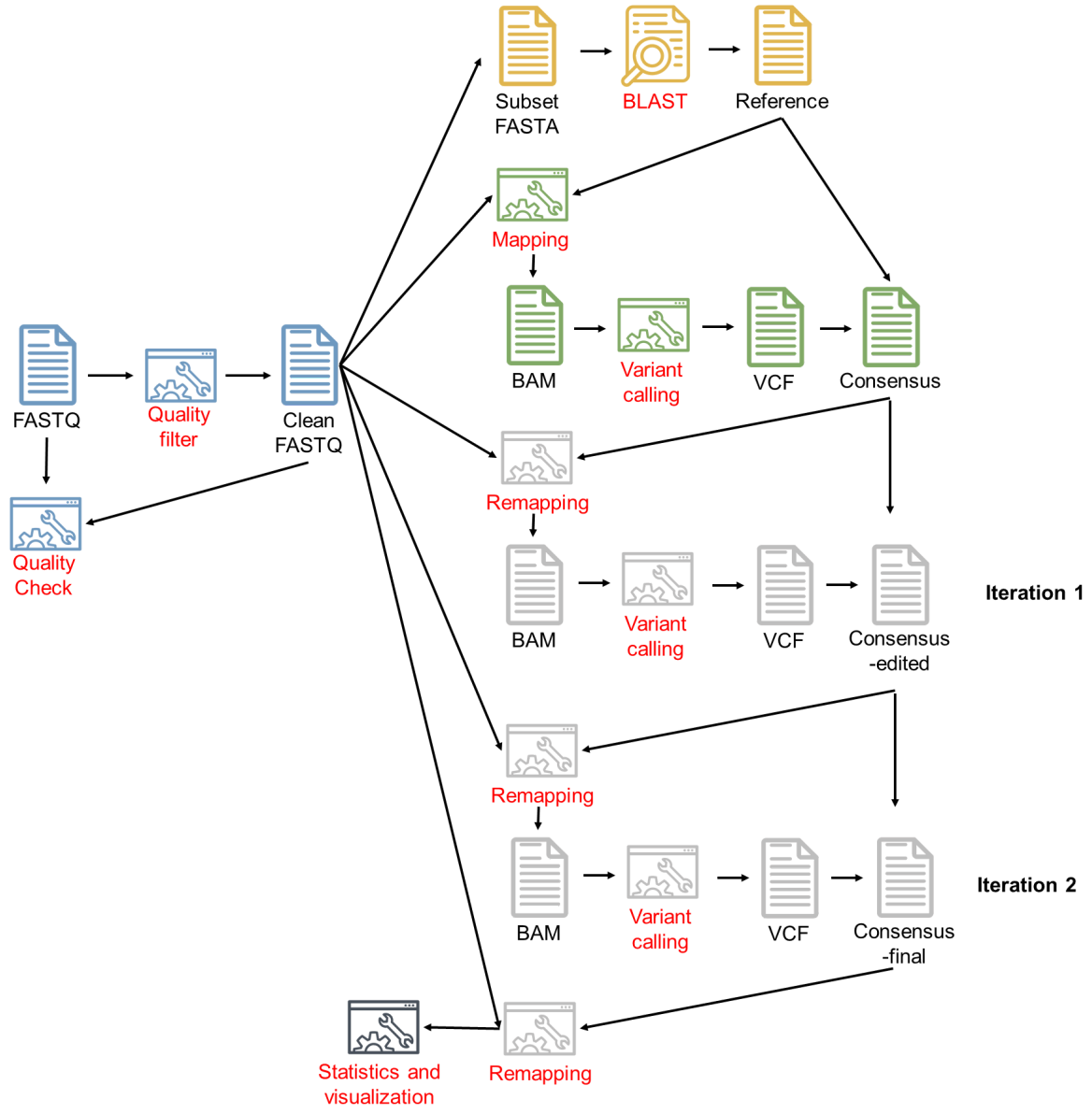
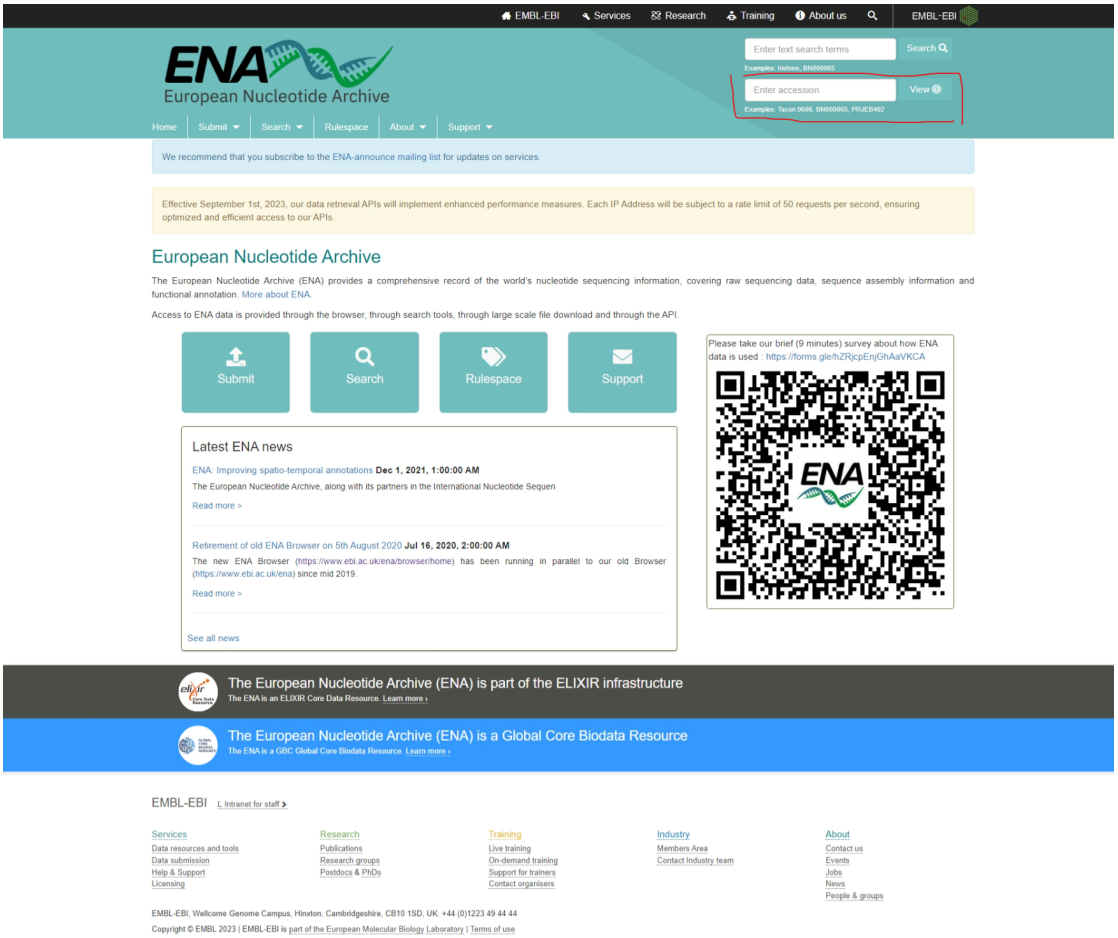
# Reference assembly with short reads

## Overview




## Download the data

1. Go to ENA Browser (ebi.ac.uk)
2. Type SRR25266112 in the search bar.




3. Right click and copy the link of the “Generated FASTQ files: FTP”
4. Copy the link to the fastq files and run wget + link.

## Quality control and filtering

1. Quality check

```
fastqc -o qc_raw -f fastq SRR25266112_1.fastq.gz SRR25266112_2.fastq.gz
```

2. Quality filtering

```
#run trimmomatic
trimmomatic PE \
-threads 16 \
-phred33 \
-trimlog log.txt \
SRR25266112_1.fastq.gz \
SRR25266112_2.fastq.gz \
./clean/SRR25266112paired_1.fastq.gz \
./clean/SRR25266112unpaired_1.fastq.gz \
./clean/SRR25266112paired_2.fastq.gz \
./clean/SRR25266112unpaired_2.fastq.gz \
LEADING:20 \
TRAILING:20 \
SLIDINGWINDOW:4:20 \
AVGQUAL:20 \
MINLEN:100
```

**PE** → input is paired end reads
**phred33** → use phred33 scoring system

**trimlog** → file to keep the output log
**LEADING:20** → trim bases at the front if quality below threshold (20).
**TRAILING:20** → trim bases at the end if quality below threshold (20).
**SLIDINGWINDOW:4:20** → perform sliding window trimming: check the quality every 4 nucleotides, trim when quality falls below the threshold (20).
**AVGQUAL:20** → remove read if the average base quality is below threshold (20).
**MINLEN:100** → remove read if the length is shorter than threshold (100).
If adapter sequences need to be trimmed - add the **ILLUMINACLIP** option.

3. Quality check post-filtering

```
fastqc -o qc_clean -f fastq ./clean/SRR25266112paired_1.fastq.gz ./clean/SRR25266112paired_2.fastq.gz
```

## Mapping

We will use the **sample01.fastq, sample02.fastq, and sample03.fastq** located in ./**training/fastq/**.

1. BLAST search
   a. Subset 1000 reads from the file.

```
#make a directory for the analysis
mkdir ./SRR25266112/

#subset 1000 reads for blast search
zcat ./clean/SRR25266112paired_1.fastq.gz | \
head -4000 | \
seqkit fq2fa -w 0 \
> ./SRR25266112/subsetSRR25266112.fasta
```

`zcat`: similar to cat, but for compressed file.
`head -4000`: subset 4000 lines from the beginning.
`seqkit fq2fa`: transform fastq file to fasta file.
`-w 0`: print the sequences in one line.

   b. Run blast

```
blastn -query ./SRR25266112/subsetSRR25266112.fasta \
-db ./db/fluadb.fa \
-outfmt "6 qseqid bitscore pident length sseqid stitle gapopen qstart qend sstart send evalue bitscore qlen slen" \
-num_threads 16 -perc_identity 90 -max_target_seqs 1 -out ./SRR25266112/blastresSRR25266112.txt &
```

`-query`: the input file (fasta).
`-outfmt`: the type of format to output the result. In this example we asked for format 6 (tabular).
`-num_threads`: number of threads to use.
`-perc_identity`: threshold for percent identity (`pident`).
`-mar_target_seqs`: number of hits for each sequence.
`-out`: output filed
`&`: tell the shell to run the command on the background.

Quickly check the result.

```
more ./SRR25266112/blastresSRR25266112.txt
```

c. sort the results

```
grep "|PB2" ./SRR25266112/blastresSRR25266112.txt | sort -k 2n | tail -1 | awk '{print $5}' > ./SRR25266112/ref.acc
grep "|PB1" ./SRR25266112/blastresSRR25266112.txt| sort -k 2n | tail -1 | awk '{print $5}' >> ./SRR25266112/ref.acc
grep "|PA" ./SRR25266112/blastresSRR25266112.txt | sort -k 2n | tail -1 | awk '{print $5}' >> ./SRR25266112/ref.acc
grep "|HA" ./SRR25266112/blastresSRR25266112.txt | sort -k 2n | tail -1 | awk '{print $5}' >> ./SRR25266112/ref.acc
grep "|NA" ./SRR25266112/blastresSRR25266112.txt | sort -k 2n | tail -1 | awk '{print $5}' >> ./SRR25266112/ref.acc
grep "|NP" ./SRR25266112/blastresSRR25266112.txt | sort -k 2n | tail -1 | awk '{print $5}' >> ./SRR25266112/ref.acc
grep "|MP" ./SRR25266112/blastresSRR25266112.txt | sort -k 2n | tail -1 | awk '{print $5}' >> ./SRR25266112/ref.acc
grep "|NS" ./SRR25266112/blastresSRR25266112.txt | sort -k 2n | tail -1 | awk '{print $5}' >> ./SRR25266112/ref.acc
```

or

```
for gene in "|PB2" "|PB1" "|PA" "|HA" "|NA" "|NP" "|MP" "|NS"; do \
grep ${gene} ./SRR25266112/blastresSRR25266112.txt | sort -k 2n | tail -1 | awk '{print $5}';done \
> ./SRR25266112/ref.acc
```

`sort -k 2n`: sort the data based on column two and treat the data as numeric.
`tail -1`: print last line.
`awk '{print $5}'`: print the fifth column, can be replaced with `cut -f5`.

d. Check if all eight segments are available

```
more ./SRR25266112/ref.acc #or
wc -l ./SRR25266112/ref.acc #should return 8. If it is less than 8, consider to blast more sequences.
```

e. Extract the reference sequence from list of sequence used for blast database

```
#we will use loop and then mask any degenerate bases to N
while read line; \
do grep -A1 $line ./db/fluadb.fa; \
done < ./SRR25266112/ref.acc | sed '2~2s/[RYMKSWBDHV]/N/g' > ./SRR25266112/SRR25266112ref.fasta
```

The sed command replace the degenerate bases with N → because freebayes will interepret non-ATCG bases as N, which will clashes with `bcftools consensus`.

2. Mapping

Build index and mapping

```
bowtie2-build ./SRR25266112/SRR25266112ref.fasta ./SRR25266112/SRR25266112ref.fasta
```

Usage `bowtie2-build ref ref_name`

Mapping

```
bowtie2 -x ./SRR25266112/SRR25266112ref.fasta \
-1 ./clean/SRR25266112paired_1.fastq.gz \
-2 ./clean/SRR25266112paired_2.fastq.gz -p 8 \
-S ./SRR25266112/SRR25266112.sam
```

`-x` : bowtie2 index name
`-1` and `-2`: paired end reads
`-p` : number of threads
`-S` : sam file output location

Convert sam to bam and filter the unmapped reads.

```
samtools view -b -F 2052 ./SRR25266112/SRR25266112.sam | \
samtools sort > ./SRR25266112/SRR25266112.bam
```

`samtools view`: convert a sam/bam/cram file into a sam/bam/cram file.
`-b`: output as a bam file.

Index the bam file

```
samtools index ./SRR25266112/SRR25266112.bam
```

Check mapping stats

```
samtools idxstats ./SRR25266112/SRR25266112.bam
```

3. Consensus calling

   Generating a pileup file

```
bcftools mpileup --max-depth 10000 --max-idepth 10000 \
-f ./SRR25266112/SRR25266112ref.fasta ./SRR25266112/SRR25266112.bam \
> ./SRR25266112/call.bcl
```

`-Ou` : output as standard format, `-B` : do not recalculate the base alignment quality (BAQ), `-Q:` lower threshold BAQ, `--max-BQ` : upper threshold for BAQ, `--max-depth` : max coverage being considered when running pileup. Type `bcftools mpileup -h` for explanation of the full command, or go to https://samtools.github.io/bcftools/bcftools.html#mpileup.

Call the variants

```
bcftools call -c -Oz --ploidy 1 -p 0.01 ./SRR25266112/call.bcl > ./SRR25266112/call.vcf.gz
bcftools index ./SRR25266112/call.vcf.gz
```

`-c` : classic consensus caller, `--ploidy` : the ploidy of the organism, `-p` : p-value threshold.

Apply variants to reference

```
bcftools consensus -f ./SRR25266112/SRR25266112ref.fasta  \
-H I ./SRR25266112/call.vcf.gz | seqkit seq -w 0 \
> ./SRR25266112/consensus_temp.fa
```

`-f` : reference fasta file
`-H I` : haplotype, IUPAC code for all genotypes
`seqkit seq` : sequence editor
`-w 0:` concatenate all sequence in one line.

4. Realignment

   If we are to align our reads back to the consensus, a lot of errors can be observed.

   Build index and mapping

```
bowtie2-build ./SRR25266112/consensus_temp.fa ./SRR25266112/consensus_temp.fa

bowtie2 -x ./SRR25266112/consensus_temp.fa  \
-1 ./clean/SRR25266112paired_1.fastq.gz -2 \
./clean/SRR25266112paired_2.fastq.gz -p 8 \
-S ./SRR25266112/aligntoconsensus.samAlingment
```

   Sorting and indexing

```
samtools view -bS -F 2052 ./SRR25266112/aligntoconsensus.sam | samtools sort > ./SRR25266112/aligntoconsensus.bam
samtools index ./SRR25266112/aligntoconsensus.bam
```

   Download the aligntoconsensus.bam, aligntoconsensus.bam.bai, and consensus_temp.fa to visualize in IGV.


   These errors stem from the reference selection. Because the sequences are short, the blast hit only represent a certain region of the genome. The similar length of the sequence also complicate the reference score based on bitscore. Iteration of correction and realignment can improve the result.

5. Iteration

   We will “improve” the reference based on the alignment.

   **First correction**

   Get variants:

```
freebayes -f ./SRR25266112/SRR25266112ref.fasta -p 1 -P 0.01 \
```

```
./SRR25266112/SRR25266112.bam > ./SRR25266112/var.vcf
```

`-f`: reference fasta file
`-p`: ploidy
`-P`: p-value threshold

Compress then index the vcf file:

```
bgzip ./SRR25266112/var.vcf && bcftools index ./SRR25266112/var.vcf.gz
```

Apply the variants to the reference

```
bcftools consensus -f ./SRR25266112/SRR25266112ref.fasta -H I \
./SRR25266112/var.vcf.gz | seqkit -seq -w 0 > ./SRR25266112/draft1.fa
```

`-f`: reference fasta file
`--mark-del`: mark deletion with certain character.
`-a`: mark missing base with certain character.
`-H 1`: output only the first genotype

**First iteration** - align the read to the the corrected first draft

Build index and mapping

```
bowtie2-build ./SRR25266112/draft1.fa ./SRR25266112/draft1.fa

bowtie2 -x ./SRR25266112/draft1.fa  \
-1 ./clean/SRR25266112paired_1.fastq.gz \
-2 ./clean/SRR25266112paired_2.fastq.gz -p 8 \
-S ./SRR25266112/aln2.sam
```

Sam to bam

```
samtools view -bS -F 2052 ./SRR25266112/aln2.sam | samtools sort \
> ./SRR25266112/aln2.bam
```

Index the bam file

```
samtools index ./SRR25266112/aln2.bam
```

Run freebayes

```
freebayes -f ./SRR25266112/draft1.fa -p 1 -P 0.01 ./SRR25266112/aln2.bam \
> ./SRR25266112/var2.vcf
```

Compressed and index the vcf

```
bgzip ./SRR25266112/var2.vcf && bcftools index ./SRR25266112/var2.vcf.gz
```

Apply the variants to the reference

```
bcftools consensus -f ./SRR25266112/draft1.fa -H I \
./SRR25266112/var2.vcf.gz | seqkit seq -w 0 > ./SRR25266112/draft2.fa
```

**Second iteration**

Build index and mapping

```
bowtie2-build ./SRR25266112/draft2.fa ./SRR25266112/draft2.fa

bowtie2 -x ./SRR25266112/draft2.fa  \
-1 ./clean/SRR25266112paired_1.fastq.gz \
-2 ./clean/SRR25266112paired_2.fastq.gz -p 8 \
-S ./SRR25266112/aln3.sam
```

Sam to bam

```
samtools view -bS -F 2052 ./SRR25266112/aln3.sam | samtools sort \
> ./SRR25266112/aln3.bam
```

Index the bam file

```
samtools index ./SRR25266112/aln3.bam
```

Run freebayes

```
freebayes -f ./SRR25266112/draft2.fa -p 1 -P 0.01 ./SRR25266112/aln3.bam \
> ./SRR25266112/var3.vcf
```

Compressed and index the vcf

```
bgzip ./SRR25266112/var3.vcf && bcftools index ./SRR25266112/var3.vcf.gz
```

Apply the variants to the reference

```
bcftools consensus -f ./SRR25266112/draft2.fa -H I \
./SRR25266112/var3.vcf.gz | seqkit seq -w 0 > ./SRR25266112/draft3.fa
```

5. Wrap up

Create a file with old and new fasta header:

```
grep ">" ./SRR25266112/draft3.fa | cut -d'|' -f4 | sed 's/^/sample01_/' \
> ./SRR25266112/newheader01

paste ./SRR25266112/ref.acc ./SRR25266112/newheader01 > ./SRR25266112/header01.txt
```

Use seqkit to replace the old header with the new header:

```
seqkit replace -p "(.+)" -r '{kv}' -w 0 -k ./SRR25266112/header01.txt \
./SRR25266112/consensus_temp.fa > ./SRR25266112/consensusSRR25266112.fasta
```

6. Remapping

   Build index and mapping

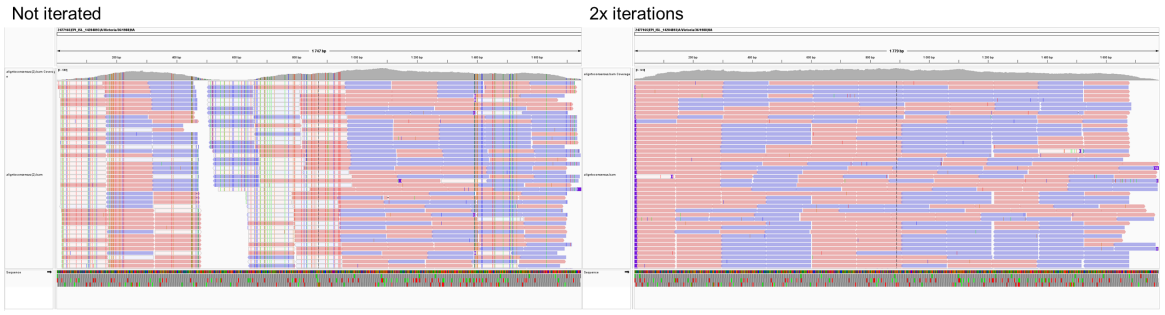
```
bowtie2-build ./SRR25266112/consensusSRR25266112.fasta \
./SRR25266112/consensusSRR25266112.fasta

bowtie2 -x ./SRR25266112/consensusSRR25266112.fasta  \
-1 ./clean/SRR25266112paired_1.fastq.gz -2 \
./clean/SRR25266112paired_2.fastq.gz -p 8 \
-S ./SRR25266112/remapping.sam
```

   Sam to bam and filtering

```
samtools view -bS -F 2052 ./SRR25266112/remapping.sam | \
samtools sort > ./SRR25266112/remapping.bam
```

   Index the bam file

```
samtools index ./SRR25266112/remapping.bam
```




Practice:
Perform a reference assembly for sample05.