Processing NGS data

Epigenomics Data Analysis

Jacques Serizay

Physalia 2025

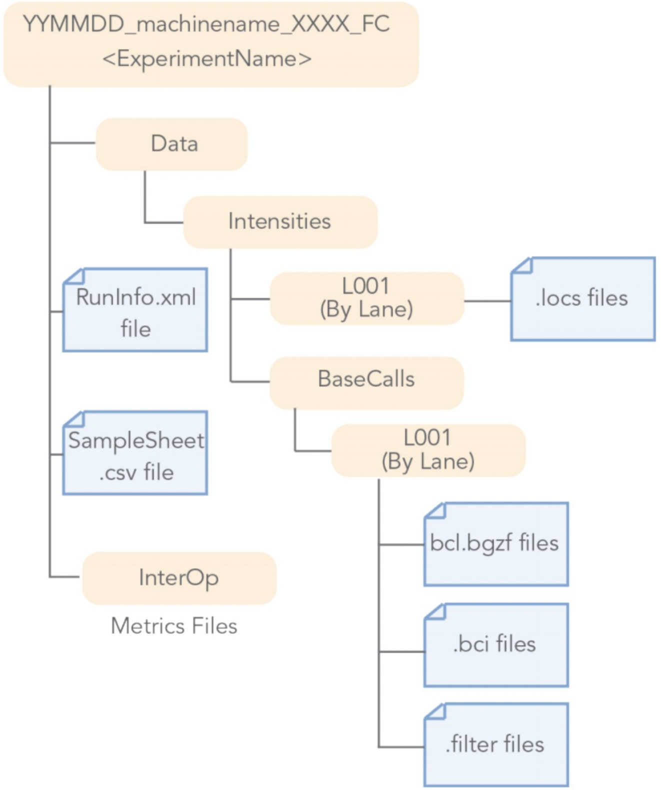
NGS processing workflow

- Get .bcl files
- Create fastq files
- QC: remove/trim low quality reads
- Align fastq to BAM
- Filter duplicates, artifacts, …
- Generate tracks
- Assay-specific downstream analysis

.bcl files

.bcl:

- **Raw data** output of a sequencing run
- **Binary**, non-human-readable file
- Contains the **base calling** and **quality score per cluster, per sequencing lane, per cycle**
- **Huge files**
- **No aggregated sequence per read**



YYMMDD_machinename_XXXX_FC <ExperimentName>

Data

Intensities

RunInfo.xml file

L001 (By Lane)

.locs files

BaseCalls

SampleSheet .csv file

L001 (By Lane)

bcl.bgzf files

InterOp

Metrics Files

.bci files

.filter files

NGS processing workflow

✓ Get .bcl files

● Create fastq files

● QC: remove/trim low quality reads

● Align fastq to BAM

● Filter duplicates, artifacts, …

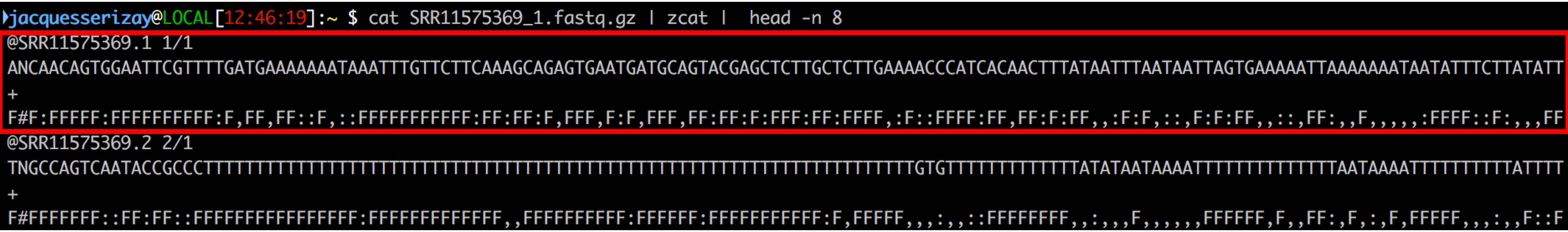
● Generate tracks

● Assay-specific downstream analysis

Fastq files

A fastq file contains reads, each read is composed of 4 lines:

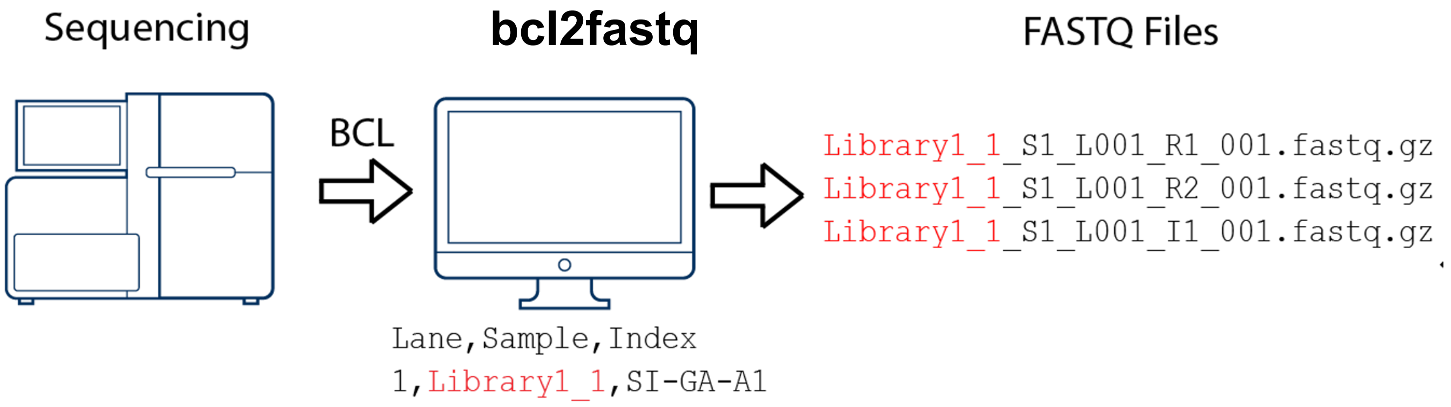
1. A sequence identifier with information about the sequencing run
2. The sequence (the base calls; A, C, T, G and N).
3. A separator, which is simply a plus (+) sign.
4. The base call quality scores, using ASCII characters to represent the numerical quality scores.



```
jacquesserizay@LOCAL[12:46:19]:~ $ cat SRR11575369_1.fastq.gz | zcat |  head -n 8
@SRR11575369.1 1/1
ANCAACAGTGGAATTCGTTTTGATGAAAAAAATAAATTTGTTCTTCAAAGCAGAGTGAATGATGCAGTACGAGCTCTTGCTCTTGAAAACCCATCACAACTTTATAATTTAATAATTAGTGAAAAATTAAAAAAATAATATTTCTTATATT
+
F#F:FFFFF:FFFFFFFFFF:F,FF,FF::F,::FFFFFFFFFFF:FF:FF:F,FFF,F:F,FFF,FF:FF:F:FFF:FF:FFFF,:F::FFFF:FF,FF:F:FF,,:F:F,::,F:F:FF,,::,FF:,,F,,,,,:FFFF::F:,,,FF
@SRR11575369.2 2/1
TNGCCAGTCAATACCGCCCTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTGTGTTTTTTTTTTTTTATATAATAAAATTTTTTTTTTTTTAATAAAATTTTTTTTTATTTT
+
F#FFFFFFF::FF:FF::FFFFFFFFFFFFFFFF:FFFFFFFFFFFFF,,FFFFFFFFFF:FFFFFF:FFFFFFFFFFF:F,FFFFF,,,:,,::FFFFFFFF,,:,,,F,,,,,,FFFFFF,F,,FF:,F,:,F,FFFFF,,,:,,F::F
```

bcl2fastq

```
bcl2fastq --run-folder-dir <bcl_files_folder> --output-dir <fastq_files_folder>
```



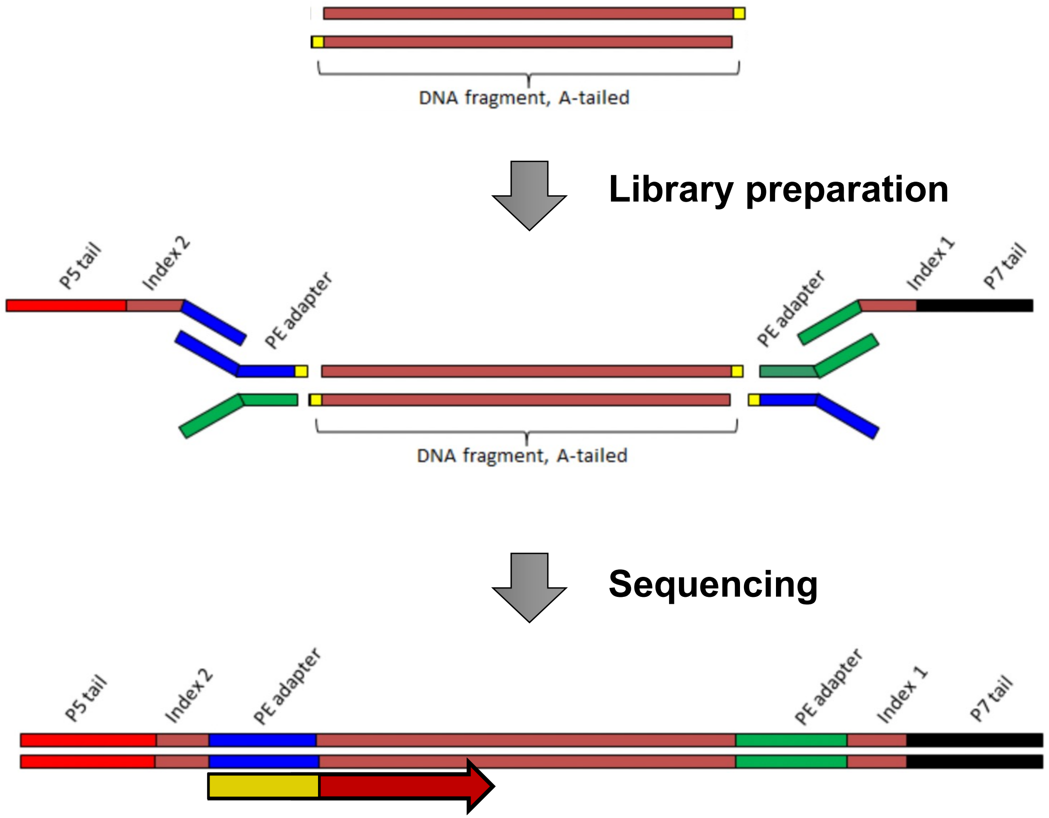
```
Library1_1_S1_L001_R1_001.fastq.gz
Library1_1_S1_L001_R2_001.fastq.gz
Library1_1_S1_L001_I1_001.fastq.gz
```

```
Lane,Sample,Index
1,Library1_1,SI-GA-A1
```

User guide:

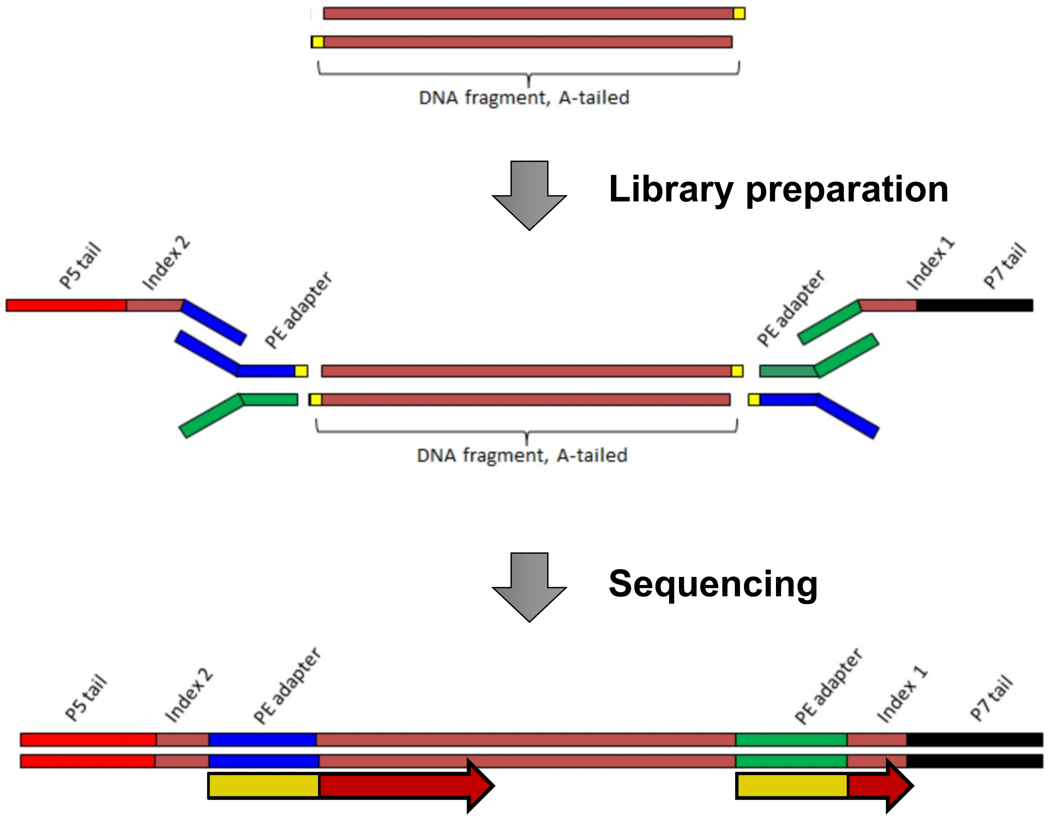
https://support.illumina.com/content/dam/illumina-support/documents/documentation/software_documentation/bcl2fastq/bcl2fastq_letterbooklet_15038058brpmi.pdf

Why so many fastq files?



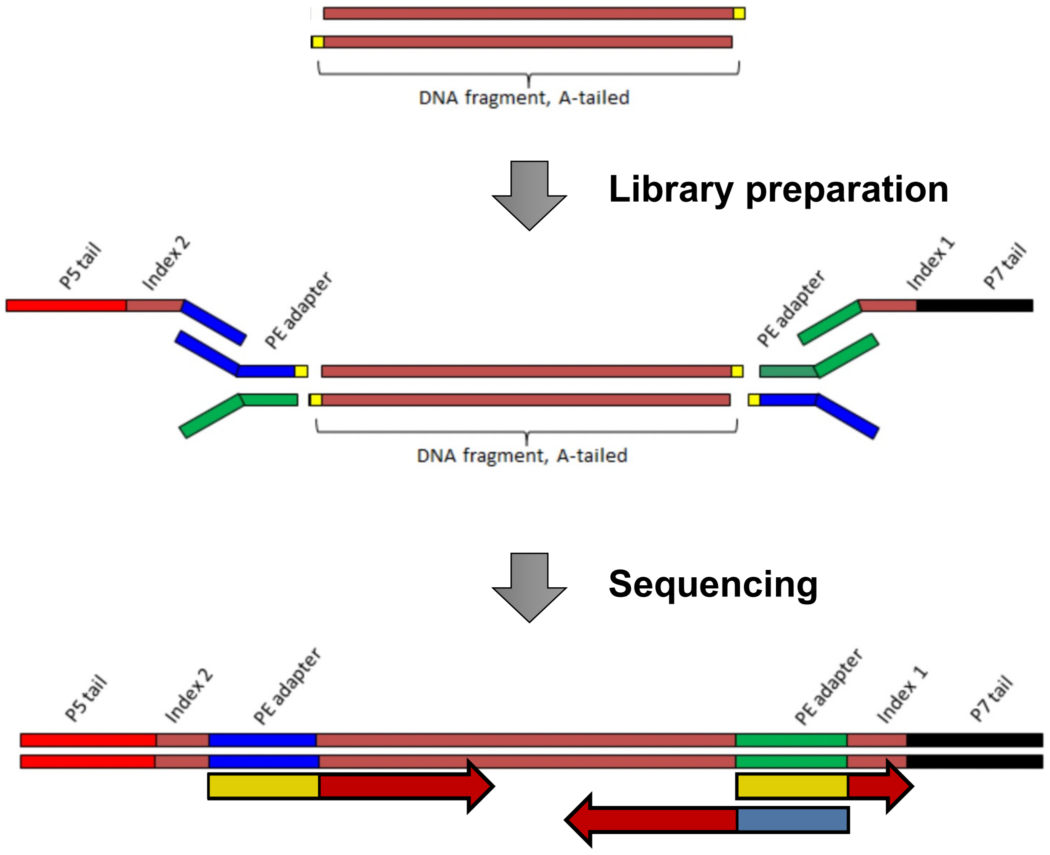
“R1” sequence

Why so many fastq files?



“I1” sequence

Why so many fastq files?

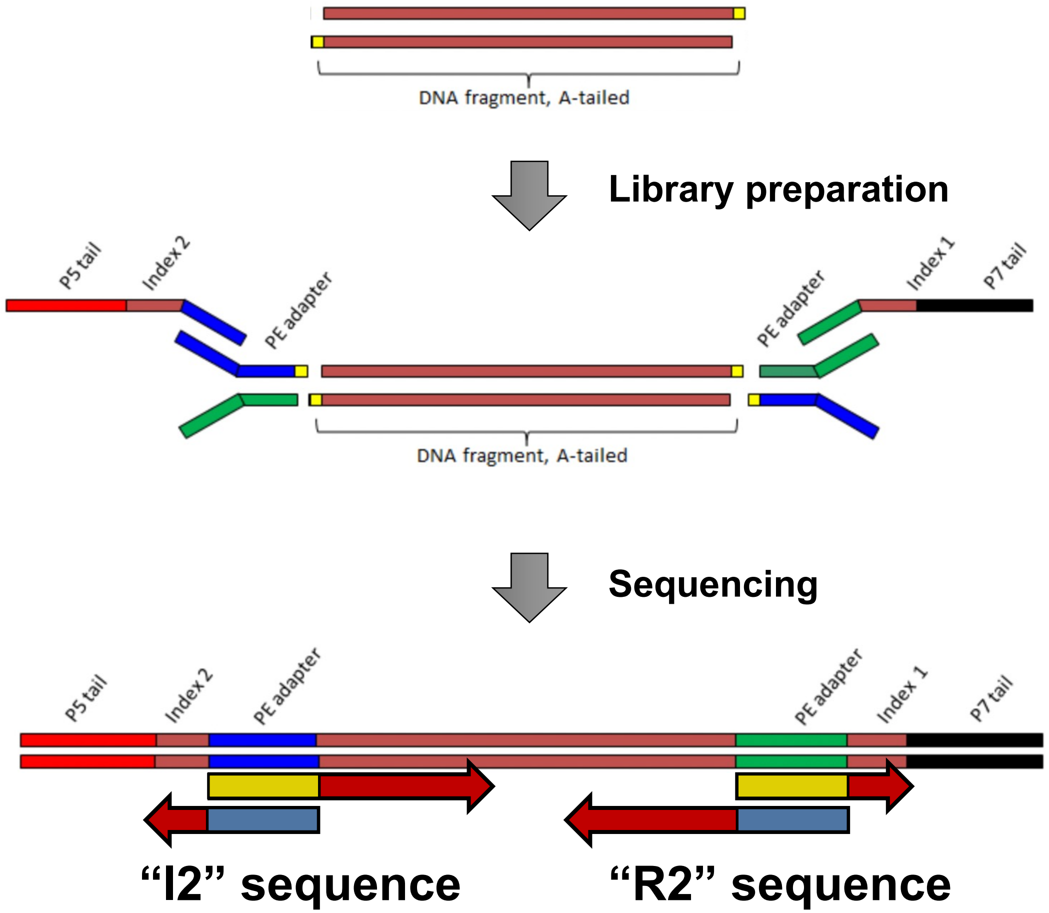


Library preparation

Sequencing

“R2” sequence

Why so many fastq files?



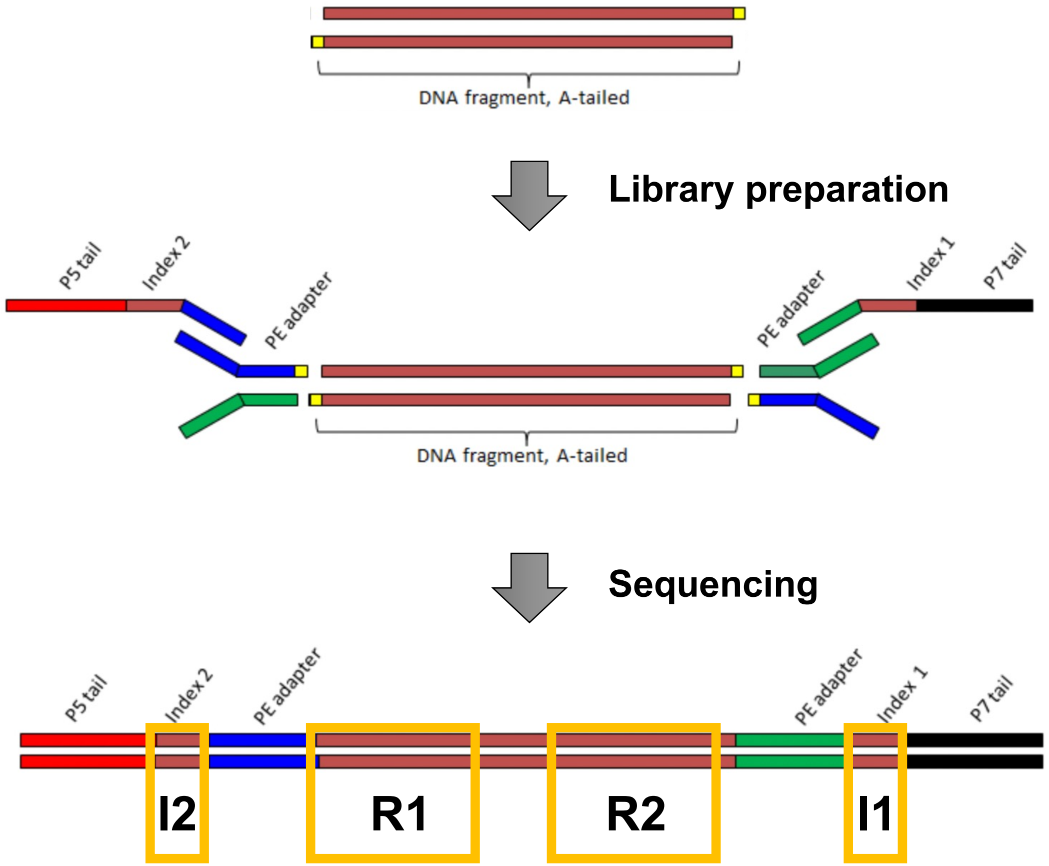
Library preparation

Sequencing

“I2” sequence

“R2” sequence

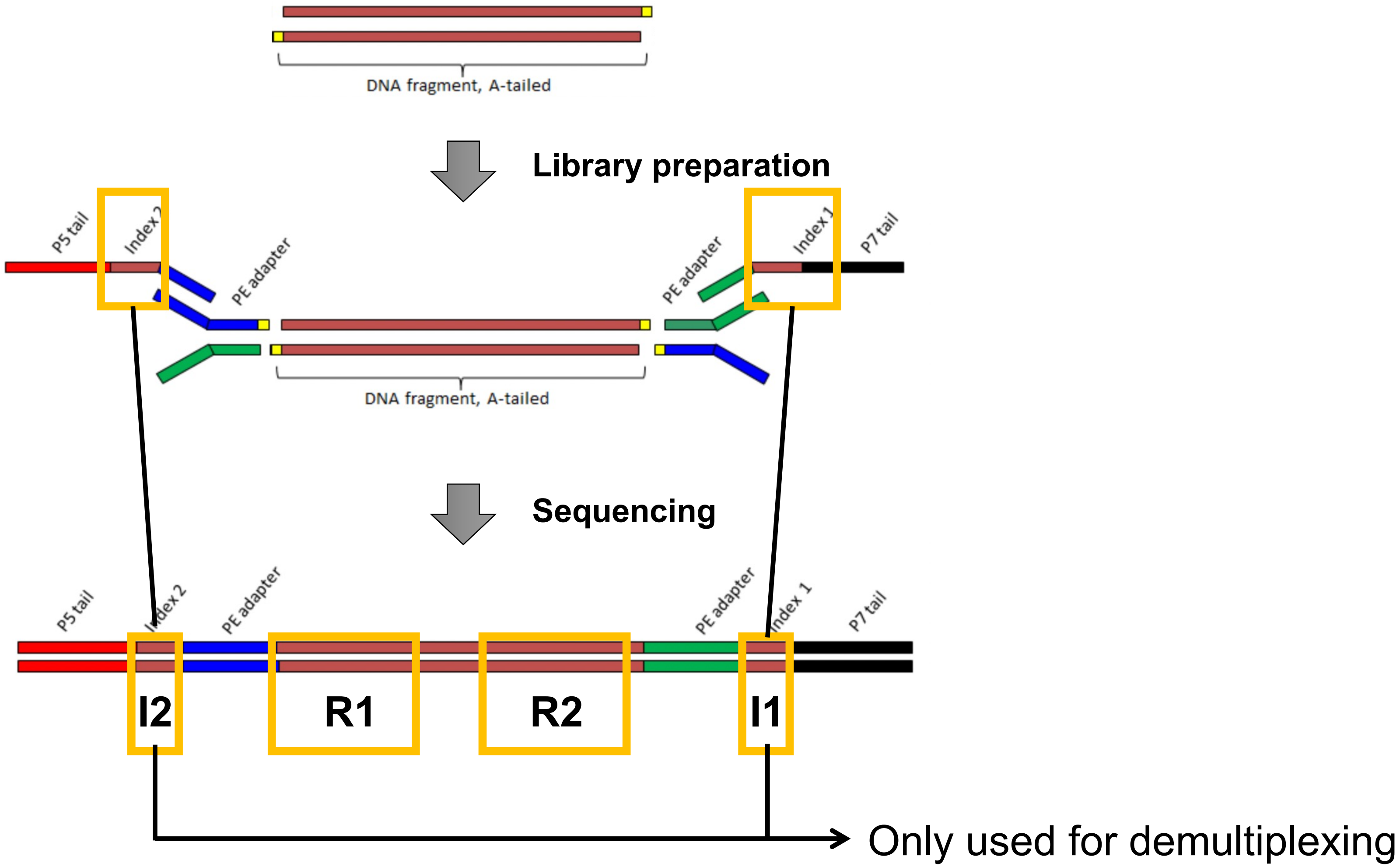
Why so many fastq files?



Library preparation

Sequencing

Why so many fastq files?



NGS processing workflow

✓ Get .bcl files

✓ Create fastq files

Or **`bcl2fastq`**

● QC: remove/trim low quality reads

● Align fastq to BAM

● Filter duplicates, artifacts, …

● Generate tracks

● Assay-specific downstream analysis

NGS processing workflow

✓ Get .bcl files

✓ Create fastq files

Or **`bcl2fastq`**

CHECK YOUR DATA

● QC: remove/trim low quality reads

● Align fastq to BAM

● Filter duplicates, artifacts, …

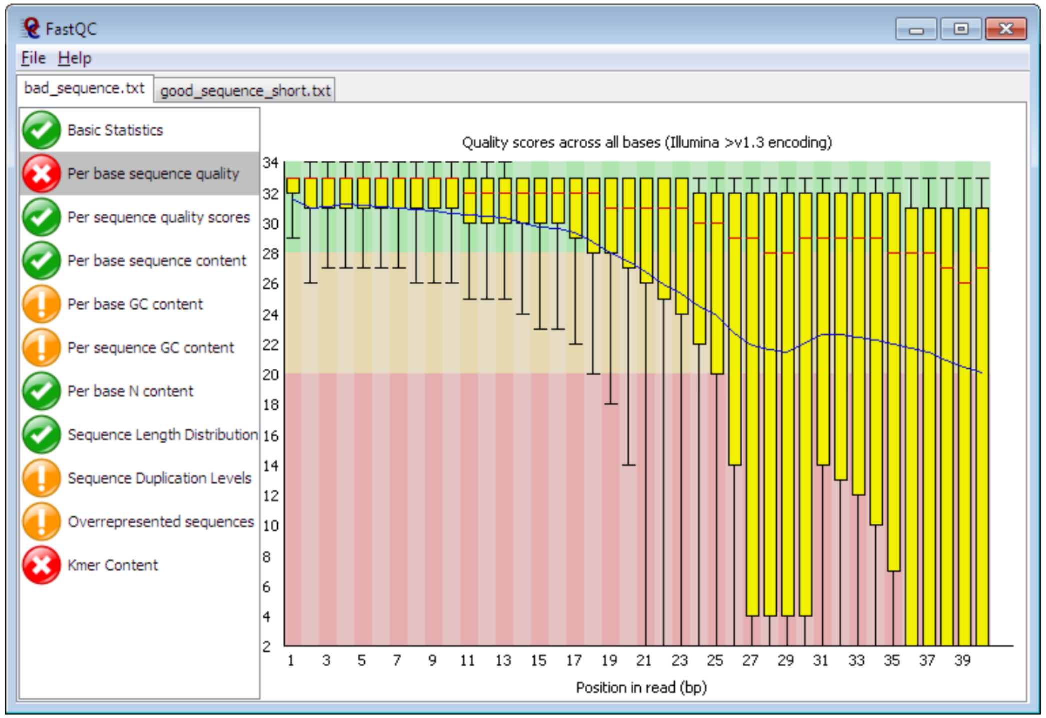
● Generate tracks

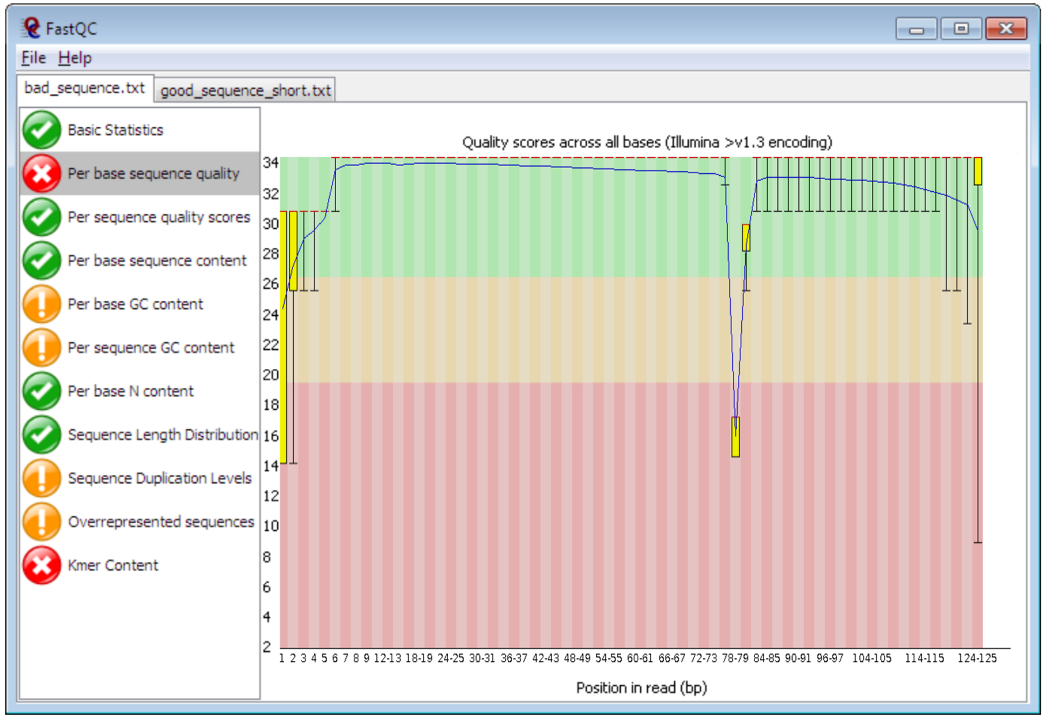
● Assay-specific downstream analysis

FastQC

FastQC

FastQC is a program designed to spot potential problems in high througput sequencing datasets. It runs a set of analyses on one or more raw sequence files in fastq or bam format and produces a report which summarises the results.



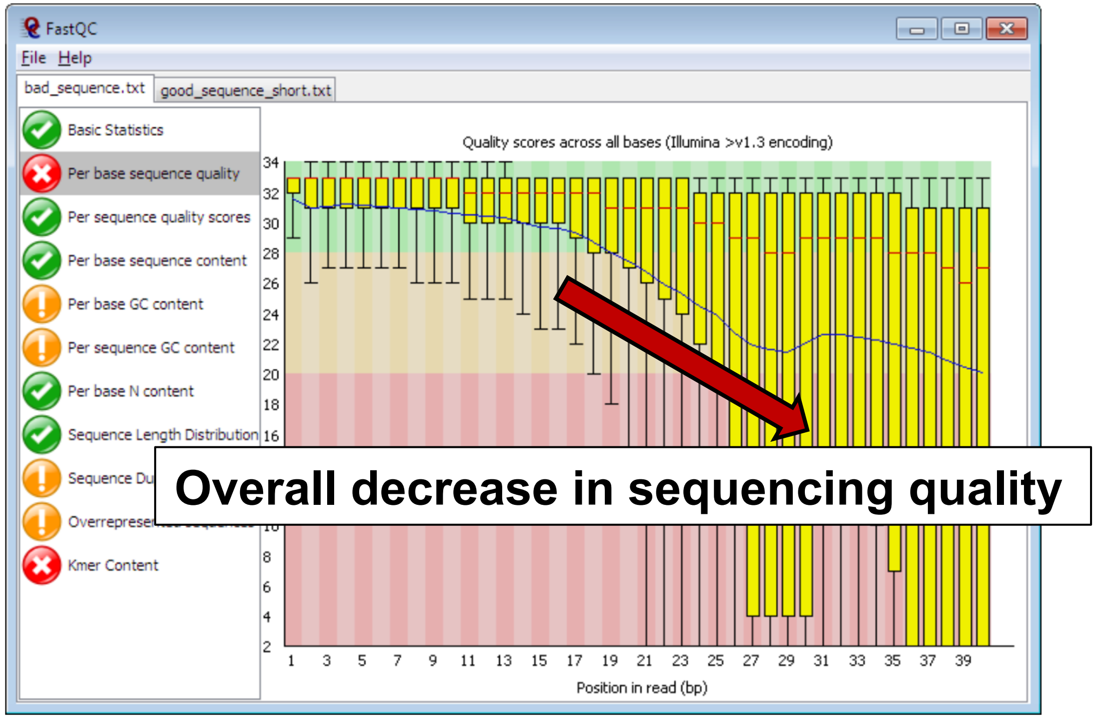


FastQC will highlight any areas where this library looks unusual and where you should take a closer look. The program is not tied to any specific type of sequencing technique and can be used to look at libraries coming from a large number of different experiment types (Genomic Sequencing, ChIP-Seq, RNA-Seq, BS-Seq etc etc).

FastQC

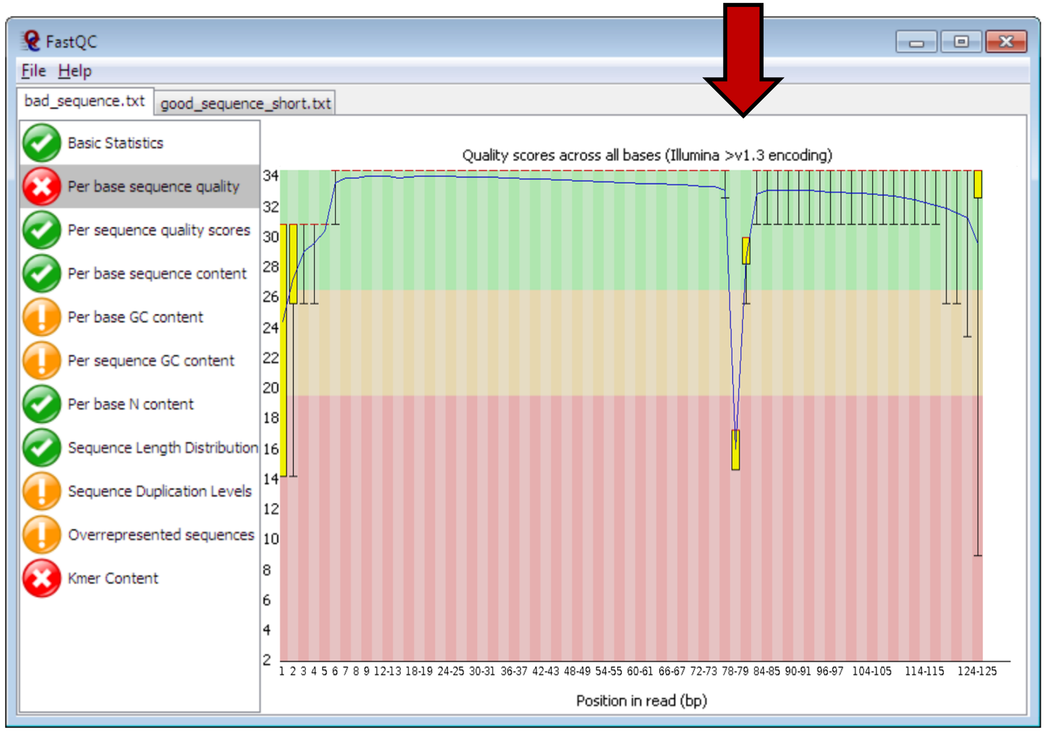
FastQC

FastQC is a program designed to spot potential problems in high througput sequencing datasets. It runs a set of analyses on one or more raw sequence files in fastq or bam format and produces a report which summarises the results.



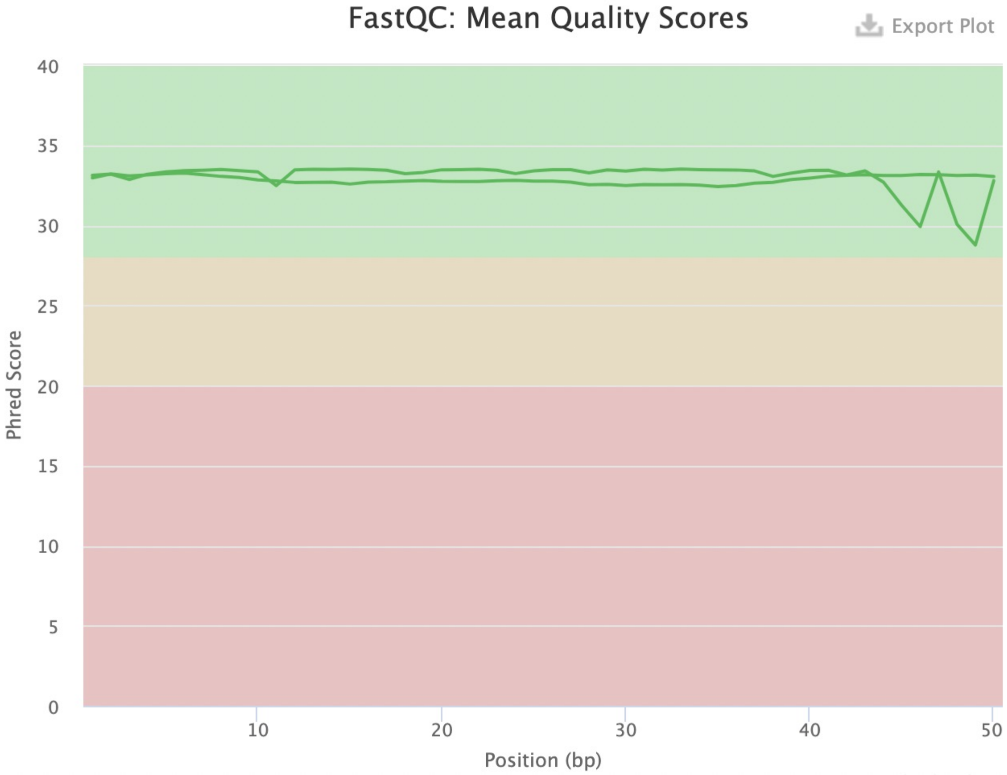
Overall decrease in sequencing quality

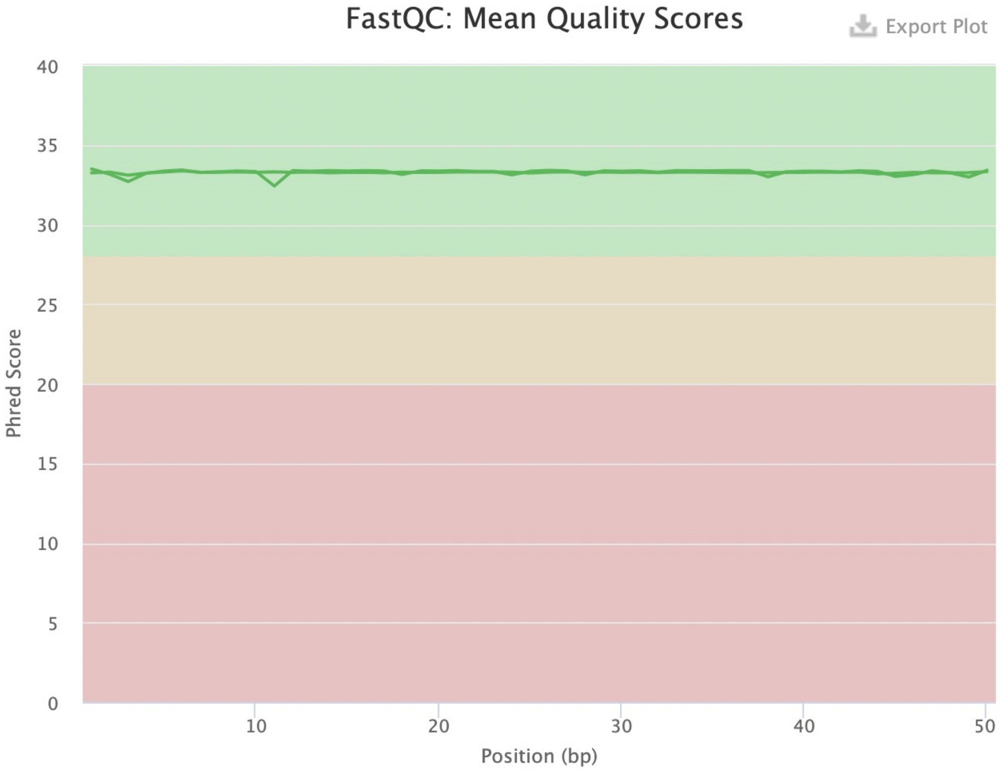
Local artefact



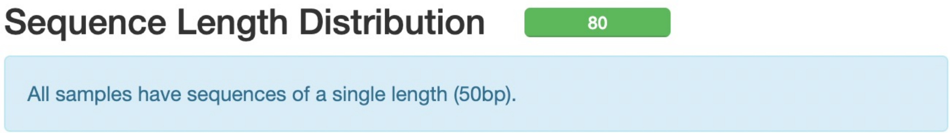
FastQC will highlight any areas where this library looks unusual and where you should take a closer look. The program is not tied to any specific type of sequencing technique and can be used to look at libraries coming from a large number of different experiment types (Genomic Sequencing, ChIP-Seq, RNA-Seq, BS-Seq etc etc).

Cutadapt: trim away adapter sequences and low-quality ends





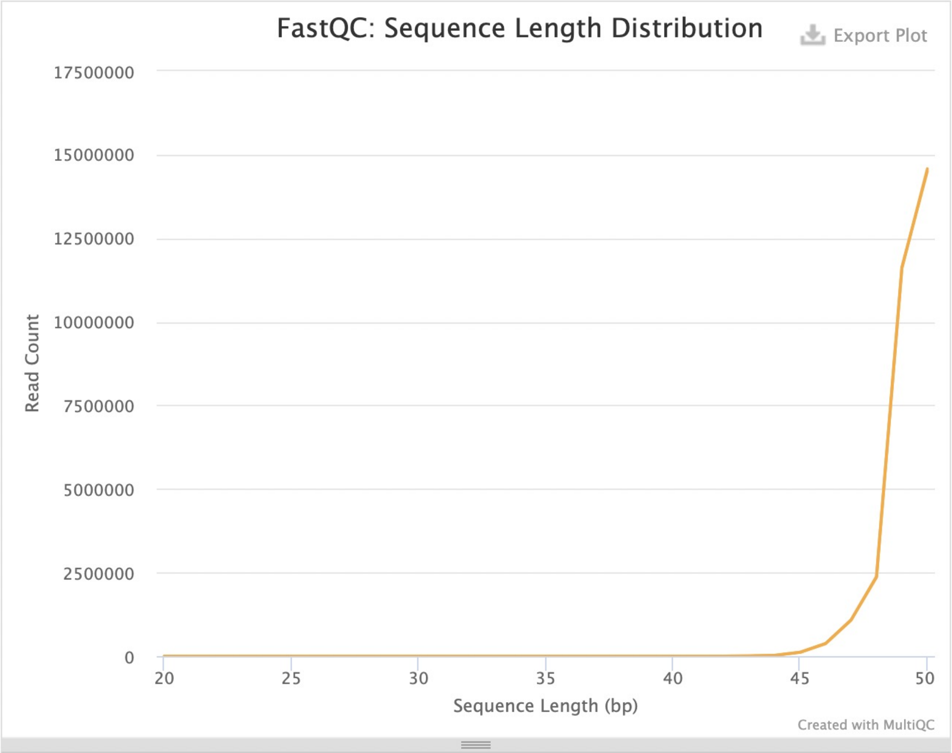
Cutadapt: trim away adapter sequences and low-quality ends



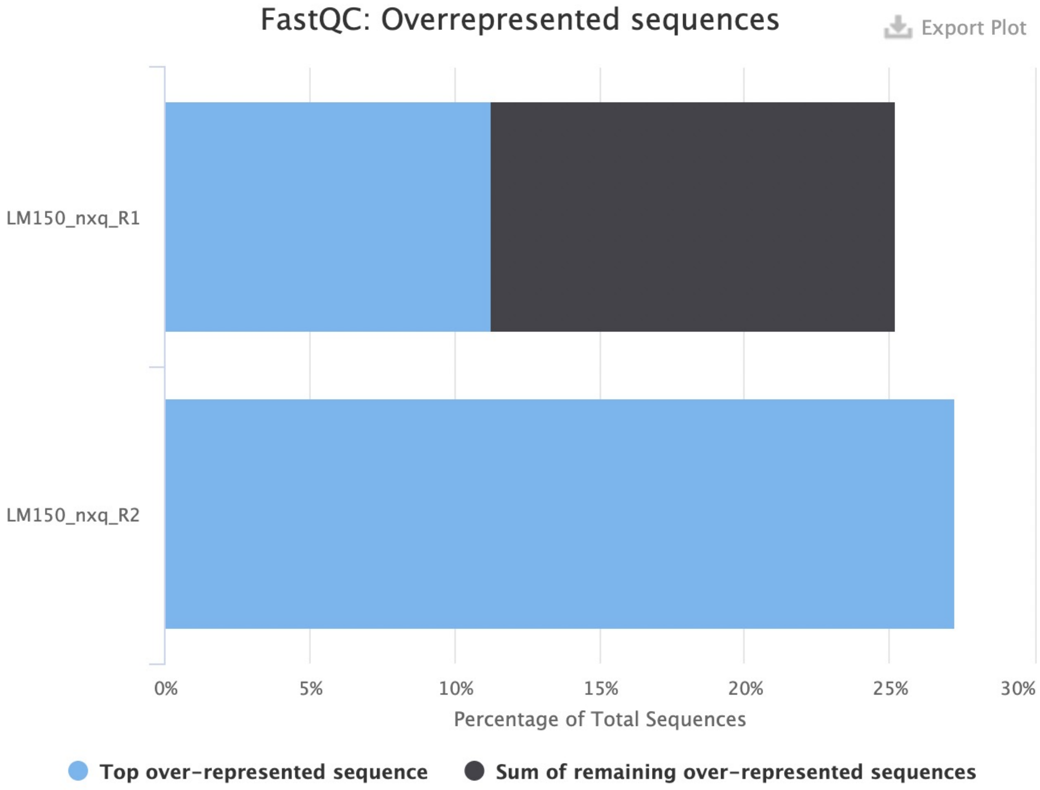
Sequence Length Distribution

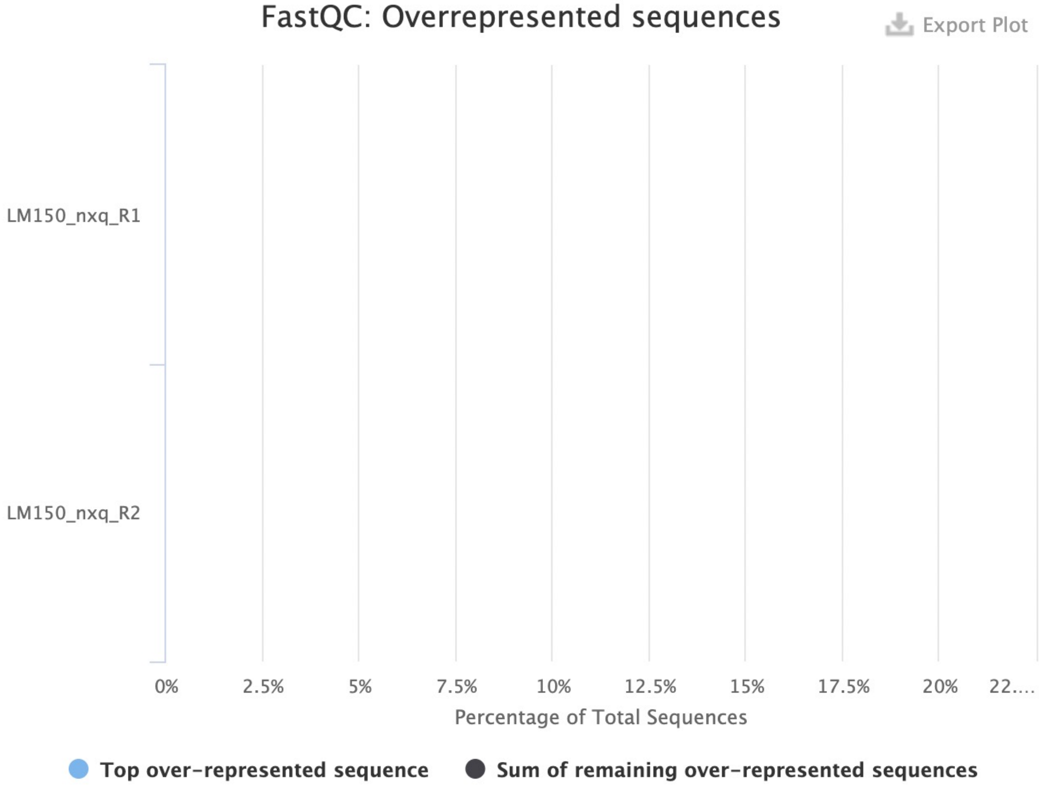
80

All samples have sequences of a single length (50bp).

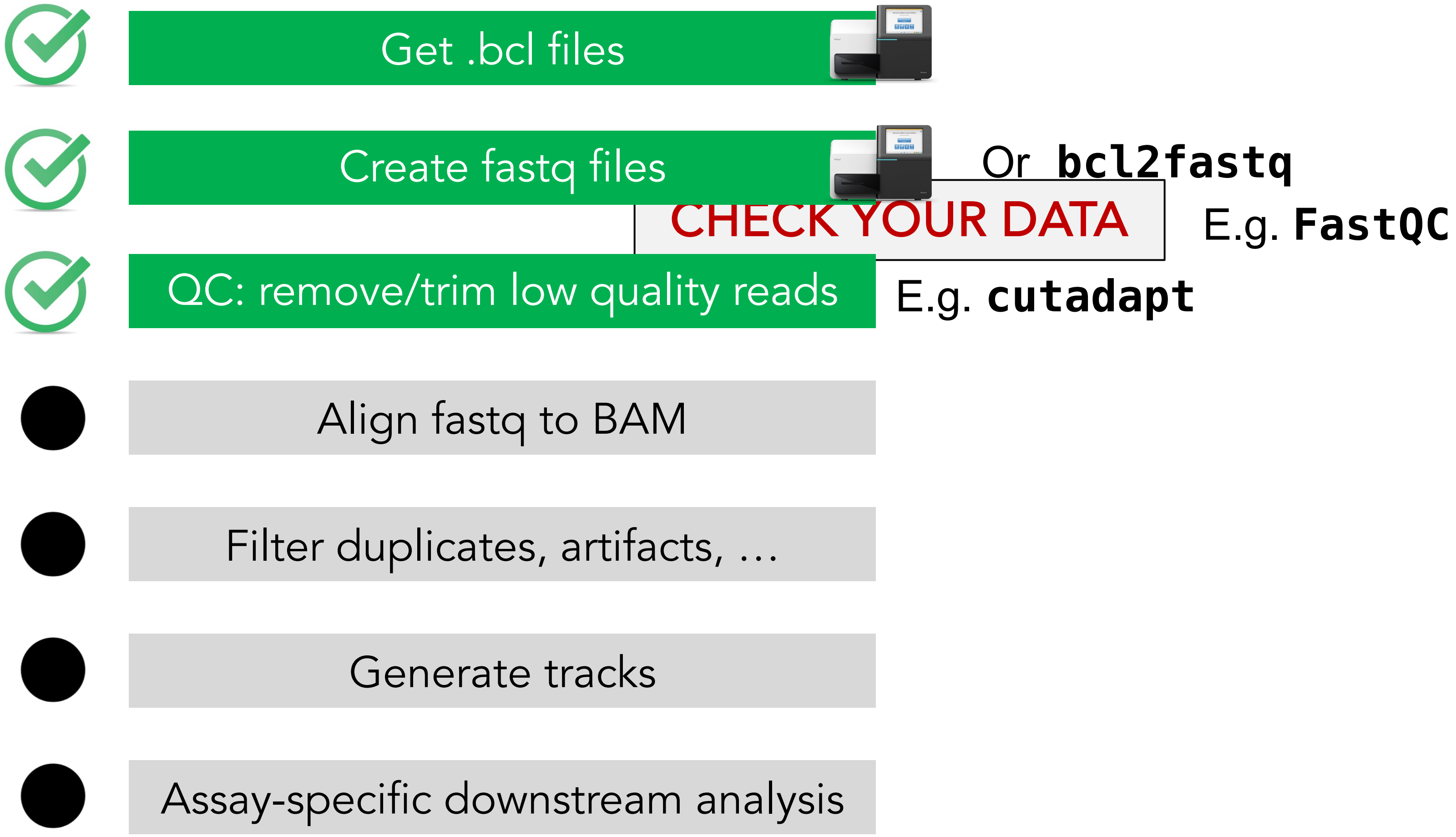


Cutadapt: trim away adapter sequences and low-quality ends





NGS processing workflow

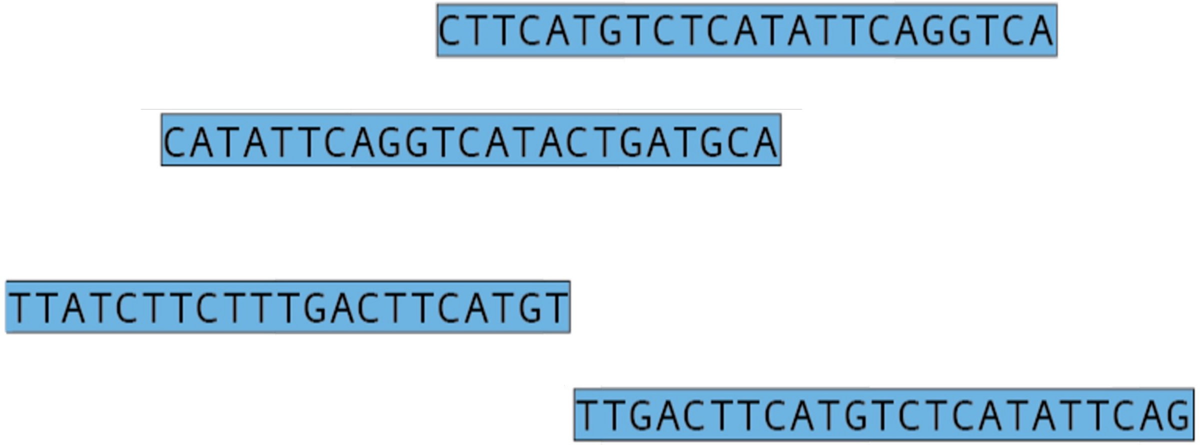


- ✓ Get .bcl files
- ✓ Create fastq files — Or **`bcl2fastq`**

CHECK YOUR DATA — E.g. **`FastQC`**

- ✓ QC: remove/trim low quality reads — E.g. **`cutadapt`**
- ● Align fastq to BAM
- ● Filter duplicates, artifacts, …
- ● Generate tracks
- ● Assay-specific downstream analysis

Mapping sequencing reads to a reference



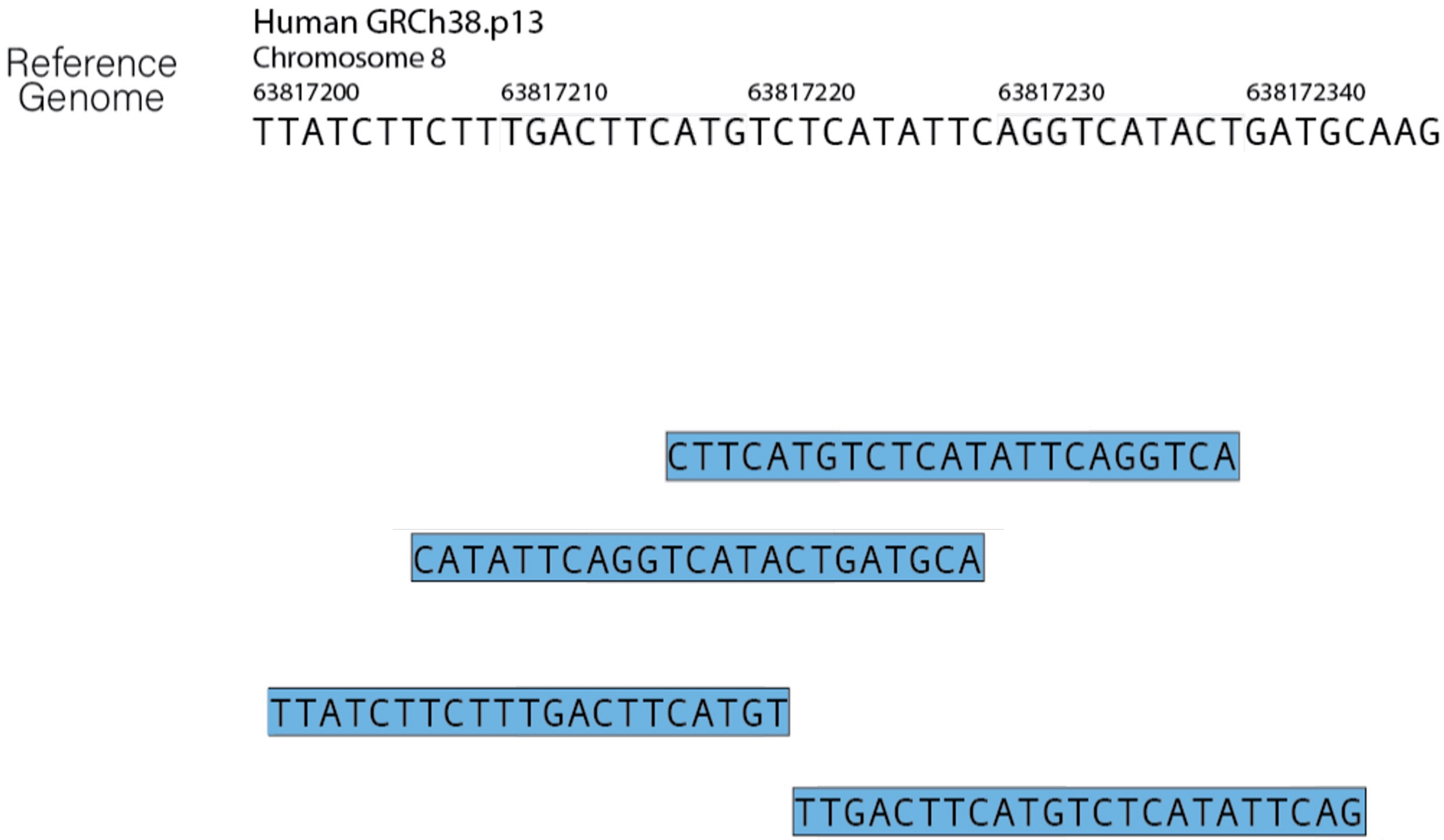
CTTCATGTCTCATATTCAGGTCA

CATATTCAGGTCATACTGATGCA

TTATCTTCTTTGACTTCATGT

TTGACTTCATGTCTCATATTCAG

Mapping sequencing reads to a reference



Reference Genome

Human GRCh38.p13
Chromosome 8

63817200 63817210 63817220 63817230 638172340

TTATCTTCTTTGACTTCATGTCTCATATTCAGGTCATACTGATGCAAG

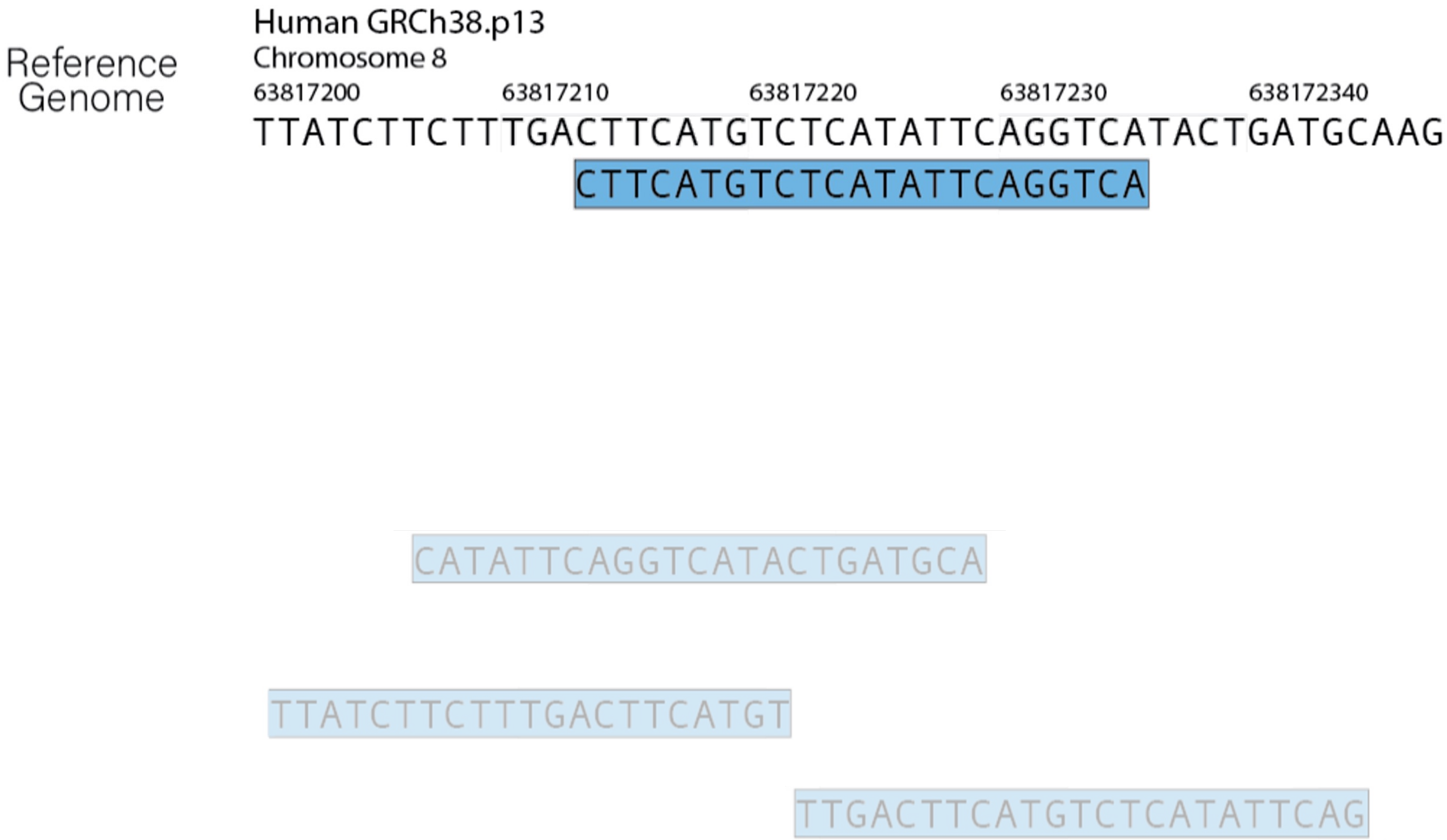
CTTCATGTCTCATATTCAGGTCA

CATATTCAGGTCATACTGATGCA

TTATCTTCTTTGACTTCATGT

TTGACTTCATGTCTCATATTCAG

Mapping sequencing reads to a reference



Reference Genome

Human GRCh38.p13

Chromosome 8

63817200 63817210 63817220 63817230 638172340

TTATCTTCTTTGACTTCATGTCTCATATTCAGGTCATACTGATGCAAG

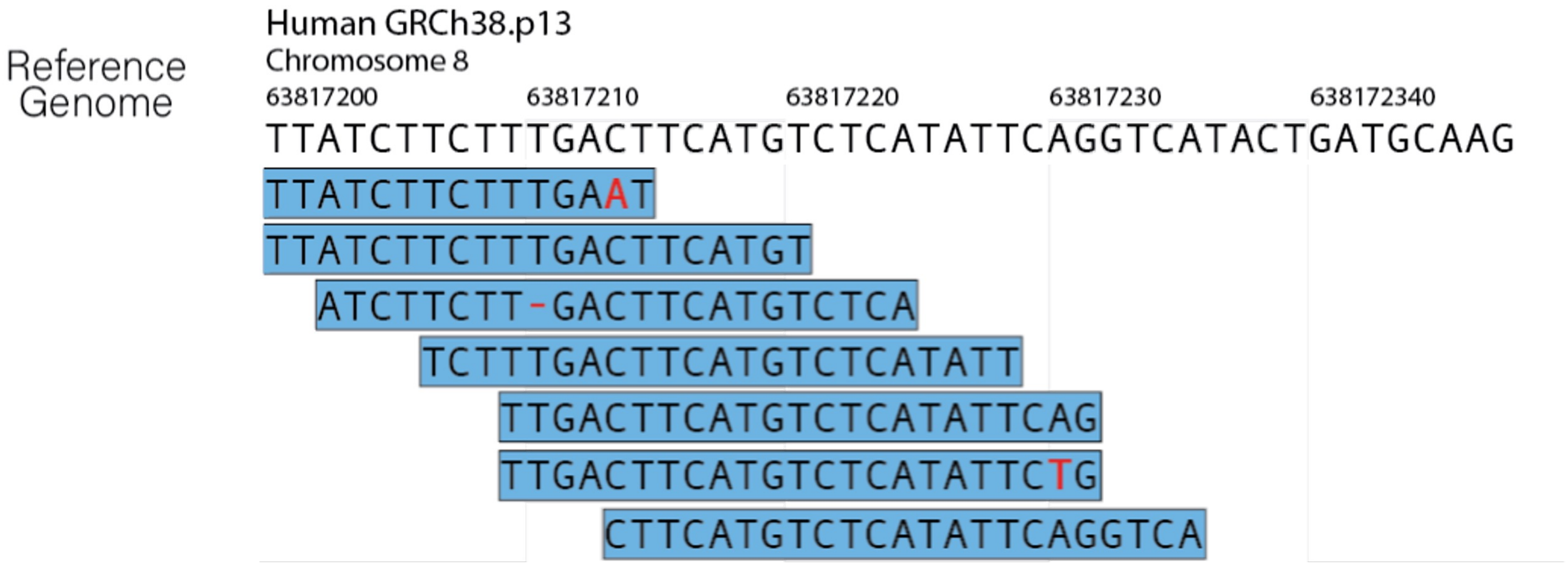
CTTCATGTCTCATATTCAGGTCA

CATATTCAGGTCATACTGATGCA

TTATCTTCTTTGACTTCATGT

TTGACTTCATGTCTCATATTCAG

Mapping sequencing reads to a reference



Reference Genome

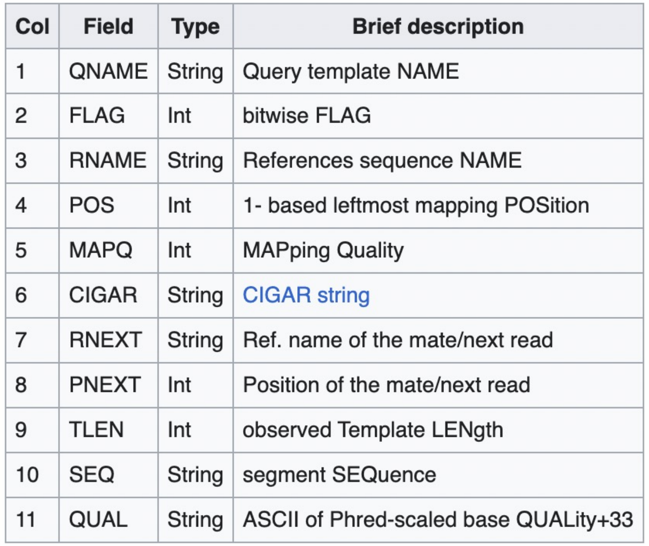
Human GRCh38.p13
Chromosome 8

```
63817200  63817210  63817220  63817230  638172340
TTATCTTCTTTGACTTCATGTCTCATATTCAGGTCATACTGATGCAAG
TTATCTTCTTTGAAT
TTATCTTCTTTGACTTCATGT
  ATCTTCTT-GACTTCATGTCTCA
      TCTTTGACTTCATGTCTCATATT
         TTGACTTCATGTCTCATATTCAG
         TTGACTTCATGTCTCATATTCTG
             CTTCATGTCTCATATTCAGGTCA
```

SAM file format

Sequence Alignment Map (SAM) is a human-readable, rectangular, text-based format for storing biological sequences aligned to a reference sequence.

Each entry (line) describes where a read is mapped on the reference and how it is mapped

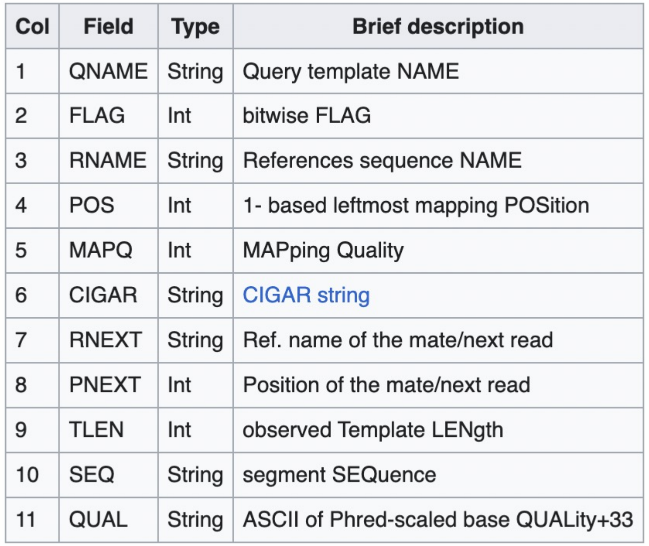


Col	Field	Type	Brief description
1	QNAME	String	Query template NAME
2	FLAG	Int	bitwise FLAG
3	RNAME	String	References sequence NAME
4	POS	Int	1- based leftmost mapping POSition
5	MAPQ	Int	MAPping Quality
6	CIGAR	String	CIGAR string
7	RNEXT	String	Ref. name of the mate/next read
8	PNEXT	Int	Position of the mate/next read
9	TLEN	Int	observed Template LENgth
10	SEQ	String	segment SEQuence
11	QUAL	String	ASCII of Phred-scaled base QUALity+33

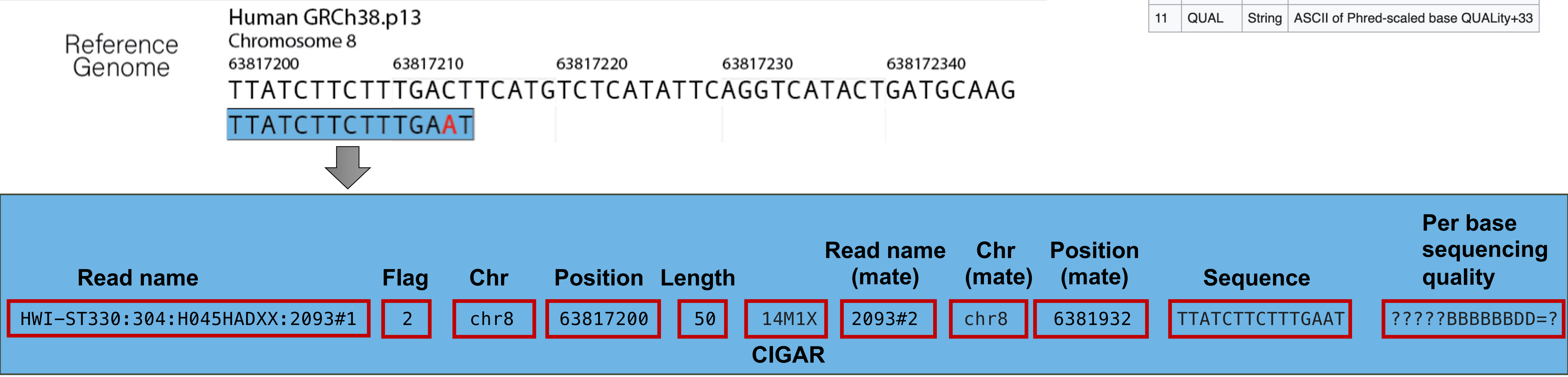
SAM file format

Sequence Alignment Map (SAM) is a human-readable, rectangular, text-based format for storing biological sequences aligned to a reference sequence.

Each entry (line) describes where a read is mapped on the reference and how it is mapped



Col	Field	Type	Brief description
1	QNAME	String	Query template NAME
2	FLAG	Int	bitwise FLAG
3	RNAME	String	References sequence NAME
4	POS	Int	1- based leftmost mapping POSition
5	MAPQ	Int	MAPping Quality
6	CIGAR	String	CIGAR string
7	RNEXT	String	Ref. name of the mate/next read
8	PNEXT	Int	Position of the mate/next read
9	TLEN	Int	observed Template LENgth
10	SEQ	String	segment SEQuence
11	QUAL	String	ASCII of Phred-scaled base QUALity+33



BAM file format

BAM files are **binarized** SAM files, allowing great compression of the alignment results.

However, bam files are not directly human-readable.

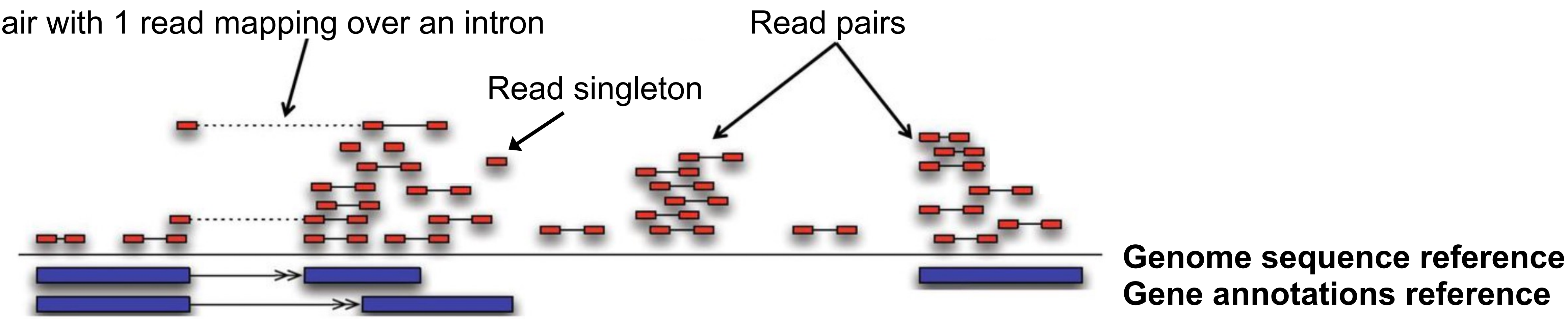
```
samtools view --bam ….sam > ….bam
```

Mapping tools

There are a plethora of alignment tools.

Each one requires the genome reference to be indexed first.

Some mappers can be ”**splice-aware**”, allowing reads to be mapped over annotated introns.



Pair with 1 read mapping over an intron

Read singleton

Read pairs

Genome sequence reference
Gene annotations reference

NGS processing workflow

✓ Get .bcl files

✓ Create fastq files

Or **`bcl2fastq`**

✓ QC: remove/trim low quality reads — E.g. **`cutadapt`**

✓ Align fastq to BAM — E.g. **`bowtie2`**

● Filter duplicates, artifacts, …

● Generate tracks

● Assay-specific downstream analysis

NGS processing workflow

- ✓ Get .bcl files

- ✓ Create fastq files

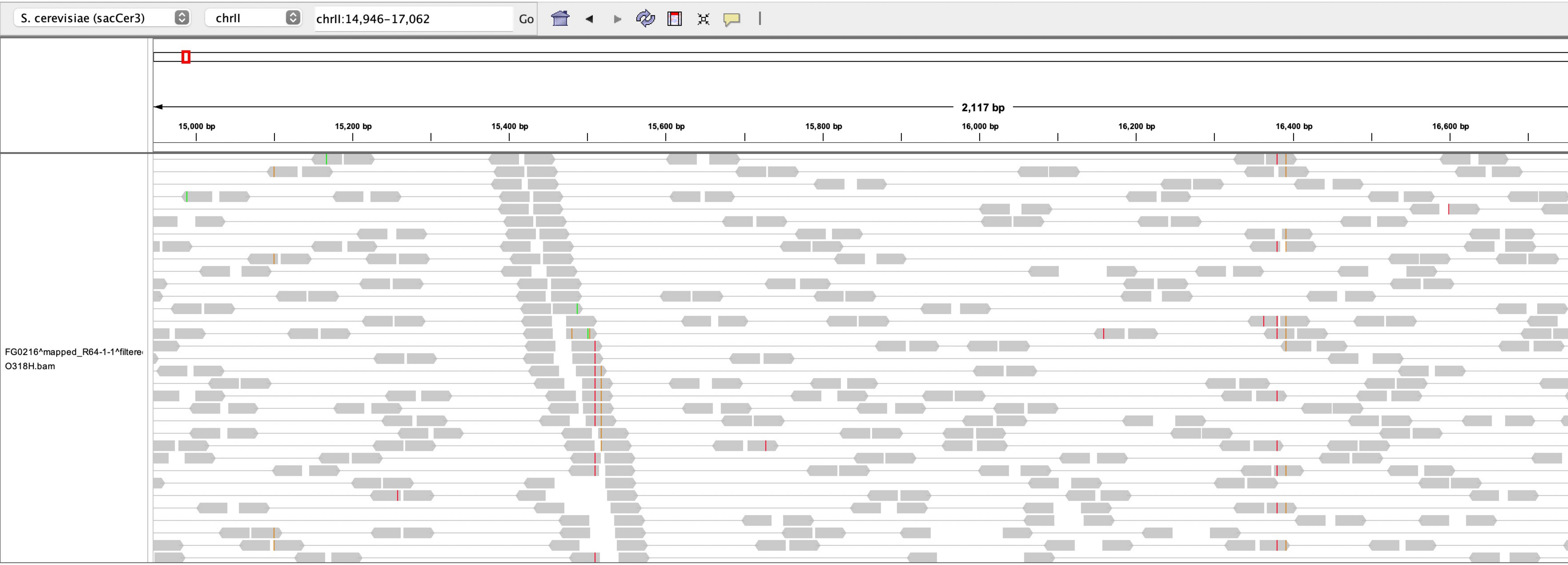
Or **`bcl2fastq`**

- ✓ QC: remove/trim low quality reads — E.g. **`cutadapt`**
- ✓ Align fastq to BAM — E.g. **`bowtie2`**

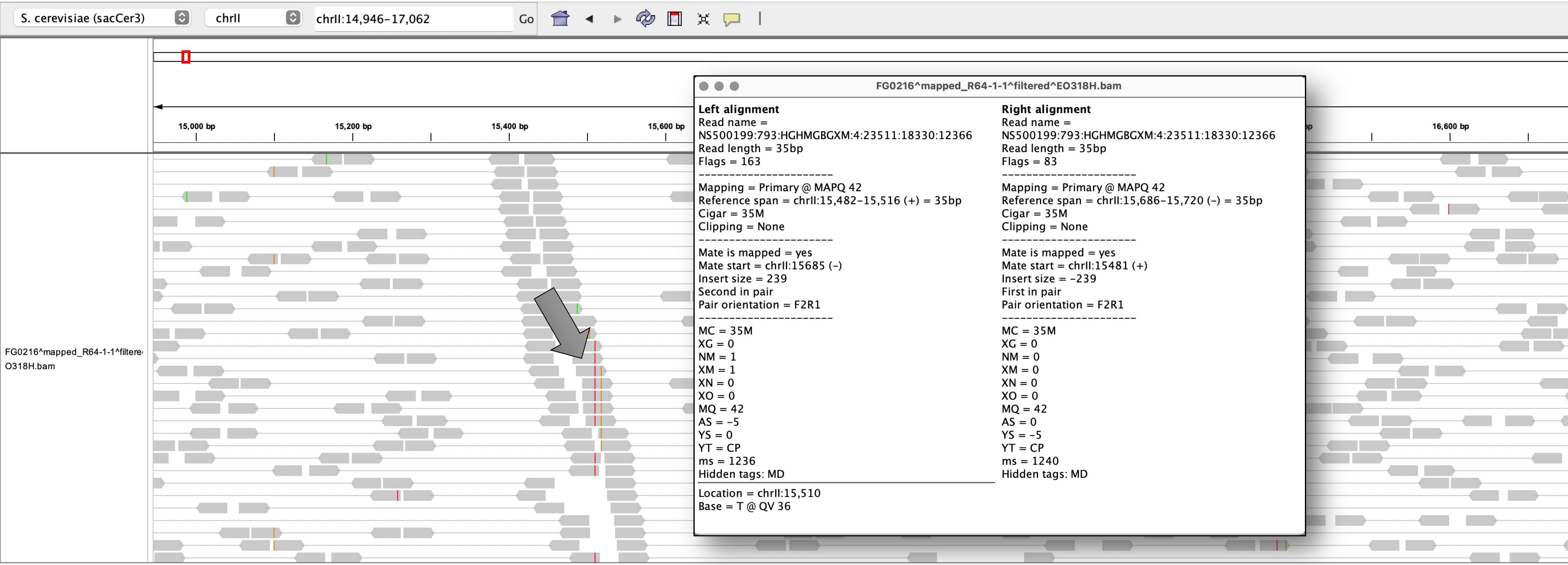
CHECK YOUR DATA

- ● Filter duplicates, artifacts, …
- ● Generate tracks
- ● Assay-specific downstream analysis

IGV: Integrative Genome Browser



IGV: Integrative Genome Browser



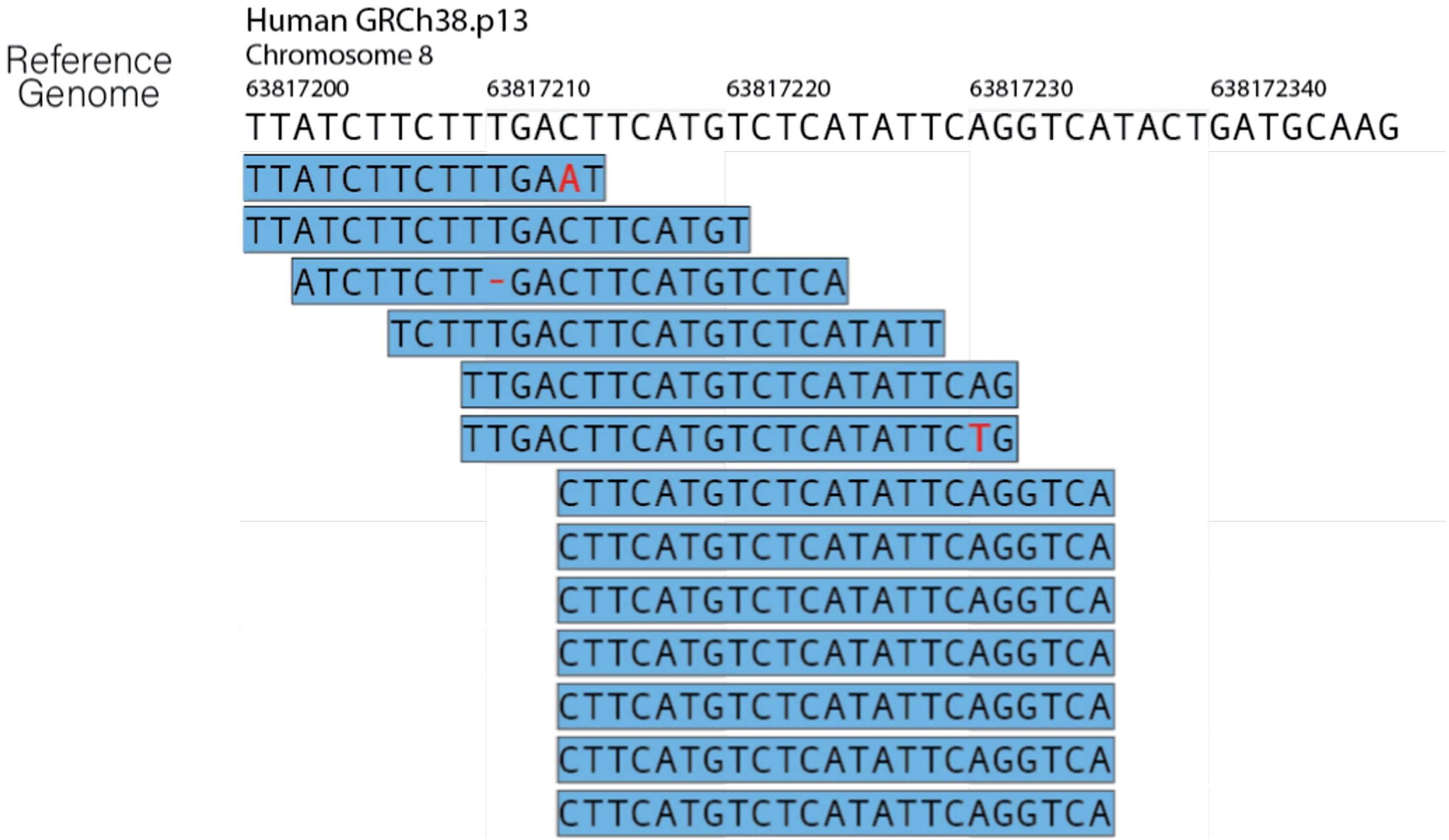
FG0216^mapped_R64-1-1^filtered^EO318H.bam

```
Left alignment
Read name =
NS500199:793:HGHMGBGXM:4:23511:18330:12366
Read length = 35bp
Flags = 163
----------------------
Mapping = Primary @ MAPQ 42
Reference span = chrII:15,482-15,516 (+) = 35bp
Cigar = 35M
Clipping = None
----------------------
Mate is mapped = yes
Mate start = chrII:15685 (-)
Insert size = 239
Second in pair
Pair orientation = F2R1
----------------------
MC = 35M
XG = 0
NM = 1
XM = 1
XN = 0
XO = 0
MQ = 42
AS = -5
YS = 0
YT = CP
ms = 1236
Hidden tags: MD
Location = chrII:15,510
Base = T @ QV 36
```

```
Right alignment
Read name =
NS500199:793:HGHMGBGXM:4:23511:18330:12366
Read length = 35bp
Flags = 83
----------------------
Mapping = Primary @ MAPQ 42
Reference span = chrII:15,686-15,720 (-) = 35bp
Cigar = 35M
Clipping = None
----------------------
Mate is mapped = yes
Mate start = chrII:15481 (+)
Insert size = -239
First in pair
Pair orientation = F2R1
----------------------
MC = 35M
XG = 0
NM = 0
XM = 0
XN = 0
XO = 0
MQ = 42
AS = 0
YS = -5
YT = CP
ms = 1240
Hidden tags: MD
```

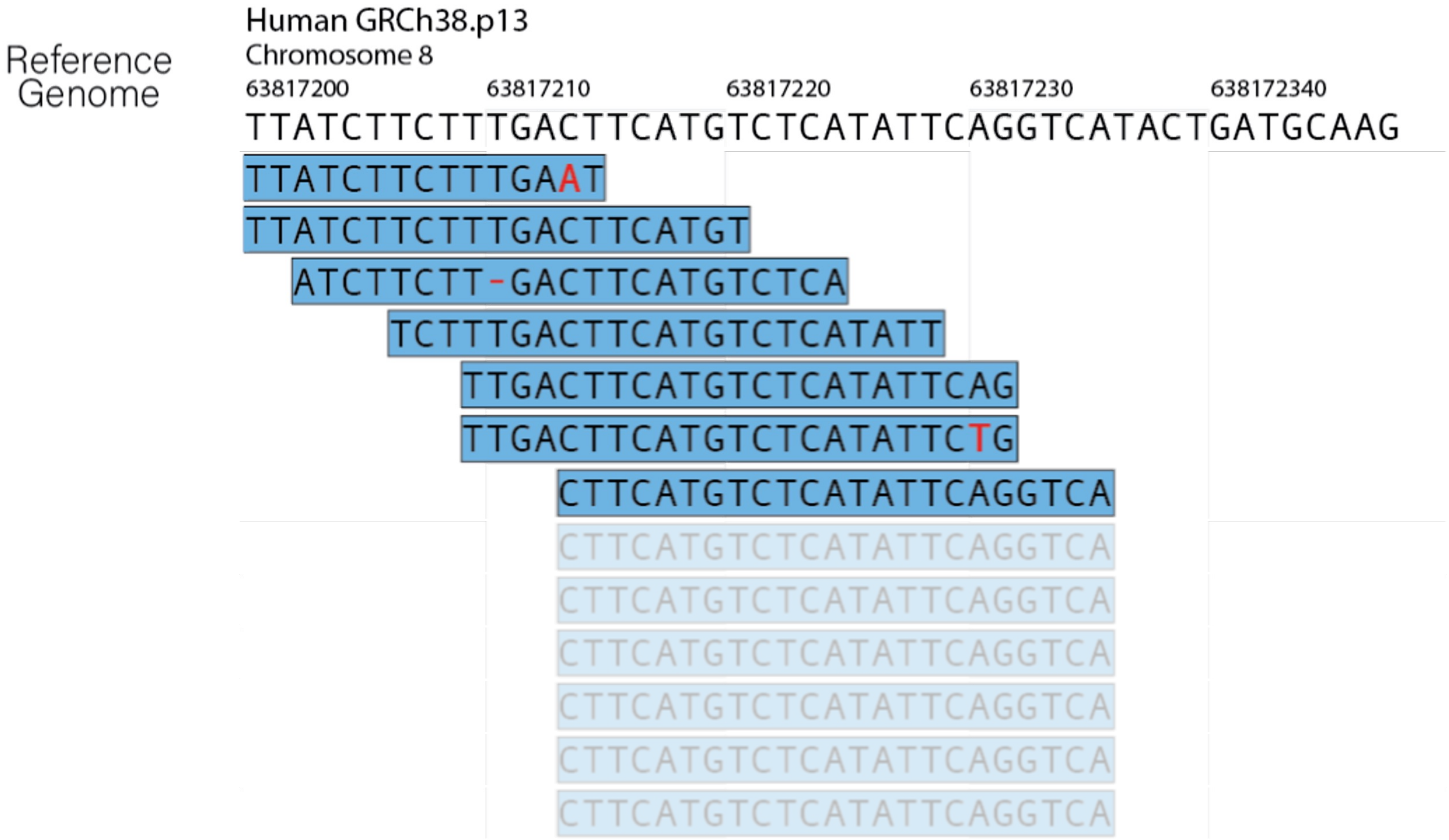
Filtering duplicates

Multiple reads (fragments) with same mapping position (start & end) can be viewed as PCR duplicates.



Filtering duplicates

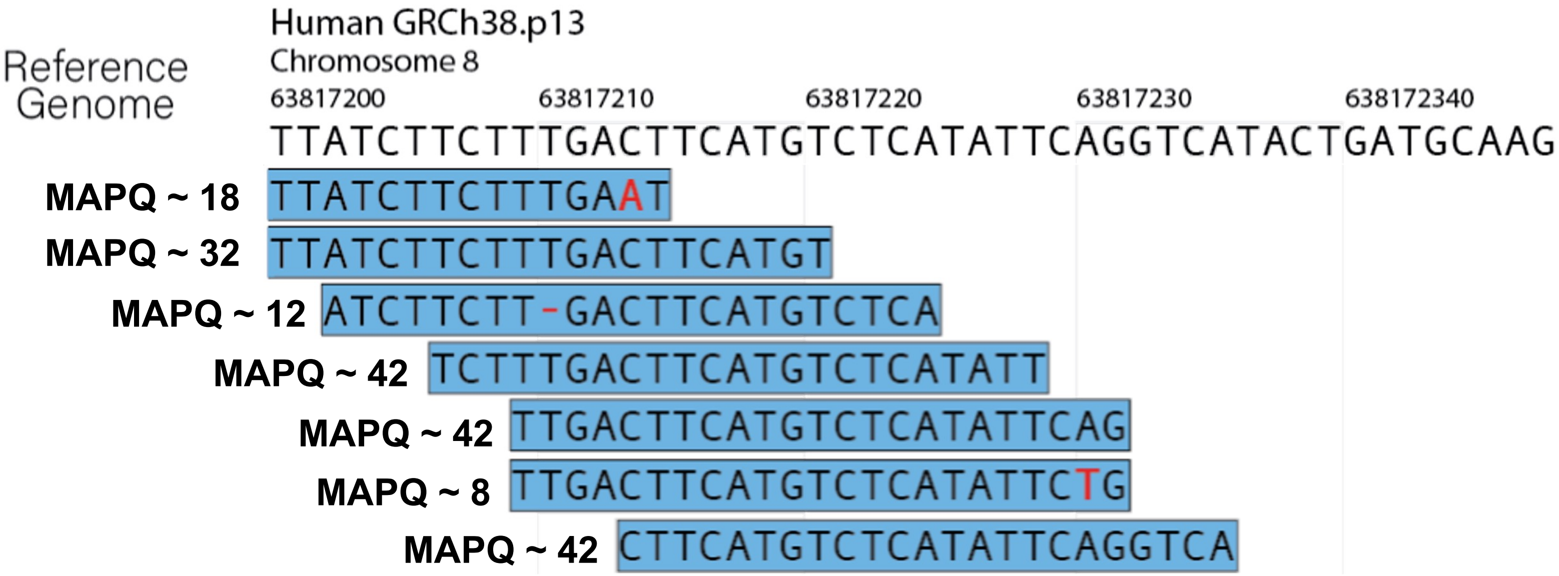
Multiple reads (fragments) with same mapping position (start & end) can be viewed as PCR duplicates.



Filtering out low-quality mapping reads

Mapping Quality Scores (**MAPQ**) quantify the probability that a read is misplaced.

$$\boxed{MAPQ = -10 * log_{10}\big(P(read\ is\ wrongly\ mapped)\big)}$$



Reference Genome

Human GRCh38.p13

Chromosome 8

63817200 63817210 63817220 63817230 638172340

TTATCTTCTTTGACTTCATGTCTCATATTCAGGTCATACTGATGCAAG

MAPQ ~ 18 TTATCTTCTTTGAAT

MAPQ ~ 32 TTATCTTCTTTGACTTCATGT

MAPQ ~ 12 ATCTTCTT-GACTTCATGTCTCA

MAPQ ~ 42 TCTTTGACTTCATGTCTCATATT

MAPQ ~ 42 TTGACTTCATGTCTCATATTCAG

MAPQ ~ 8 TTGACTTCATGTCTCATATTCTG

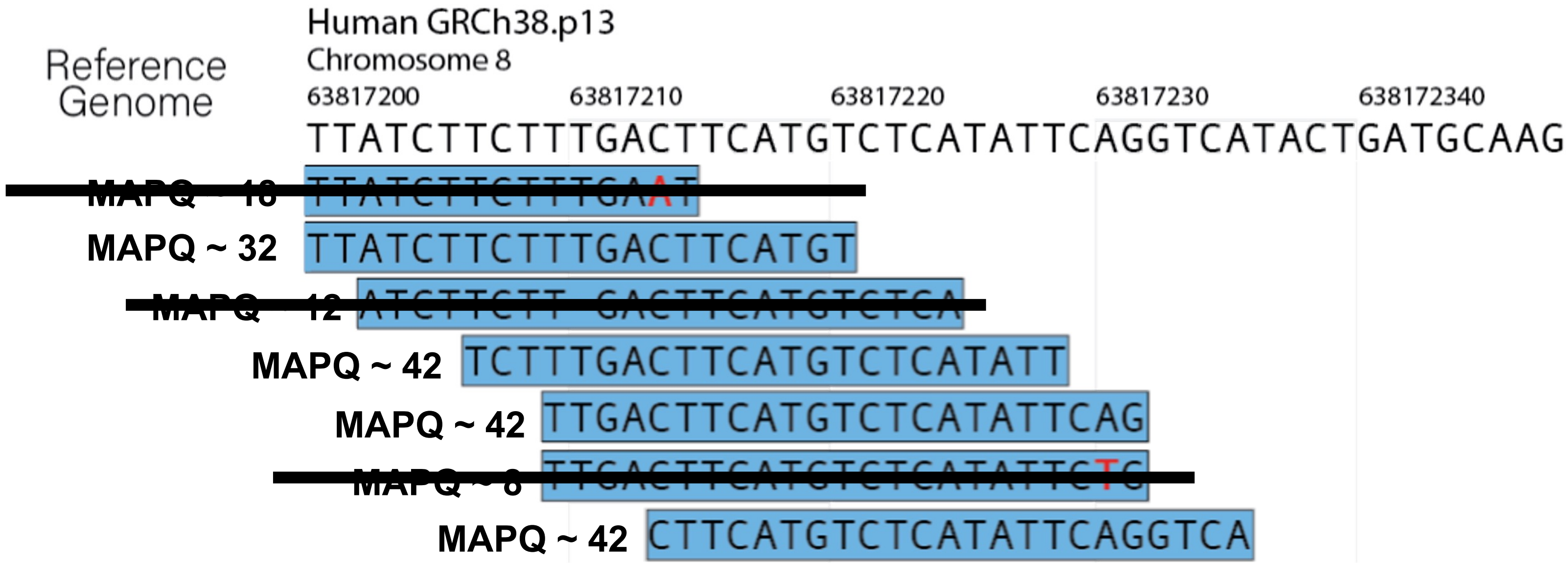
MAPQ ~ 42 CTTCATGTCTCATATTCAGGTCA

Filtering out low-quality mapping reads

Mapping Quality Scores (**MAPQ**) quantify the probability that a read is misplaced.

$$MAPQ = -10 * log_{10}\big(P(read\ is\ wrongly\ mapped)\big)$$

For example, a MAPQ score of 20 indicates that the probability for the read to be map at the indicated position is 0.01.



NGS processing workflow

✓ Get .bcl files

✓ Create fastq files

Or **`bcl2fastq`**

✓ QC: remove/trim low quality reads — E.g. **`cutadapt`**

✓ Align fastq to BAM — E.g. **`bowtie2`**

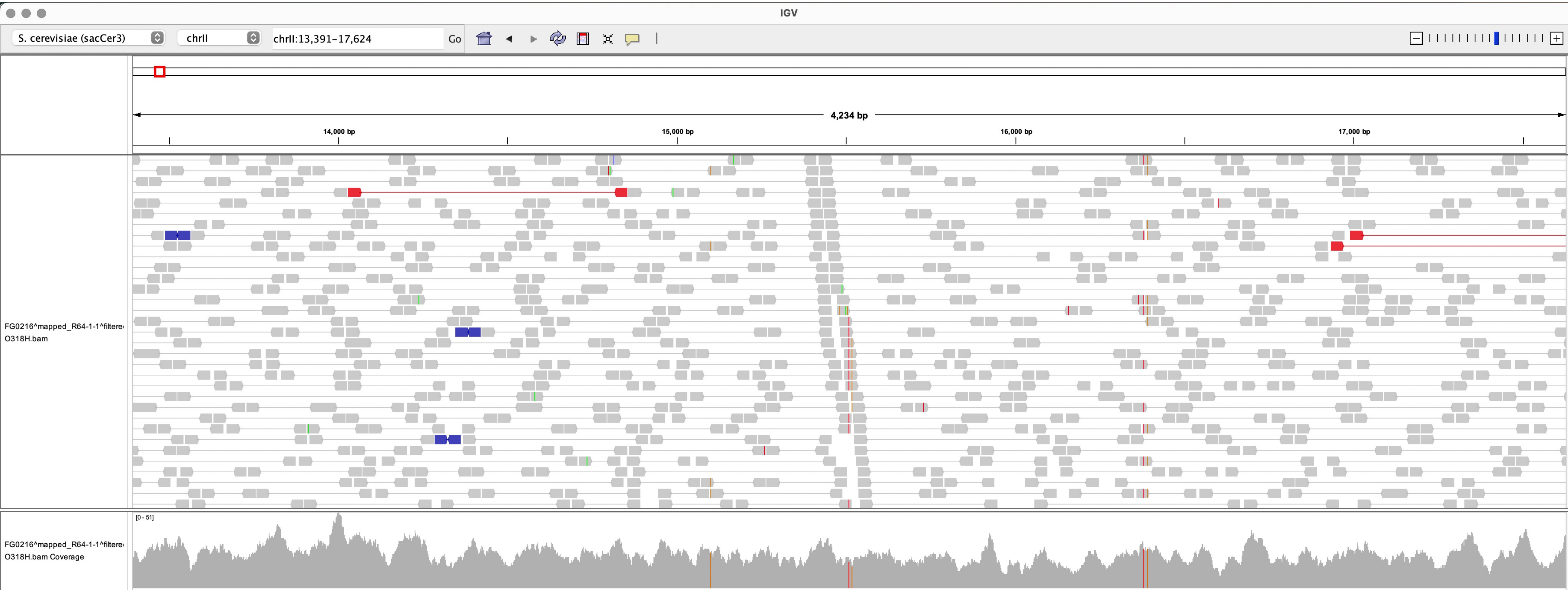
✓ Filter duplicates, artifacts, … — E.g. **`samtools`**

● Generate tracks

● Assay-specific downstream analysis

Generating tracks from mapped reads

Basic approach: ”pile-up” of all the fragments in `bam` files to generate a **coverage track**.



S. cerevisiae (sacCer3)

chrII

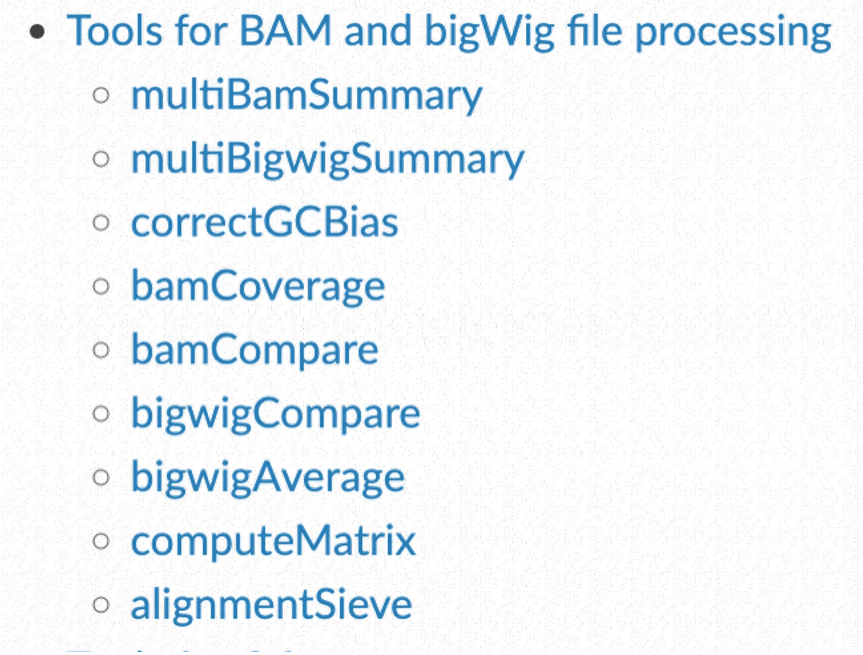
chrII:13,391-17,624

FG0216^mapped_R64-1-1^filtere O318H.bam

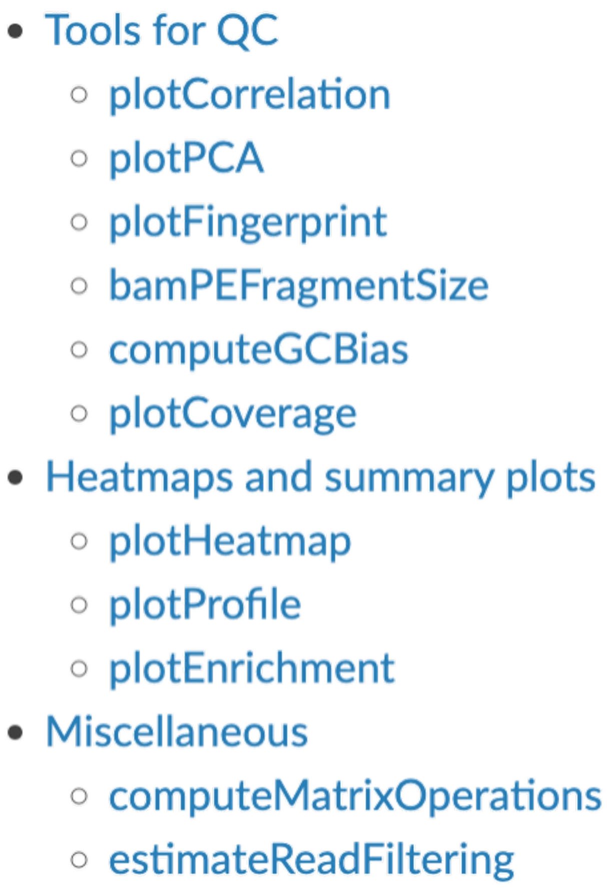
FG0216^mapped_R64-1-1^filtere O318H.bam Coverage

deepTools: a software suite to manage/produce genomic tracks

https://deeptools.readthedocs.io/en/latest/content/list_of_tools.html

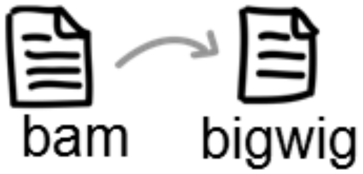


- Tools for BAM and bigWig file processing
  - multiBamSummary
  - multiBigwigSummary
  - correctGCBias
  - bamCoverage
  - bamCompare
  - bigwigCompare
  - bigwigAverage
  - computeMatrix
  - alignmentSieve

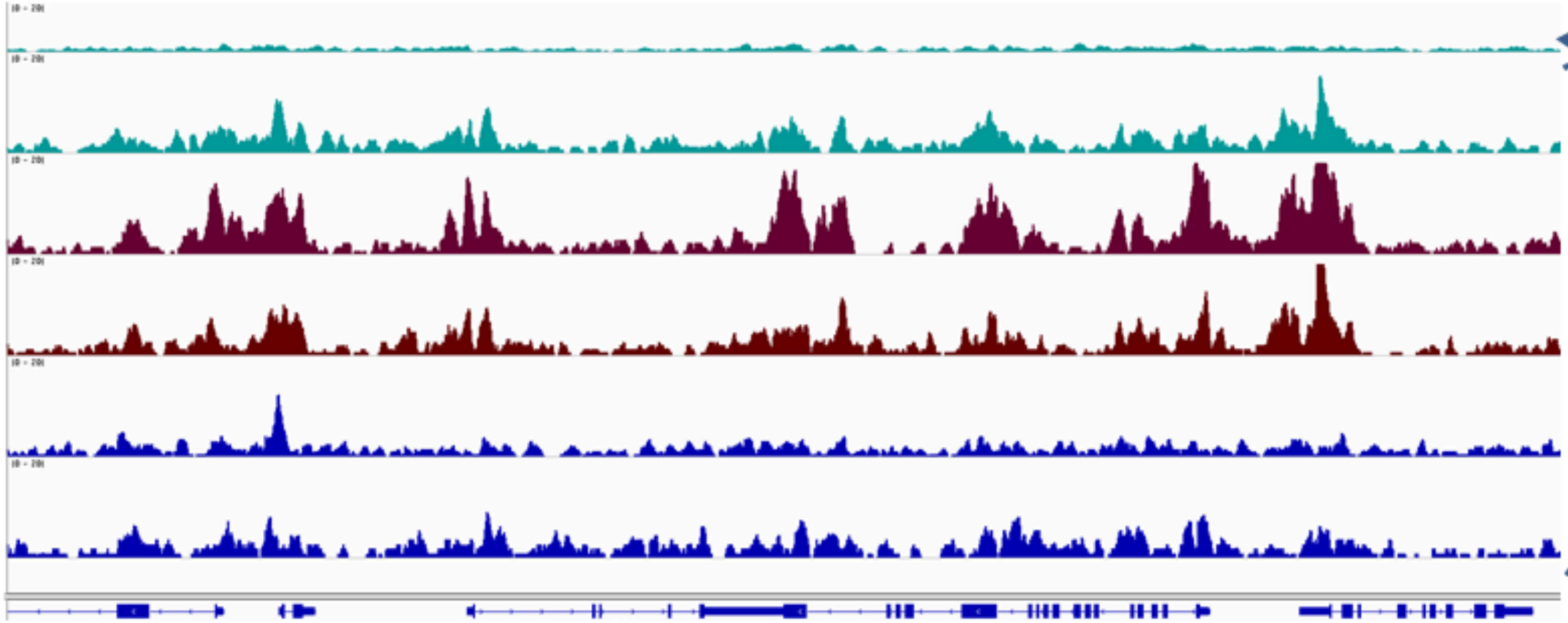


- Tools for QC
  - plotCorrelation
  - plotPCA
  - plotFingerprint
  - bamPEFragmentSize
  - computeGCBias
  - plotCoverage
- Heatmaps and summary plots
  - plotHeatmap
  - plotProfile
  - plotEnrichment
- Miscellaneous
  - computeMatrixOperations
  - estimateReadFiltering

deepTools: a software suite to manage/produce genomic tracks



for visualizing continuous data, e.g. in the UCSC Genome Browser or IGV, bigWig files come in really handy



input

different ChIP-seq samples

remember that there are 2 deepTools for bam → bigWig conversion:

- ***bamCoverage:*** *for individual files (like those shown here)*
- ***bamCompare:*** *to normalize two files to each other*

NGS processing workflow

✓	Get .bcl files	
✓	Create fastq files	Or `bcl2fastq`
✓	QC: remove/trim low quality reads	E.g. `cutadapt`
✓	Align fastq to BAM	E.g. `bowtie2`
✓	Filter duplicates, artifacts, …	E.g. `samtools`
✓	Generate tracks	E.g. `deepTools`
●	Assay-specific downstream analysis	

NGS processing workflow

✓ Get .bcl files

✓ Create fastq files

Or **`bcl2fastq`**

✓ QC: remove/trim low quality reads — E.g. **`cutadapt`**

✓ Align fastq to BAM — E.g. **`bowtie2`**

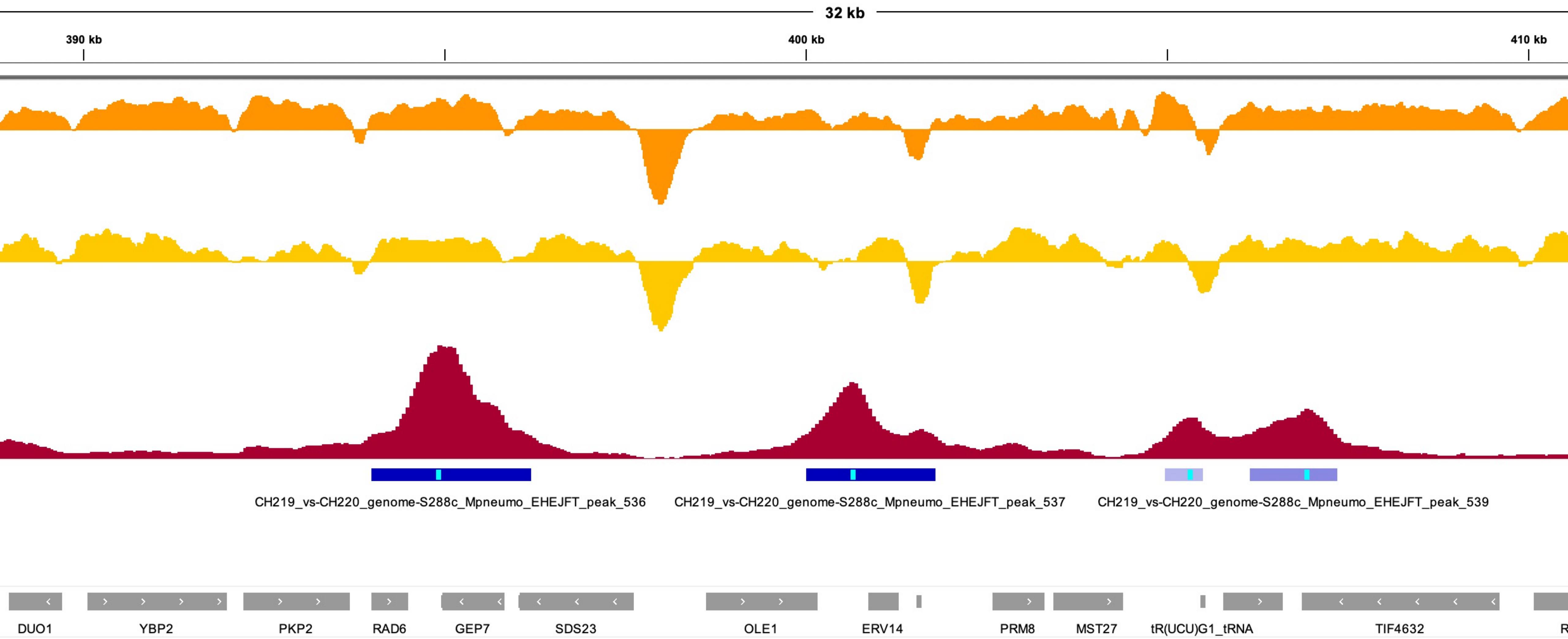
✓ Filter duplicates, artifacts, … — E.g. **`samtools`**

✓ Generate tracks — E.g. **`deepTools`**

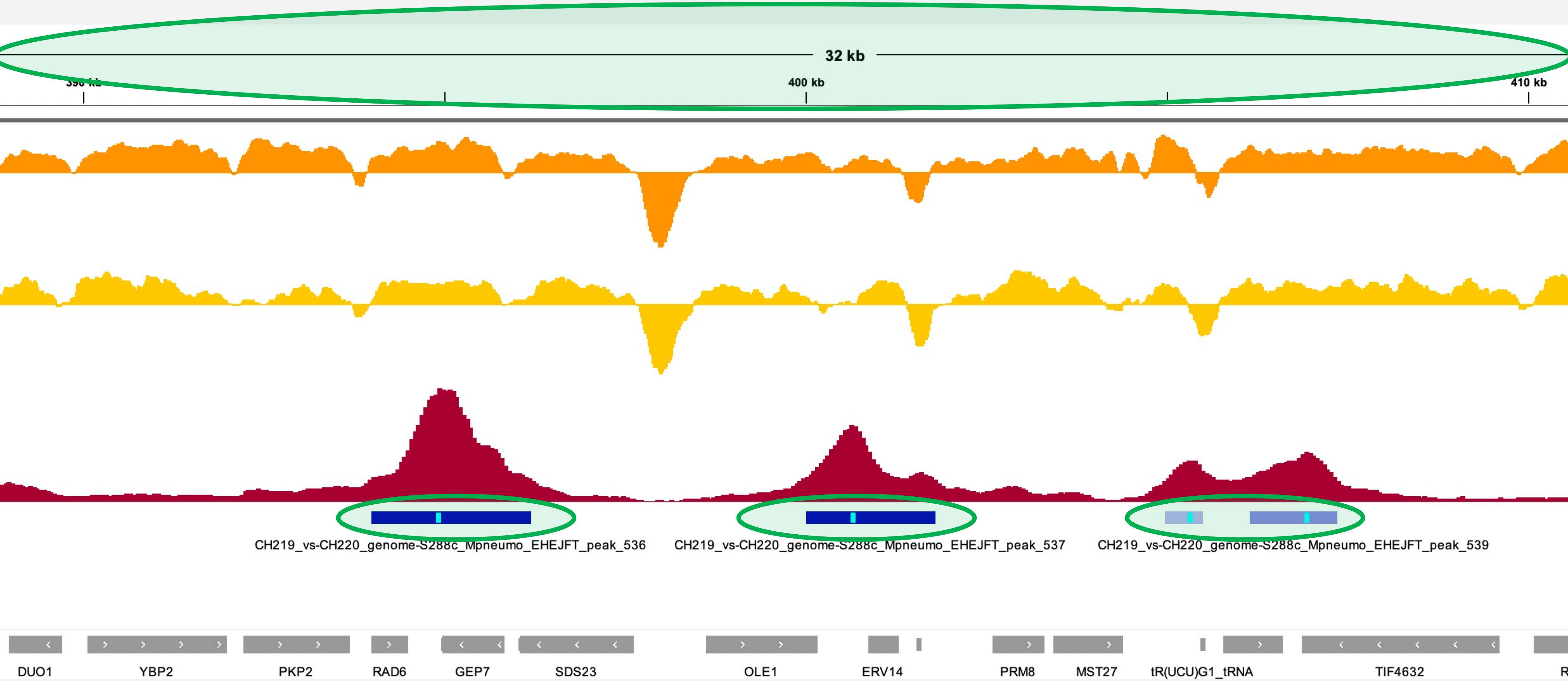
CHECK YOUR DATA

● Assay-specific downstream analysis

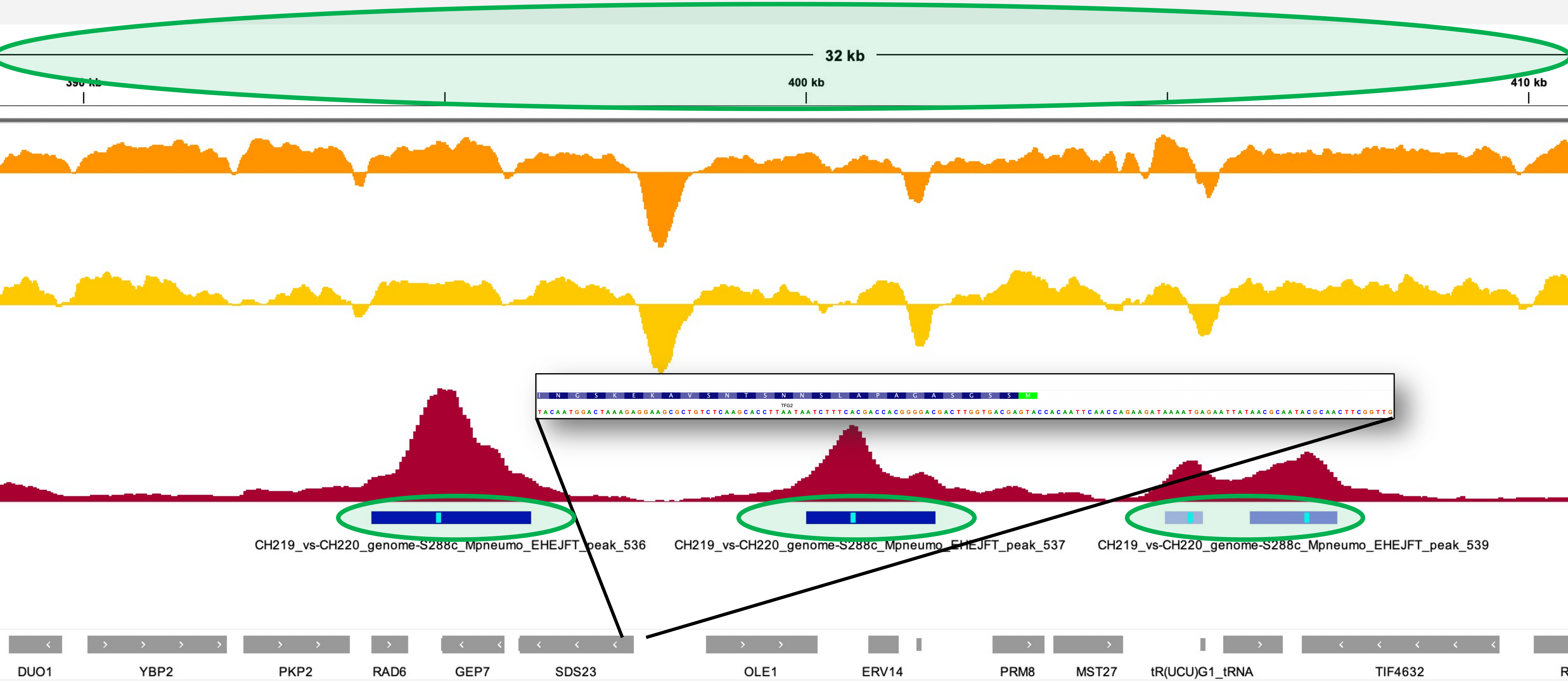
Epigenomics in a browser


32 kb
390 kb
400 kb
410 kb
CH219_vs-CH220_genome-S288c_Mpneumo_EHEJFT_peak_536
CH219_vs-CH220_genome-S288c_Mpneumo_EHEJFT_peak_537
CH219_vs-CH220_genome-S288c_Mpneumo_EHEJFT_peak_539
DUO1
YBP2
PKP2
RAD6
GEP7
SDS23
OLE1
ERV14
PRM8
MST27
tR(UCU)G1_tRNA
TIF4632
R

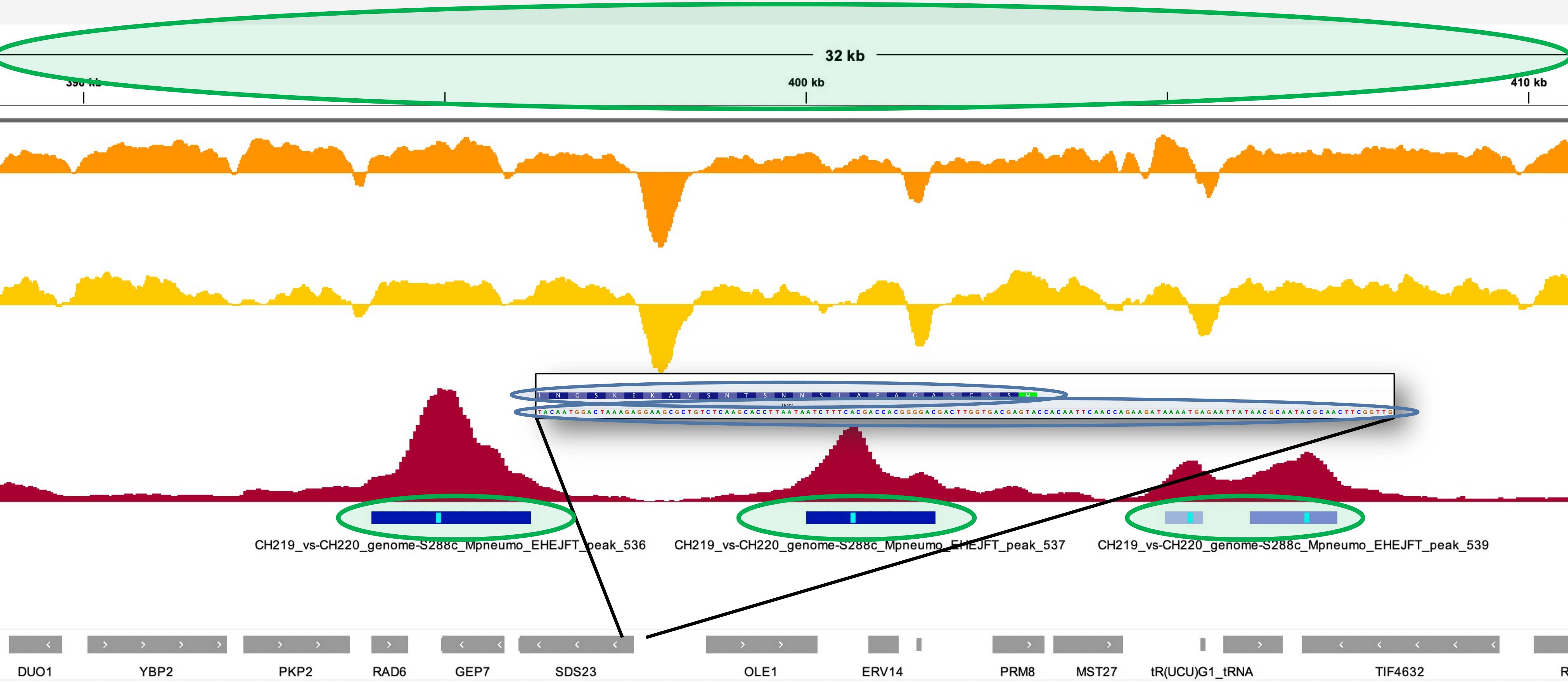
Epigenomics in a browser


32 kb
390 kb
400 kb
410 kb
CH219_vs-CH220_genome-S288c_Mpneumo_EHEJFT_peak_536
CH219_vs-CH220_genome-S288c_Mpneumo_EHEJFT_peak_537
CH219_vs-CH220_genome-S288c_Mpneumo_EHEJFT_peak_539
DUO1
YBP2
PKP2
RAD6
GEP7
SDS23
OLE1
ERV14
PRM8
MST27
tR(UCU)G1_tRNA
TIF4632
R

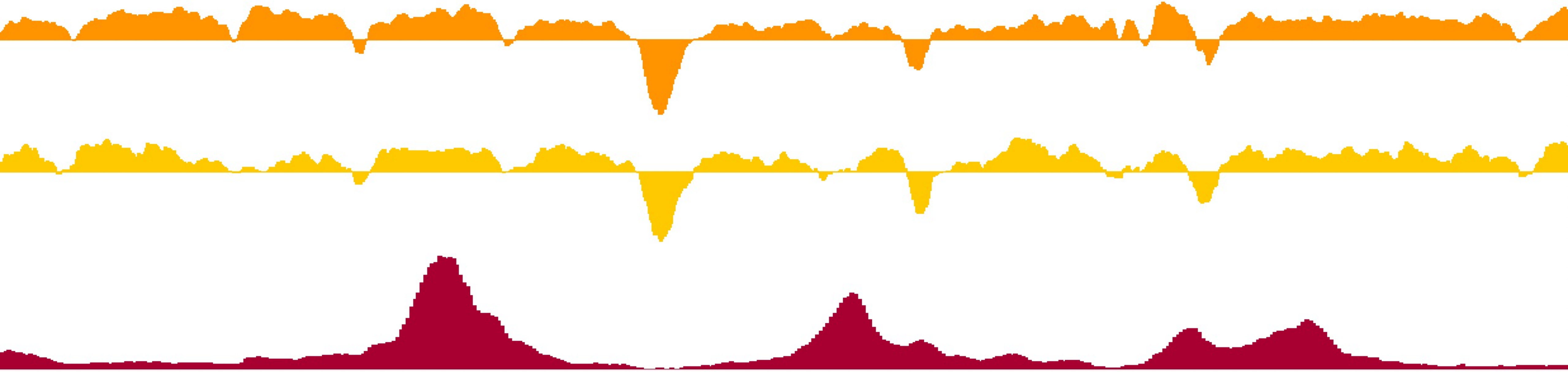
Epigenomics in a browser


32 kb
390 kb
400 kb
410 kb
I N G S K E K A V S N T S N N S L A P A G A S G S S M
TFG2
TACAATGGACTAAAGAGGAAGCGCTGTCTCAAGCACCTTAATAATCTTTCACGACCACGGGGACGACTTGGTGACGAGTACCACAATTCAACCAGAAGATAAAATGAGAATTATAACGCAATACGCAACTTCGGTTG
CH219_vs-CH220_genome-S288c_Mpneumo_EHEJFT_peak_536
CH219_vs-CH220_genome-S288c_Mpneumo_EHEJFT_peak_537
CH219_vs-CH220_genome-S288c_Mpneumo_EHEJFT_peak_539
DUO1
YBP2
PKP2
RAD6
GEP7
SDS23
OLE1
ERV14
PRM8
MST27
tR(UCU)G1_tRNA
TIF4632
R

Epigenomics in a browser


32 kb
390 kb
400 kb
410 kb
I N G S K E K A V S N T S N N S L A P A G A S G S S M
TACAATGGACTAAAGAGGAAGCGCTGTCTCAAGCACCTTAATAATCTTTCACGACCACGGGGACGACTTGGTGACGAGTACCACAATTCAACCAGAAGATAAAATGAGAATTATAACGCAATACGCAACTTCGGTTG
CH219_vs-CH220_genome-S288c_Mpneumo_EHEJFT_peak_536
CH219_vs-CH220_genome-S288c_Mpneumo_EHEJFT_peak_537
CH219_vs-CH220_genome-S288c_Mpneumo_EHEJFT_peak_539
DUO1
YBP2
PKP2
RAD6
GEP7
SDS23
OLE1
ERV14
PRM8
MST27
tR(UCU)G1_tRNA
TIF4632
R

Epigenomics in a browser



Epigenomics in a browser

- Genomic tracks are generally stored as bigwig files.

- bigwig files store long numerical vectors in a binarized format

```
I          2          5          0.153096
I          5          7          0.459288
I          7          9          0.612384
I          9          11         0.76548
I          11         15         0.918576
I          15         16         1.07167
I          16         17         1.37786
I          17         30         1.68406
```

Epigenomics in a browser

- Genomic tracks are generally stored as bigwig files.

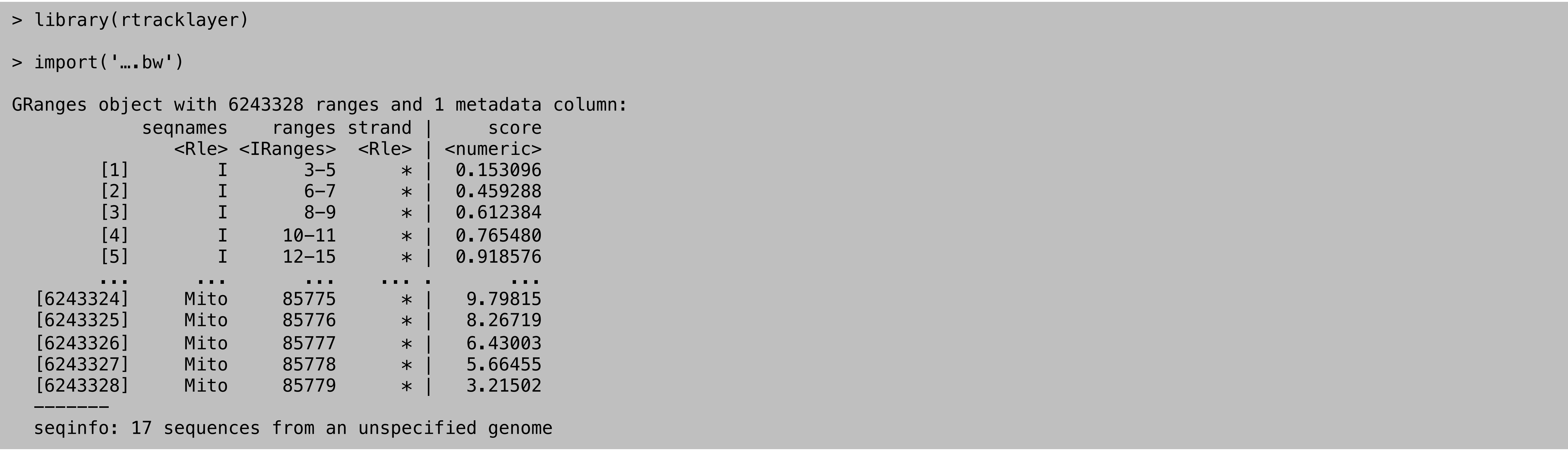
- bigwig files store long numerical vectors in a binarized format

```
I          2          5          0.153096
I          5          7          0.459288
I          7          9          0.612384
I          9          11         0.76548
I          11         15         0.918576
I          15         16         1.07167
I          16         17         1.37786
I          17         30         1.68406
```

```
�&XHA'\0�eJ]R�@6���09��0|]<pc<�d?�f?�,@@,−@~_@K�`���?�V�@53d�ej�A��i�B���IJ�II�h
                                                                            III��IV`VjVIQVII|�VIIIIP�X     a
                                                                                                              XI
,�
XII
�   sXIII
       ��XIV
  XVXVIbwMito0x�=�    �U�������{��so�J�2KD��V2U�d(���2RQ
���BÙJ�
�4(2
����'��[i����JM���~X�w��9"��Y���Jp<�����4ں|YQ��4NL[e��X_���[ub�S80'����i��S|$mÓ3|"�m�O�mM����N�i�ū�$�,k�!�:X�Y��NO�[&xn��\���i[Y94�8O�nH�F����
[f��ےi�bشg2�]�'8;mm���s|��H�ۆF&�f�,�Jp����N��s�$�n�n��{��9��=�ك□����~s�6�'�����4�ω   .e�&%��������9�J��w1�]����ㅅ−��zl뙩w�f�:3���N����`gsḻKc�−
��f���c�−Z4;�����I��|ō̈%��������`/s<���/���m���c?s<�}��=>x�9�i=��8x�9���|k���/x�9^l���−
3{n�W��8�l_b�ↄ�P�78&��8�1�����9m����ٍ�o%82mc�|;�[X���M�Vse���6���  86$?������{�□m����]�_
```

Epigenomics in a browser

- Genomic tracks are generally stored as bigwig files.
- bigwig files store long numerical vectors in a binarized format
- In R, bigwig files can be imported with `` `import()` `` from the `` `rtracklayer` `` package



```
> library(rtracklayer)

> import('….bw')

GRanges object with 6243328 ranges and 1 metadata column:
            seqnames    ranges strand |     score
               <Rle> <IRanges>  <Rle> | <numeric>
        [1]        I       3-5      * |  0.153096
        [2]        I       6-7      * |  0.459288
        [3]        I       8-9      * |  0.612384
        [4]        I     10-11      * |  0.765480
        [5]        I     12-15      * |  0.918576
        ...      ...       ...    ... .       ...
  [6243324]     Mito     85775      * |   9.79815
  [6243325]     Mito     85776      * |   8.26719
  [6243326]     Mito     85777      * |   6.43003
  [6243327]     Mito     85778      * |   5.66455
  [6243328]     Mito     85779      * |   3.21502
  -------
  seqinfo: 17 sequences from an unspecified genome
```

Run-length encoding vectors

- Genomic tracks are generally stored as bigwig files.
- bigwig files store long numerical vectors in a binarized format.
- In R, bigwig files can be imported with `import()` from the `rtracklayer` package.
- bigwig files can be imported as **numerical vectors**, stored as **Run-length encoding vectors**.

```
b b b b k k e e f a a a a a a a a g g g
```

Run-length encoding vectors

- Genomic tracks are generally stored as bigwig files.
- bigwig files store long numerical vectors in a binarized format.
- In R, bigwig files can be imported with `` `import()` `` from the `` `rtracklayer` `` package.
- bigwig files can be imported as **numerical vectors**, stored as **Run-length encoding vectors**.

b b b b k k e e f a a a a a a a a g g g

4 b

Run-length encoding vectors

- Genomic tracks are generally stored as bigwig files.
- bigwig files store long numerical vectors in a binarized format.
- In R, bigwig files can be imported with `` `import()` `` from the `` `rtracklayer` `` package.
- bigwig files can be imported as **numerical vectors**, stored as **Run-length encoding vectors**.

b b b b **k k** e e f a a a a a a a a g g g

4 b

2 k

Run-length encoding vectors

- Genomic tracks are generally stored as bigwig files.
- bigwig files store long numerical vectors in a binarized format.
- In R, bigwig files can be imported with `` `import()` `` from the `` `rtracklayer` `` package.
- bigwig files can be imported as **numerical vectors**, stored as **Run-length encoding vectors**.

b b b b k k **e e** f a a a a a a a a g g g

4 b

2 k

2 e

Run-length encoding vectors

- Genomic tracks are generally stored as bigwig files.
- bigwig files store long numerical vectors in a binarized format.
- In R, bigwig files can be imported with `` `import()` `` from the `` `rtracklayer` `` package.
- bigwig files can be imported as **numerical vectors**, stored as **Run-length encoding vectors**.

```
b b b b k k e e f a a a a a a a a g g g
```

```
4 b
2 k
2 e
1 f
```

Run-length encoding vectors

- Genomic tracks are generally stored as bigwig files.
- bigwig files store long numerical vectors in a binarized format.
- In R, bigwig files can be imported with `` `import()` `` from the `` `rtracklayer` `` package.
- bigwig files can be imported as **numerical vectors**, stored as **Run-length encoding vectors**.

b b b b k k e e f **a a a a a a a a** g g g

4 b

2 k

2 e

1 f

8 a

Run-length encoding vectors

- Genomic tracks are generally stored as bigwig files.
- bigwig files store long numerical vectors in a binarized format.
- In R, bigwig files can be imported with `import()` from the `rtracklayer` package.
- bigwig files can be imported as **numerical vectors**, stored as **Run-length encoding vectors**.

```
b b b b k k e e f a a a a a a a a g g g
```

```
4 b
2 k
2 e
1 f
8 a
3 g
```

Run-length encoding vectors

- Genomic tracks are generally stored as bigwig files.
- bigwig files store long numerical vectors in a binarized format.
- In R, bigwig files can be imported with `` `import()` `` from the `` `rtracklayer` `` package.
- bigwig files can be imported as **numerical vectors**, stored as **Run-length encoding vectors**.

```
b b b b k k e e f a a a a a a a a g g g
```

```
Run-values:  b k e f a g
Run-lengths: 4 2 2 1 8 3
```

Run-length encoding vectors

- Genomic tracks are generally stored as bigwig files.
- bigwig files store long numerical vectors in a binarized format.
- In R, bigwig files can be imported with `` `import()` `` from the `` `rtracklayer` `` package.
- bigwig files can be imported as **numerical vectors**, stored as **Run-length encoding vectors**.

```
b b b b k k e e f a a a a a a a a g g g
```

```
Run-values:  b k e f a g
Run-lengths: 4 2 2 1 8 3
```

12 alpha-numeric values instead of 20 alphabetic values

Epigenomics in a browser

- Genomic tracks are generally stored as bigwig files.

- bigwig files store long numerical vectors in a binarized format.

- In R, bigwig files can be imported with `` `import()` `` from the `` `rtracklayer` `` package.

- bigwig files can be imported as **numerical vectors**, stored as **Run-length encoding vectors**.

```
> library(rtracklayer)

> import('….bw')

GRanges object with 6243328 ranges and 1 metadata column:
            seqnames    ranges strand |     score
               <Rle> <IRanges>  <Rle> | <numeric>
        [1]        I       3-5      * |  0.153096
        [2]        I       6-7      * |  0.459288
        [3]        I       8-9      * |  0.612384
        [4]        I     10-11      * |  0.765480
        [5]        I     12-15      * |  0.918576
        ...      ...       ...    ... .       ...
  [6243324]     Mito     85775      * |   9.79815
  [6243325]     Mito     85776      * |   8.26719
  [6243326]     Mito     85777      * |   6.43003
  [6243327]     Mito     85778      * |   5.66455
  [6243328]     Mito     85779      * |   3.21502
  -------
  seqinfo: 17 sequences from an unspecified genome
```

Epigenomics in a browser

- Genomic tracks are generally stored as bigwig files.
- bigwig files store long numerical vectors in a binarized format.
- In R, bigwig files can be imported with `` `import()` `` from the `` `rtracklayer` `` package.
- bigwig files can be imported as **numerical vectors**, stored as **Run-length encoding vectors**.

```
> import('….bw', as = ‘Rle')

RleList of length 17
$I
numeric-Rle of length 230218 with 104639 runs
  Lengths:        2        3        2        2        2 ...        2       30        1     1084
  Values :  0.000000  0.153096  0.459288  0.612384  0.765480 ...  0.612384  0.459288  0.306192  0.000000

$II
numeric-Rle of length 813184 with 424729 runs
  Lengths:        2        1        1        2        1 ...         6        1        5        2
  Values :  0.153096  0.306192  0.612384  0.918576  1.071670 ...   0.459288  0.306192  0.153096  0.000000

...
<15 more elements>
```