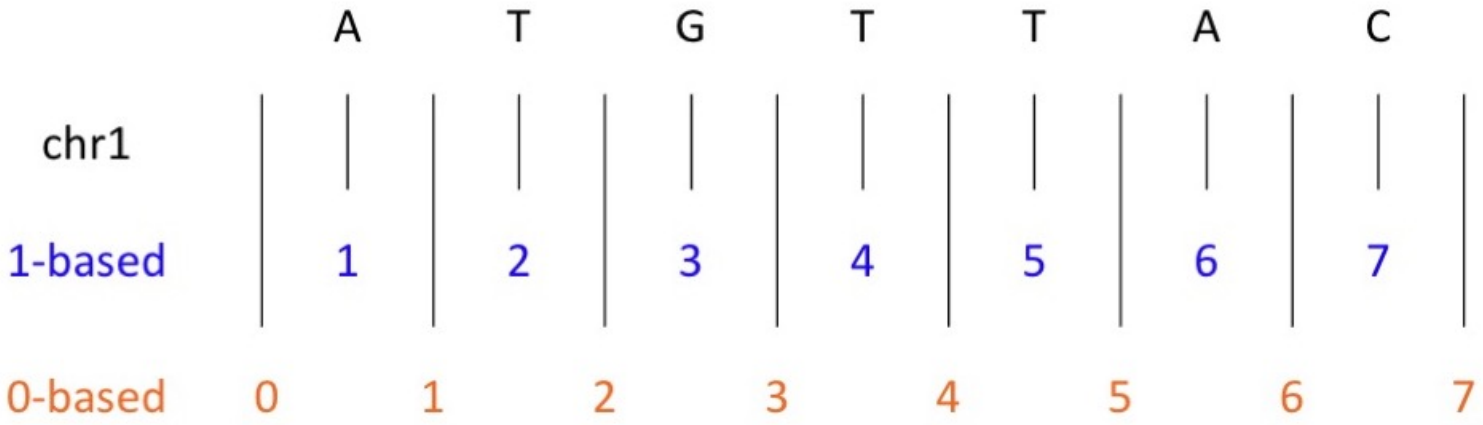


GFF3 files in Ribo-seq

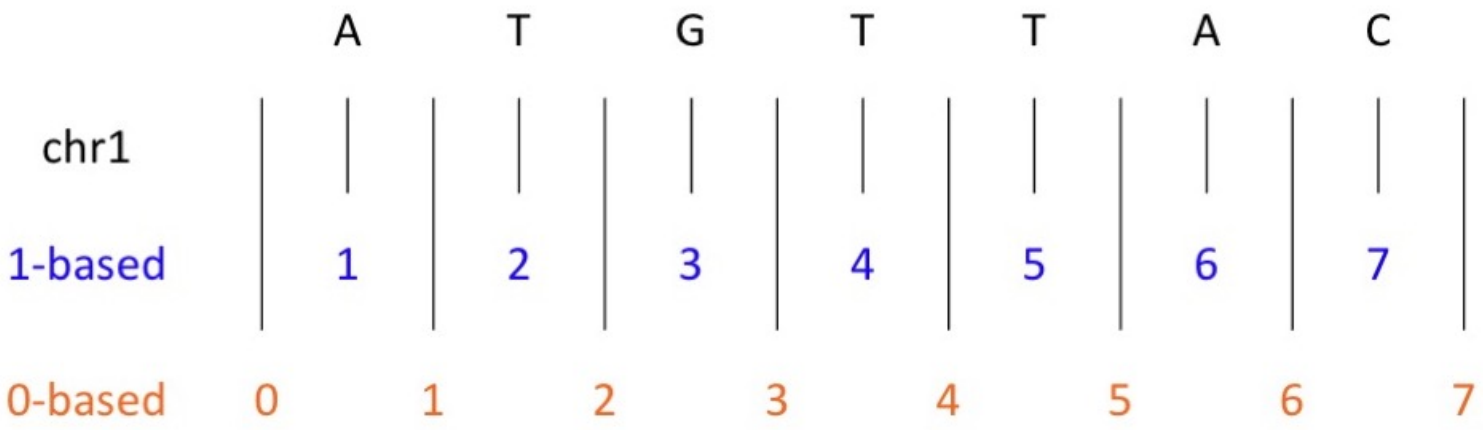


Coordinate systems

- Files are often 1-based or 0-based.
- 1-based - numbers nucleotides directly.
- 0-based - numbers between nucleotides.



Coordinate systems



Seq	1-based	0-based
ATG	chr1:1-3	chr1:0-3
C	chr1:7-7	chr1:6-7

(Griffith 2013)

Coordinate systems

1-based

- GFF
- SAM

0-based

- BED
- BAM

SAM

- Sequence Alignment Map (SAM) files contain sequence reads, and alignment data that links them to a reference sequence.
- Optional header section, followed by alignment section.

SAM - Header section

Example:

```
@HD VN:1.0  SO:unsorted
@SQ SN:gi|158246|gb|M21017.1|DRORGAB    LN:12026
@SQ SN:gi|174298|gb|M25016.1|DRORR5SEM  LN:120
@PG ID:bowtie2  PN:bowtie2  VN: CL:"bowtie2-align-s --wrapper basic-0 --threads 6
--trim3 1 -k 1 -x bowtie_indexes/rRNA_fly --passthrough -U
Quality_filter_outputs/SRR1548656.qualfilt_output.fastq"
```

@HD: File level metadata

- **VN** - format version
- **SO** - sorting order of alignments

SAM - Header section

Example:

```
@HD VN:1.0  SO:unsorted
@SQ SN:gi|158246|gb|M21017.1|DRORGAB    LN:12026
@SQ SN:gi|174298|gb|M25016.1|DRORR5SEM  LN:120
@PG ID:bowtie2  PN:bowtie2  VN: CL:"bowtie2-align-s --wrapper basic-0 --threads 6
--trim3 1 -k 1 -x bowtie_indexes/rRNA_fly --passthrough -U
Quality_filter_outputs/SRR1548656.qualfilt_output.fastq"
```

@SQ: Reference sequence metadata

- **SN** - reference sequence name
- **LN** - reference sequence length

SAM - Header section

Example:

```
@HD VN:1.0  SO:unsorted
@SQ SN:gi|158246|gb|M21017.1|DRORGAB    LN:12026
@SQ SN:gi|174298|gb|M25016.1|DRORR5SEM  LN:120
@PG ID:bowtie2  PN:bowtie2  VN: CL:"bowtie2-align-s --wrapper basic-0 --threads 6
--trim3 1 -k 1 -x bowtie_indexes/rRNA_fly --passthrough -U
Quality_filter_outputs/SRR1548656.qualfilt_output.fastq"
```

@RF: Read group - multiple lines allowed

@PG: Program

- **ID** - program ID
- **PN** - program name
- **CL** - Command line

SAM - Alignment section

- tab delimited columns
- 11 required columns
- Optional TAGs

```
QNAME FLAG RNAME POS MAPQ CIGAR RNEXT PNEXT TLEN SEQ QUAL
```

Where:

- **QNAME** - query sequence name
- **FLAG** - bitwise flag - lookup code for features of read
- **RNAME** - reference sequence name
- **POS** - leftmost mapping postion (1-based)
- **MAPQ** - mapping quality (how well the read aligned)
- **CIGAR** - cigar string

CIGAR strings

- Compact Idiosyncratic Gapped Alignment Report (CIGAR) string
- A shorthand encoding of an entire alignment:
    - Where the sequence aligns/doesn't align.
    - Deletions
    - Insertions

Example: position=2, CIGAR=3M2I3M

```
AAGTC  TAGAA (ref)
  GTCGATAG (query)
```

Starting from 2, 3 matches, 2 inserts, 3 matches.

(Fan 2017)

SAM - Alignment section

```
QNAME FLAG RNAME POS MAPQ CIGAR RNEXT PNEXT TLEN SEQ QUAL
```

Where:

- **RNEXT** - reference name of the next read
- **PNEXT** - position of the next read (1-based)
- **TLEN** - query sequence length - 0 for single segment sequence
- **SEQ** - query sequence
- **QUAL** - query sequence quality

(The SAM/BAM Format Specification Working Group 2021)

SAM - Alignment section

TAGs

Format - TAG : TYPE : VALUE

Examples:

- AS:i:score = Alignment score generated by aligner.
- H0:i:count = Number of perfect hits.

(The SAM/BAM Format Specification Working Group 2020)

SAM - Alignment section

```
QNAME FLAG RNAME POS MAPQ CIGAR RNEXT PNEXT TLEN SEQ QUAL
```

```
SRR1548656.36   0   gi|158246|gb|M21017.1|DRORGAB   2873   255 29M *   0   0
TGCTTNGACTACATATGGTTGAGGGTTGT   CCCFF#2AFHHHHJJIJJJJJIJJJHIJJ
AS:i:-1 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:5G23   YT:Z:UU
```

BAM

- BAM files are binary versions of SAM files.
- Not human readable, but much smaller files.
- .bai - index as in .fai

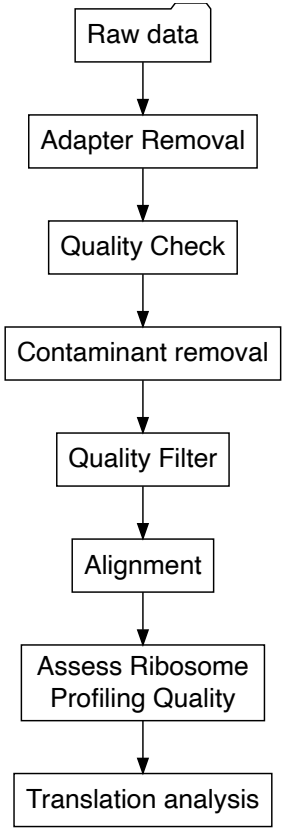
Samtools

- Convert SAM to BAM
- Sort BAM file
- Index BAM file
- View BAM file

```
samtools view sample.sorted.bam | head -n 5
```

(Genome Research Limited 2021; Danecek et al. 2021)

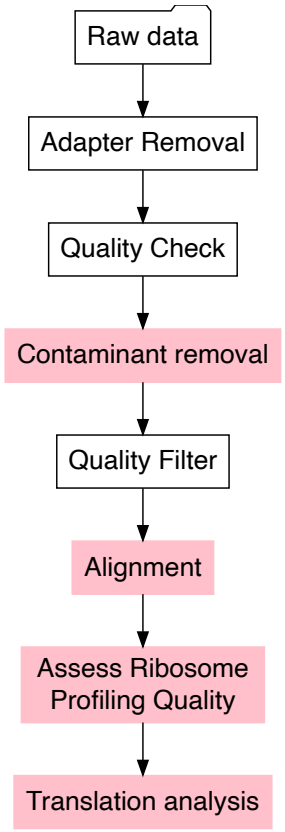
BAM/SAM files in Ribo-seq



BAM/SAM files in Ribo-seq

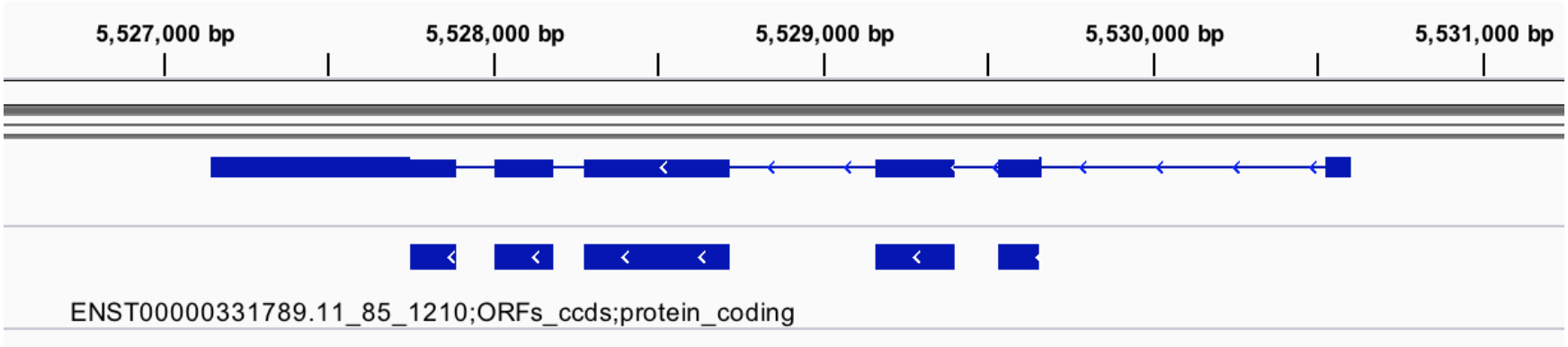
For example:

Unsorted SAM file from aligner, convert to BAM, sort and index for downstream steps.



BED

- BED files allow users to define how annotation tracks are displayed
- Upto 12 tab-delimited columns, only the first 3 are required.




```
chrom chromStart chromEnd name score strand thickStart thickEnd itemRgb blockCount
blockSizes blockStarts
```

BED

Required fields:

- **chrom** - Name of the chromosome or scaffold
- **chromStart** - Start position of chromosome (0-based)
- **chromEnd** - End postition of chromosome

BED

Optional fields

- **name** - Label for the feature
- **score** - between 0-1000, determines grayness of track
- **strand** - + or -
- **thickStart** - start position of thick blocks (start codon)
- **thickEnd** - end position of thick blocks
- **itemRgb** - colour of the data in the line
- **blockCount** - number of exons (blocks) in the line
- **blockSizes** - comma-separated list of block sizes
- **blockStarts** - comma-separated list of block starts relative to chromStart

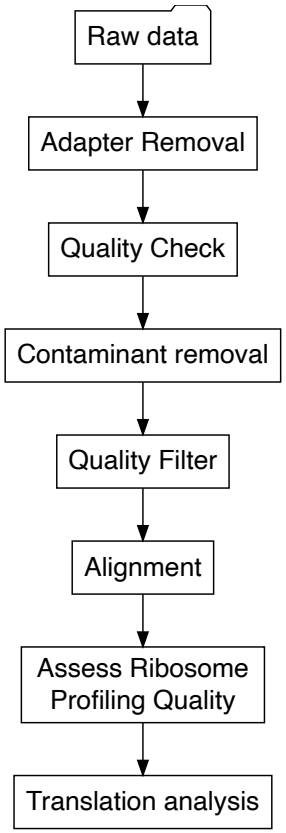
(Ensembl 2021)

BED

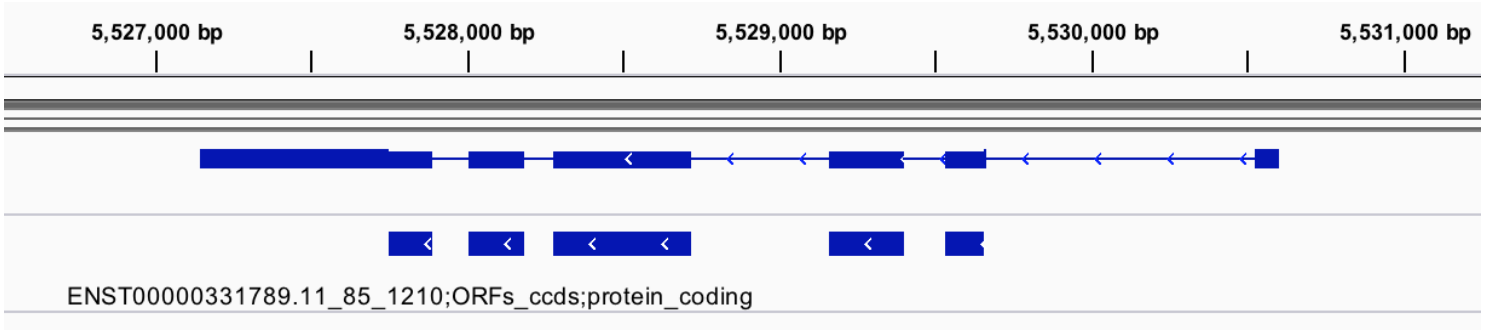
```
chrom chromStart chromEnd name  score strand thickStart thickEnd itemRgb blockCount
blockSizes blockStarts
```

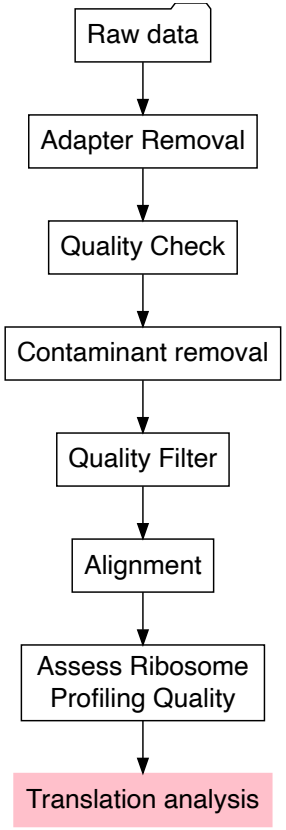
```
chr7  127471196  127472363  Pos1  0  +  127471196  127472363  255,0,0
chr7  127472363  127473530  Pos2  0  +  127472363  127473530  255,0,0
chr7  127473530  127474697  Pos3  0  +  127473530  127474697  255,0,0
chr7  127474697  127475864  Pos4  0  +  127474697  127475864  255,0,0
chr7  127475864  127477031  Neg1  0  -  127475864  127477031  0,0,255
chr7  127477031  127478198  Neg2  0  -  127477031  127478198  0,0,255
chr7  127478198  127479365  Neg3  0  -  127478198  127479365  0,0,255
chr7  127479365  127480532  Pos5  0  +  127479365  127480532  255,0,0
chr7  127480532  127481699  Neg4  0  -  127480532  127481699  0,0,255
```

BED files in ribo-seq



BED files in ribo-seq





Generic file types

- Log files
  - Program and version
  - Command used
  - Job start and end time
  - Results
  - Errors

Thanks!

Any questions?

Extras

Compressing files

- .zip - a compressed archive of file(s).
- .gz - a compressed file(s).
- .tar - an archive of files(s).

Compressing files

```
#Create a tarball
tar -czvf filename.tar.gz /path/to/dir

#Extract a tarball
tar -xzvf filename.tar.gz
```

Where:

- -c = create new archive
- -x = extract files from an archive
- -z = use gzip
- -v = verbose
- -f = use archive file

Asking for help

Online forums:

- Biostars
- stackoverflow

Communities:

- R-ladies
- The Turing Way

md5sum

- You may be asked for the md5sum or MD5 hash to verify file transfers.
- This is a code that acts as a fingerprint for a file - if the file changes, the code will change.

Example on mac:

```
md5sum TLUK2021_talk.Rmd

MD5 (TLUK2021_talk.Rmd) = 01b9630ca402b38d9d50567fa783f92d
```

References

Danecek, Petr, James K Bonfield, Jennifer Liddle, John Marshall, Valeriu Ohan, Martin O Pollard, Andrew Whitwham, et al. 2021. " Twelve years of SAMtools and BCFtools." *GigaScience* 10 (2). https://doi.org/10.1093/gigascience/giab008.

Ensembl. 2021. "BED File Format - Definition and Supported Options." https://www.ensembl.org/info/website/upload/bed.html.

Fan, Jean. 2017. "Cigar Strings for Dummies." https://jef.works/blog/2017/03/28/CIGAR-strings-for-dummies/.

Genome Research Limited. 2021. "Samtools." https://www.htslib.org/.

Griffith, Obi. 2013. "Tutorial:cheat Sheet for One-Based Vs Zero-Based Coordinate Systems." https://www.biostars.org/p/84686/#290319.

NCBI. 2021. "FASTA Format for Nucleotide Sequences." https://www.ncbi.nlm.nih.gov/genbank/fastaformat/.

"NGS Sequencing Technology and File Formats." n.d. https://learn.gencore.bio.nyu.edu/ngs-file-formats/.