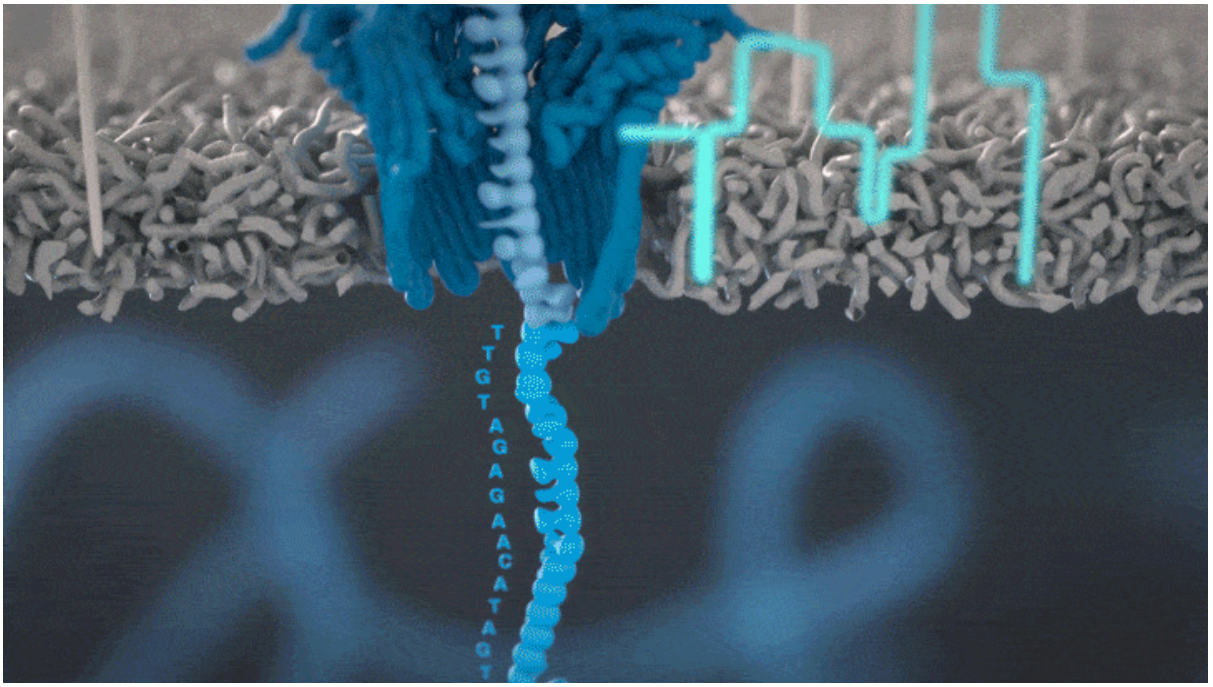
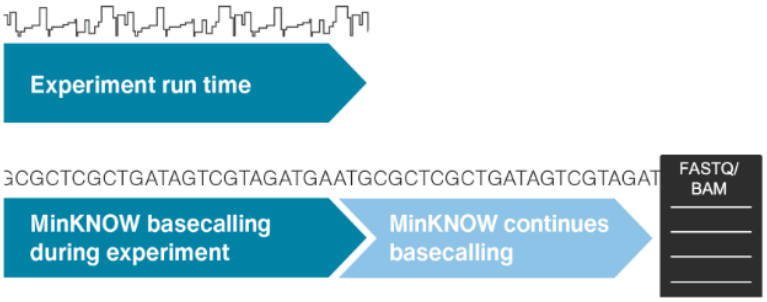
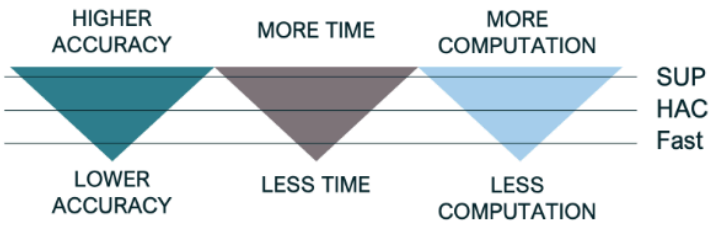
The input data are reads that have already been processed by a basecaller.

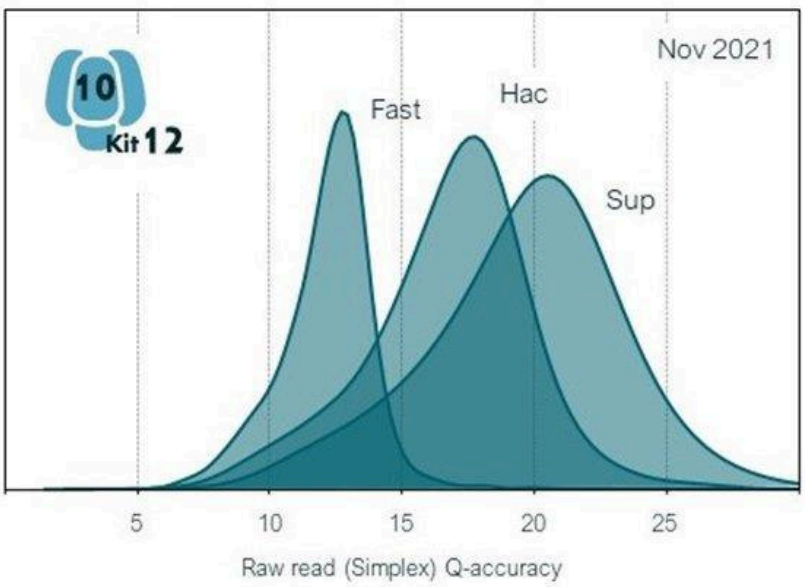
When sequencing DNA or RNA with nanopores, the changes in current caused by the DNA or RNA strand as it passes through the pore are recorded by the MinKNOW™ software running all Oxford Nanopore sequencing devices.
The processive movement of bases through the pore leads to a continuous change in the current, known as a "squiggle". MinKNOW processes the squiggle into real-time reads, each read corresponding to a single DNA/RNA strand.
These reads are written to POD5 files. This raw data contains information not only about canonical bases, but also about base modifications, such as methylation. (⛓)
These raw data obtained during sequencing are processed and converted to FASTQ format (⛓) which is one of the most widely used formats in bioinformatics and genomics.



Basecalling can be done, according to the user's needs, in 3 main modes: fast (faster and less computationally intensive), HAC (High accuracy basecalling, high accuracy, intermediate speed and computational requirements) or SUP (Super accuracy basecalling, more accurate and computationally intensive). (⛓)
This step can last from a few hours to several days, the choice is not trivial and is established during the design of the experiment. It has the advantage that the raw files (POD5) can be recalled in fast mode for a first approach and then recalled again in HAC or SUP to improve the quality of each base and produce better results in the subsequent steps.





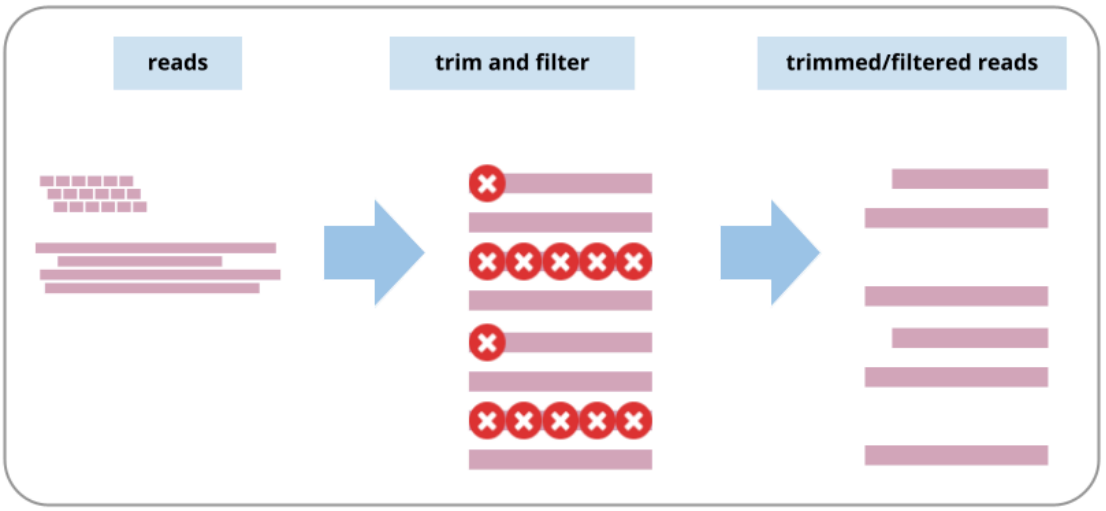


Barcode search:

At the time of basecalling, the same program (formerly Guppy, or more recently Dorado (⛓) eliminates adaptor sequences at the beginning of the reads.

In case you want to check this point you can use third-party programs such as Porechop (⛓) or Porechop_ABI (⛓).

This search for adapters at the ends of the reads, as well as internal adapters for cases where some reads are chimeric (union of 2 reads that passed through the same pore and were read as one).



Mitogenomic assembly

The above points will be carried out by the teachers in charge of the course, so the students will be provided with fastq data previously obtained and without adapters during the class.

Quality control:

Please check your path with:

```
pwd
```

Expected output:

```
/home/manager/Mitogenomic_uy_Wellcome/
```

If you are in other folder move to the correct path with:

```
cd /home/manager/Mitogenomic_uy_Wellcome/
```

FastQC

FastQC is the standard tool for NGS quality control. This tool allows corroboration of length, number of sequences and sequencing quality per base (estimated analysis time ~2 minutes):

```
fastqc -t 4 minion.fq.gz
```

-t = number of cores

Expected output:

```
Application/gzip
Started analysis of minion.fq.gz
Approx 5% complete for minion.fq.gz
Approx 10% complete for minion.fq.gz
Approx 15% complete for minion.fq.gz
Approx 20% complete for minion.fq.gz
......
Analysis complete for minion.fq.gz
```

Open the analysis result:

```
firefox minion_fastqc.html
```

Analyze fastQC output

Questions:

1. How many total reads did the sequencing by Nanopore have?
2. What was the average length of the reads?

3. What was the range of quality?
4. Are there any overrepresented reads?

NanoPlot

NanoPlot is the specialized tool for quality control of long sequences, especially developed for Nanopore Technologies.

(estimated analysis time ~4 minutes)

```
NanoPlot -t 4 --fastq /home/manager/Mitogenomic_uy_Wellcome/minion.fq.gz
--dpi 300 --N50 -o ./nanoplot --huge
```

-t = number of cores

-- fastq = fastq data

--dpi = dots per inch

--N50 = insert a line in the graphs showing the metric N50

The output file is in the "nanoplot" folder and the file is called "NanoPlot-report.html" which can be opened in a browser such as Chrome or Firefox.

```
firefox ./nanoplot/NanoPlot-report.html
```

Analyze NanoPlot output

Questions:

1. What additional information does NanoPlot versus fastQC offer?
2. Which of the NanoPlot analyses are more informative?

Diamond

We will use mitochondrial proteins from previously characterized species to select only mitochondrial reads obtained in by nanopore sequencing (since the starting DNA materia contains mixed nuclear DNA and mitochondrial DNA), this is the core strategy of Genome skimming (i.e. separate mitochondrial reads from nuclear reads.)

The cladoi.dmnd file (database for further analysis) will be provided to you in the folder "`/home/manager/Mitogenomic_uy_Wellcome/`".

- makedb = command that creates the database for the next steps.
- --threads = number of CPUs to use
- --db = name of the database
- --in = input file with the proteins of interest.

Diamond **blastx** (estimated running time: ~35-40 minutes)

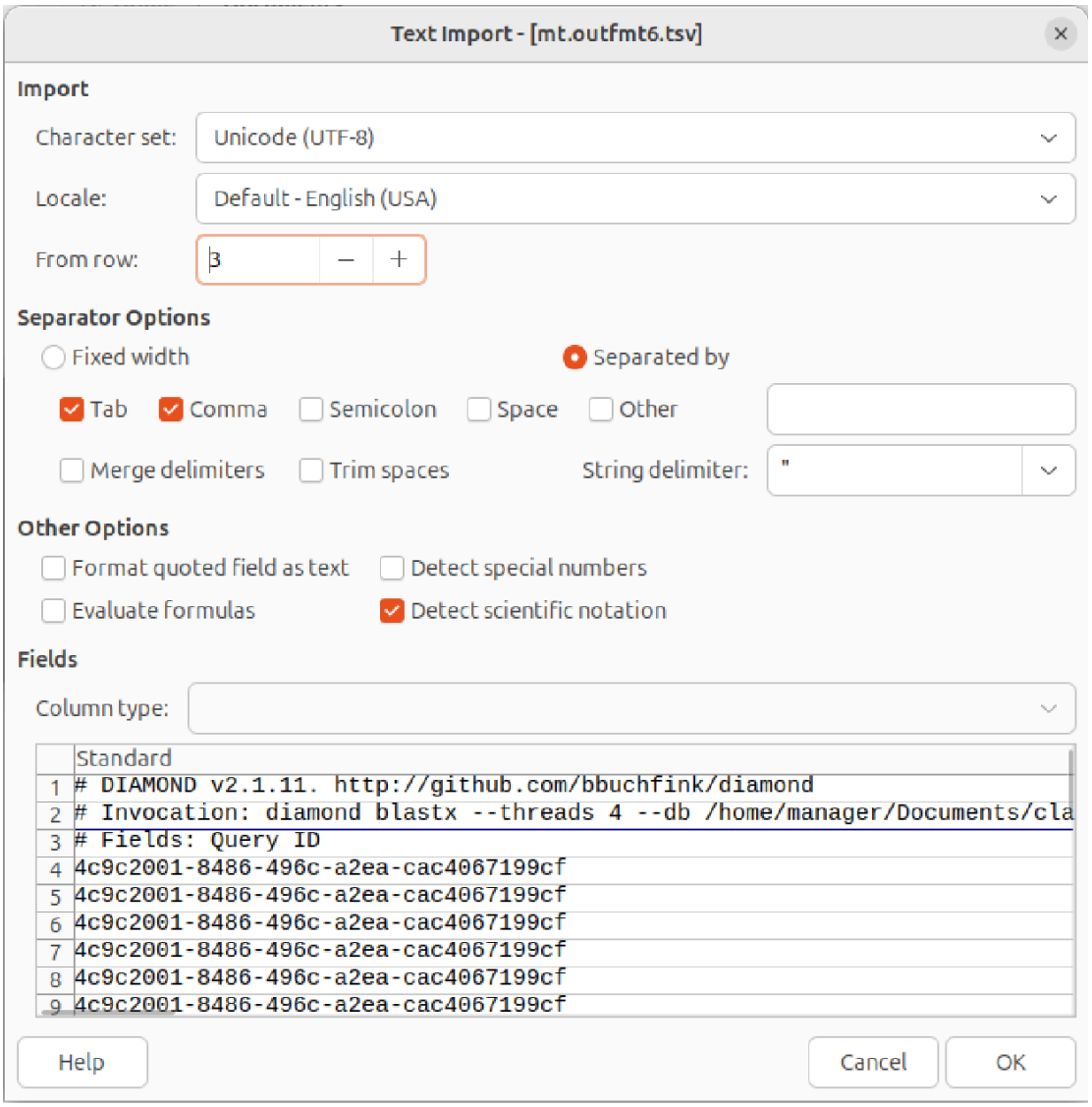
```
diamond blastx --threads 4 --db
/home/manager/Mitogenomic_uy_Wellcome/cladoi.dmnd --out mt.outfmt6.tsv
--outfmt 6 --query-gencode 5 --header --ultra-sensitive --max-hsps 5
--unal 0 --alfmt fastq --al minion_drenale.fq --query
/home/manager/Mitogenomic_uy_Wellcome/minion.fq.gz
```

- blastx = nucleic acid vs. protein DB data search command
- --threads = number of CPUs to use
- --db = name of the database created in the previous step
- --out = table with the results of reads that match against DB proteins
- --outfmt 6 = format of the table with the results, in this case NCBI format 6 (tabular) (⛓)
- --query-gencode = by default is the use of universal codons, in this case (5) indicates invertebrates mitochondrial gencode. (⛓)
- --header = the output table has the headers that identify each column.
- --ultra-sensitive = Activate the very sensitive mode designed to obtain the best sensitivity including the range of zones with <40% identity (optimized for our case without phylogenetically close species). (⛓)
- --max-hsps = The maximum number of HSP (*High-Scoring Segment Pairs*) by target sequence to be reported for each query.
- --unal 0 = reads that do not match the protein DB are not reported.
- --alfmt = output format of the file containing reads (*fasta o fastq*)
- --al = name of the file with matching reads.
- --query = input file, the sequenced reads.

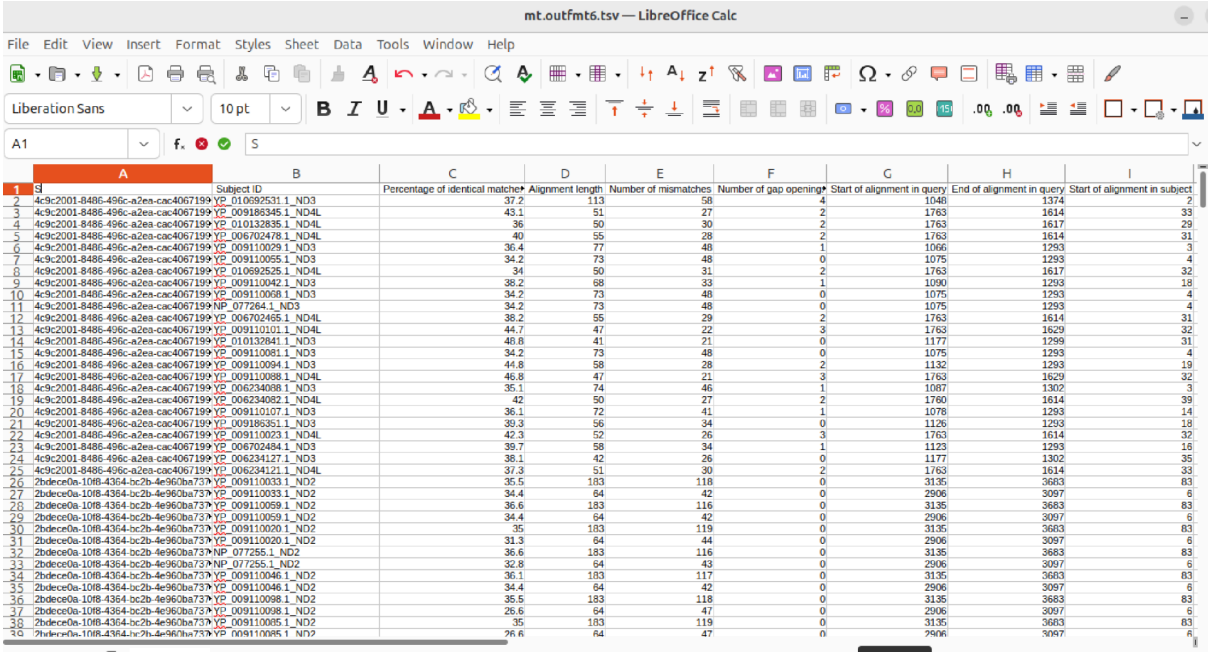
Analyze the results of diamond output:

Go to the File Explorer📁 and Double click to open the file mt.outfmt6.tsv.

Select the option to import following this image:



<u>Expected output:</u>



mt.outfmt6.tsv — LibreOffice Calc

	A	B	C	D	E	F	G	H	I
1	S	Subject ID	Percentage of identical matche	Alignment length	Number of mismatches	Number of gap opening	Start of alignment in query	End of alignment in query	Start of alignment in subject
2	4c9c2001-8486-496c-a2ea-cac4067199	YP_010692531.1_ND3	37.2	113	58	4	1048	1374	2
3	4c9c2001-8486-496c-a2ea-cac4067199	YP_009186345.1_ND4L	43.1	51	27	2	1763	1614	33
4	4c9c2001-8486-496c-a2ea-cac4067199	YP_010132835.1_ND4L	36	50	30	2	1763	1617	29
5	4c9c2001-8486-496c-a2ea-cac4067199	YP_006702478.1_ND4L	40	55	28	2	1763	1614	31
6	4c9c2001-8486-496c-a2ea-cac4067199	YP_009110029.1_ND3	36.4	77	48	1	1066	1293	3
7	4c9c2001-8486-496c-a2ea-cac4067199	YP_009110055.1_ND3	34.2	73	48	0	1075	1293	4
8	4c9c2001-8486-496c-a2ea-cac4067199	YP_010692525.1_ND4L	34	50	31	2	1763	1617	32
9	4c9c2001-8486-496c-a2ea-cac4067199	YP_009110042.1_ND3	38.2	68	33	1	1090	1293	18
10	4c9c2001-8486-496c-a2ea-cac4067199	YP_009110068.1_ND3	34.2	73	48	0	1075	1293	4
11	4c9c2001-8486-496c-a2ea-cac4067199	NP_077264.1_ND3	34.2	73	48	0	1075	1293	4
12	4c9c2001-8486-496c-a2ea-cac4067199	YP_006702465.1_ND4L	38.2	55	29	2	1763	1614	31
13	4c9c2001-8486-496c-a2ea-cac4067199	YP_009110101.1_ND4L	44.7	47	22	3	1763	1629	32
14	4c9c2001-8486-496c-a2ea-cac4067199	YP_010132841.1_ND3	48.8	41	21	0	1177	1299	31
15	4c9c2001-8486-496c-a2ea-cac4067199	YP_009110081.1_ND3	34.2	73	48	0	1075	1293	4
16	4c9c2001-8486-496c-a2ea-cac4067199	YP_009110094.1_ND3	44.8	58	28	2	1132	1293	19
17	4c9c2001-8486-496c-a2ea-cac4067199	YP_009110088.1_ND4L	46.8	47	21	3	1763	1629	32
18	4c9c2001-8486-496c-a2ea-cac4067199	YP_006234088.1_ND3	35.1	74	46	1	1087	1302	3
19	4c9c2001-8486-496c-a2ea-cac4067199	YP_006234082.1_ND4L	42	50	27	2	1760	1614	39
20	4c9c2001-8486-496c-a2ea-cac4067199	YP_009110107.1_ND3	36.1	72	41	1	1078	1293	14
21	4c9c2001-8486-496c-a2ea-cac4067199	YP_009186351.1_ND3	39.3	56	34	0	1126	1293	18
22	4c9c2001-8486-496c-a2ea-cac4067199	YP_009110023.1_ND4L	42.3	52	26	3	1763	1614	32
23	4c9c2001-8486-496c-a2ea-cac4067199	YP_006702484.1_ND3	39.7	58	34	1	1123	1293	16
24	4c9c2001-8486-496c-a2ea-cac4067199	YP_006234127.1_ND3	38.1	42	26	0	1177	1302	35
25	4c9c2001-8486-496c-a2ea-cac4067199	YP_006234121.1_ND4L	37.3	51	30	2	1763	1614	33
26	2bdece0a-10f8-4364-bc2b-4e960ba737	YP_009110033.1_ND2	35.5	183	118	0	3135	3683	83
27	2bdece0a-10f8-4364-bc2b-4e960ba737	YP_009110033.1_ND2	34.4	64	42	0	2906	3097	6
28	2bdece0a-10f8-4364-bc2b-4e960ba737	YP_009110059.1_ND2	36.6	183	116	0	3135	3683	83
29	2bdece0a-10f8-4364-bc2b-4e960ba737	YP_009110059.1_ND2	34.4	64	42	0	2906	3097	6
30	2bdece0a-10f8-4364-bc2b-4e960ba737	YP_009110020.1_ND2	35	183	119	0	3135	3683	83
31	2bdece0a-10f8-4364-bc2b-4e960ba737	YP_009110020.1_ND2	31.3	64	44	0	2906	3097	6
32	2bdece0a-10f8-4364-bc2b-4e960ba737	NP_077255.1_ND2	36.6	183	116	0	3135	3683	83
33	2bdece0a-10f8-4364-bc2b-4e960ba737	NP_077255.1_ND2	32.8	64	43	0	2906	3097	6
34	2bdece0a-10f8-4364-bc2b-4e960ba737	YP_009110046.1_ND2	36.1	183	117	0	3135	3683	83
35	2bdece0a-10f8-4364-bc2b-4e960ba737	YP_009110046.1_ND2	34.4	64	42	0	2906	3097	6
36	2bdece0a-10f8-4364-bc2b-4e960ba737	YP_009110098.1_ND2	35.5	183	118	0	3135	3683	83
37	2bdece0a-10f8-4364-bc2b-4e960ba737	YP_009110098.1_ND2	26.6	64	47	0	2906	3097	6
38	2bdece0a-10f8-4364-bc2b-4e960ba737	YP_009110085.1_ND2	35	183	119	0	3135	3683	83
39	2bdece0a-10f8-4364-bc2b-4e960ba737	YP_009110085.1_ND2	26.6	64	47	0	2906	3097	6

<u>Questions:</u>

1. For the first read (row 2):
   - Which mitochondrial protein matches with this read?
   - What is the percentage of identity for this march?
   - What is the bit-score and e-value?

Rechecking of reads filtered by diamond using Nanoplot:

```
NanoPlot -t 4 --fastq minion_drenale.fq --dpi 300 --N50 -o
./nanoplot_mt_drenale --huge
```

Open the Nanoplot output:

```
firefox ./nanoplot_mt_drenale/NanoPlot-report.html
```

Compare diamond's nanoplot output with previous nanoplot output

<u>Questions:</u>

1. How many reads were left in this analysis?
2. What percentage of the total reads does it represent?
3. Is it an expected percentage of reads?
4. What range of read sizes were left in this analysis?
5. Is it an expected range of sizes?
6. What is the average quality of the reads?
7. Is it different from the quality of previous reads? Why?

Note: If you want to make customized DB with a particular set of proteins you could use this commands:

```
diamond makedb --threads 4 --db customized_DB.dmnd --in
particular_set_of_proteins.fa
```

The multifasta file must be as follows:

To make Diamond DB is recommended that the ID from each protein of the mitochondrial reference sequences have this format (ID_COX1, ID_COX2…):

```
>YP_913152.1_COX2 (mitochondrion) [Romanomermis culicivorax]
MSNFMGLNLMDQMNFLNWKIYLYNDVVIFIESIIAFMVFSFMISMSLNKSWTQSMGHWFA
LELIWTISPVLILLFLGLPSLKMLYFSEIYNFSSYLSLKVMGHQWYWEYSFPEFNTNILS
FPKVLSELIRFGESILLVLPFNFKIRAIISSSDVIHSWALPSMSFKMDAIPGRLNFYMMM
FMMPGKFIGQCSELCGTYHSWMPIYIETTSISLFFEWMKSI
>YP_913153.1_ND4L (mitochondrion) [Romanomermis culicivorax]
MLEMNFIFFMLIFCMILLILNYKMLVFFLIIIELISLTLIIYLMYFINFYFIFLTLFVQV
FESVILILLTFDNFSNSNMIENLMKINY
>YP_913154.1_ND6 (mitochondrion) [Romanomermis culicivorax]
MVGLLGFWIFLLKSFYSSWMLLIFLIIMISGVFLMLFYLSLLMSKLFKLKKKSLLFIFLL
MFPNFFFFKNYYFSELSLNLMDLQLNSMKLILFSLFVFLLSLMIVNNLSLKSKYYRQMKF
LKSEI
>YP_913155.1_ND2 (mitochondrion) [Romanomermis culicivorax]
MMYMIMLLISMNLFSWWWFWMMLEILNWFLITWMKKKVIKLLFLLWQSLSSLLLLFYLLI
NLNFFFYFFFFFMKMSLPPFQQMFWKLHIYLNWKIFIIFMTLHKFLPMMFMTMFFMKNFM
NIIIFFPLIIFYMFWNKMNLISNLFMFLMSDSFWMIIAFFLSLKMAIVYMLVTTMMFIIF
WSYKNQNKENISNKMNLKFILLLMFSLPPFFTFLIKFNLVFSMMFHFMGFFLMMYLISIF
FYWEIFYLTVINLLMFNLKLNMFMYLFILIHLMFLFLL
```

De novo assembly

Flye

From the filtered readings we will proceed to mitochondrial assembly by Flye (estimated time ~20-30 min):

```
flye --nano-raw minion_drenale.fq -t 4 --meta --keep-haplotypes -o
./flye
```

Analyze the assembly obtained

```
seqkit stats ./flye/assembly.fasta
```

expected output:

```
format  type num_seqs  sum_len  min_len  avg_len  max_len
./flye/assembly.fasta  FASTA   DNA 14   20,127 312  1,437.6 6,924
```

to see the detailed result:

```
cat ./flye/assembly_info.txt
```

expected output:

```
seq_name    length      cov.  circ. repeat      mult. alt_group
graph_path
contig_14   6924  6     N     N     1     *     *,14,*
contig_13   3586  2538  N     N     282   *     *,13,*
contig_4    1380  625   N     Y     1     *     4
contig_12   1322  263   N     Y     1     *     12
contig_11   1241  141   N     Y     1     *     11
contig_9    1106  290   N     Y     1     *     9
contig_10   805   297   N     Y     1     *     10
contig_7    691   185   N     Y     1     *     7
contig_3    668   171   N     Y     1     *     3
contig_1    546   102   N     Y     1     *     1
contig_5    528   211   N     Y     1     *     5
contig_2    513   130   N     Y     1     *     2
contig_8    505   187   N     Y     1     *     8
contig_6    312   146   N     Y     1     *     6
```

It is possible to display the assembly results in GFA format (⛓) graphically with bandage (⛓):

open the folder “`/home/manager/Mitogenomic_uy_Wellcome/`” and in the "bandage" folder start the program.

Within the program go to “FILE > LOAD GRAPH” and look for the file “`./flye/assembly.gfa` “ this file contains the sequence (like the fasta file) and additionally has the connections between the segments that were assembled.

Once loaded, if you press the "Draw graph" button, you will be rewarded with the assembly. If you press "More info" it will give you detailed statistics of the assembly.

<u>Questions:</u>

1. How many contigs were obtained in the de novo assembly?
2. Which one is the longest and what size is it?
3. Which is the shortest and what size is it?
4. Which contig has the largest coverage?
5. Which contig has the lowest coverage?

Annotation of Assembly

Diamond

Now we will use Diamond to detect the assembled contigs containing mitochondrial protein coding sequences (CDS) .

We will change the following parameters:

```
diamond blastx --threads 4 --db
/home/manager/Mitogenomic_uy_Wellcome/cladoi.dmnd --out
mt_drenale_ensamble.outfmt6.tsv --outfmt 6 --query-gencode 5 --header
--unal 0 --alfmt fasta --al mt_drenale_ensamble.fa --query
/home/manager/Mitogenomic_uy_Wellcome/flye/assembly.fasta
```

With these results, the segment containing the COX1 gene will be manually extracted

Go to the File Explorer📁 and Double click in to open the file mt_drenale_ensamble.outfmt6.tsv

<u>Expected output:</u>

A	B	C	D	E	F	G	H	I
# Fields: Query	Subject ID	Percentage of identical matc	Alignment len	Number of mismatc	Number of gap open	Start of alignment in qu	End of alignment in qu	Start of alignment in sub
contig_9	YP_009186339.1_COX1	71.4	192	55	0	3	578	157
contig_9	YP_009110082.1_COX1	70.8	192	56	0	3	578	158
contig_9	YP_010132842.1_COX1	72.4	192	53	0	3	578	158
contig_9	YP_006702459.1_COX1	70.8	192	56	0	3	578	157

<u>Questions:</u>

- Which Is the **best match** for COX1 protein?

You can help yourself by answering these questions:

- What is the percentage of identity for this match?
- What is the bit-score and e-value?
- What is the <u>name of the</u> **selected contig?**

Note: In this example the name of the contig is “contig_9”. For your data the name could be different.

Based <u>on the name of</u> **selected contig**, continue with:

From the file “mt_drenale_ensamble.fa”, copy and paste the **selected contig** to a new file called "cox1.fasta"

MitoZ

MitoZ consists of independent modules for annotation of de novo assemblies. Automatically find mitochondrial contigs, annotation and visualization.

Due to time constraints we will only analyze the contig that we know codes for the COX1 protein since we pre-selected it with Diamond:

```
conda activate mitozEnv

mitoz annotate --workdir ./ --fastafiles cox1.fasta --clade Nematoda
--genetic_code 5 --outprefix mitoz_drenale
```

Open the summary.txt file:

```
 open ./mitoz_drenale.cox1.fa.result/summary.txt
```

<u>Expected output (first lines):</u>

```
#Seq_id        Length(bp)     Circularity    Closely_related_species
contig_9       579            no             Trichinella nelsoni


#Seq_id        Start  End    Length(bp) Direction  Type   Gene_name  Gene_prodcut                    Total_freq_occurred
-----------------------------------------------------------------------------------------------------------------------------
contig_9       <3     580    578        +          CDS    COX1       cytochrome c oxidase subunit I    1


Protein coding genes totally found:  1
tRNA genes totally found:            0
rRNA genes totally found:            0
-------------------------------------
Genes totally found:                 1


Potential missing genes:
#Gene             total_missing_number
-------------------------------------
ATP6                                 1
ATP8                                 1
COX2                                 1
COX3                                 1
```

<u>Questions:</u>

-How many mitochondrial proteins were found in the selected contig?

- Is the same protein as Diamond previously detected?

Additional activities

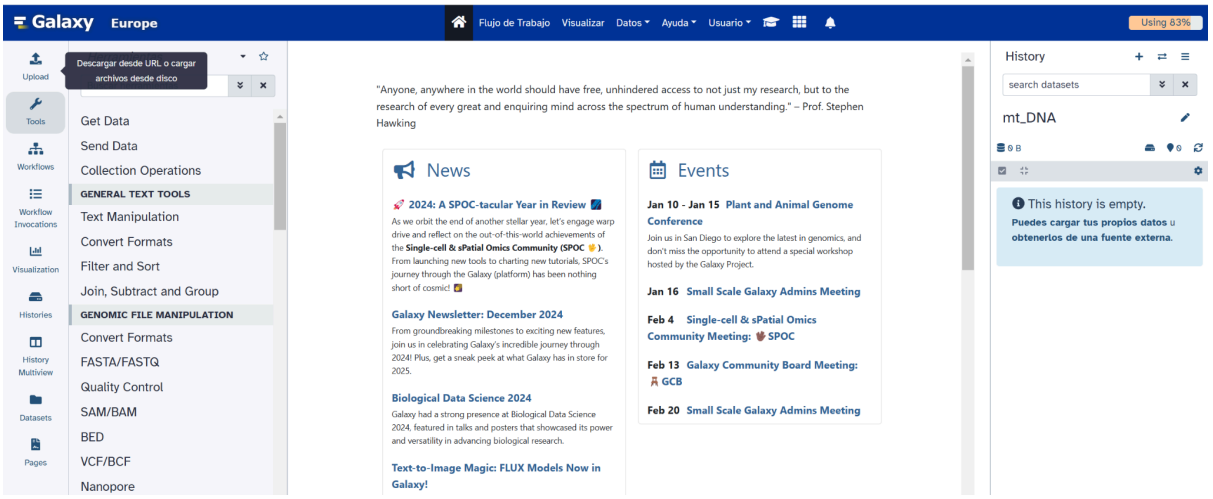
Optional steps:

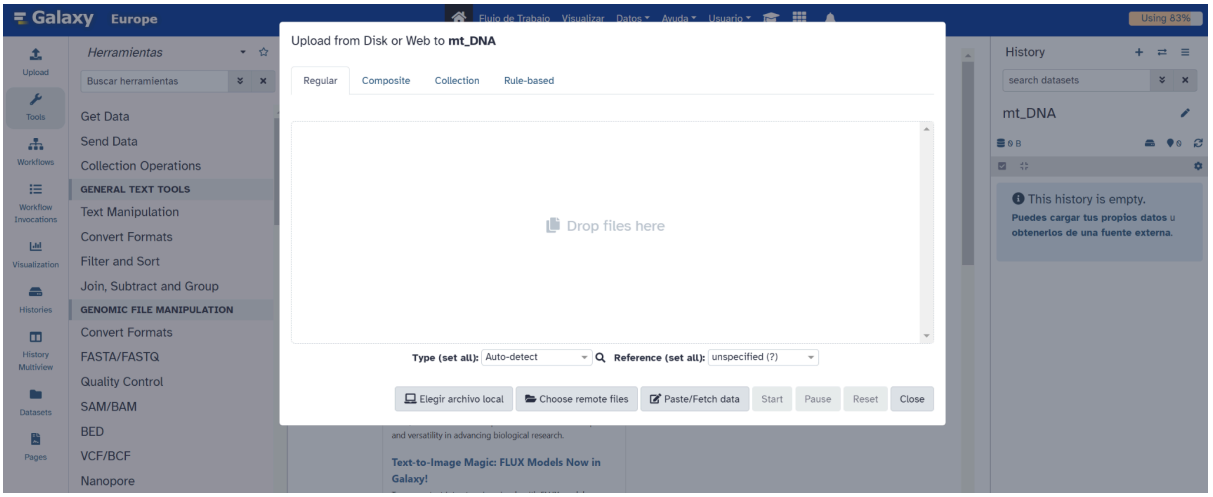
It is recommended to check for contaminants in any sequencing protocol.

Kraken2 is an easy and fast tool to check common species that could contaminate your samples.

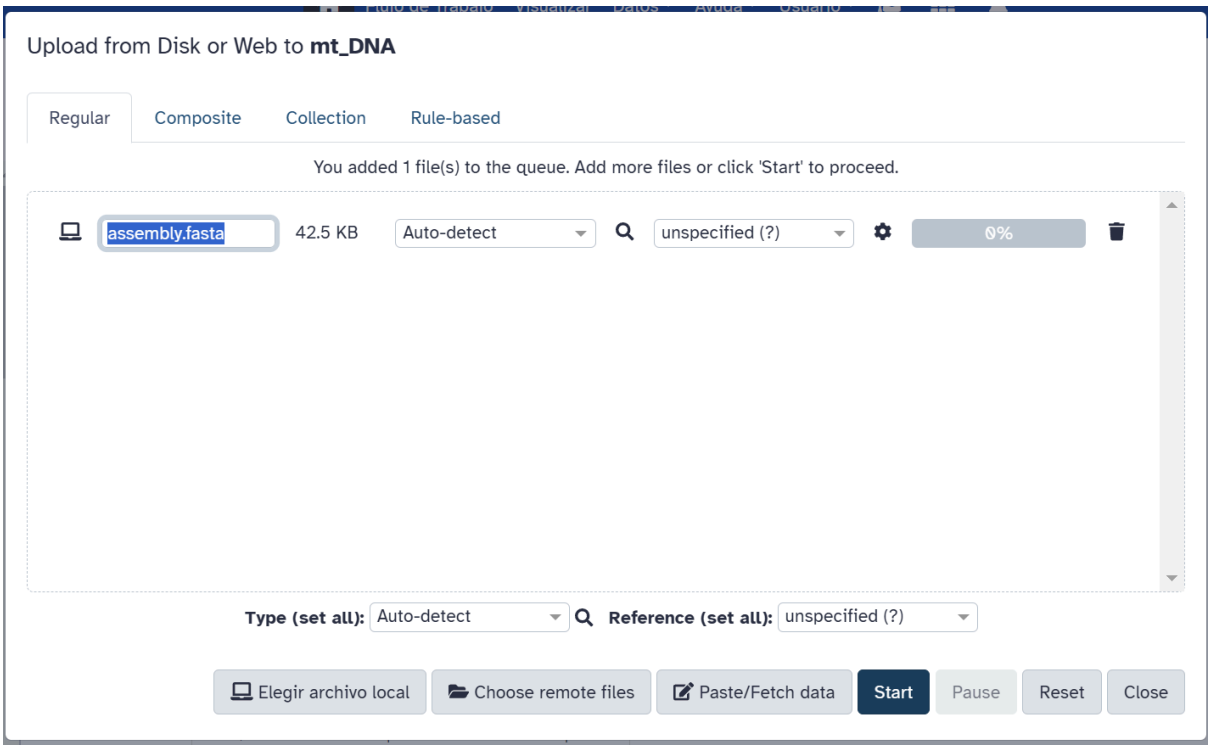
Kraken2

They will log in to http://usegalaxy.eu and upload the assembly file in fasta format:





Select the file and click on the "start" option:



Once uploaded, we will select Kraken2 (⛓) a specialized tool for quality control and detection of foreign genetic material in assemblies or raw data.

TIP: A filtering step could be added at the beginning of this guide following these steps shown below, but remember that it is a computationally expensive process for raw reads and the duration of its execution for a complete genomic dataset (5-10 gigabytes) can be around a day of execution.

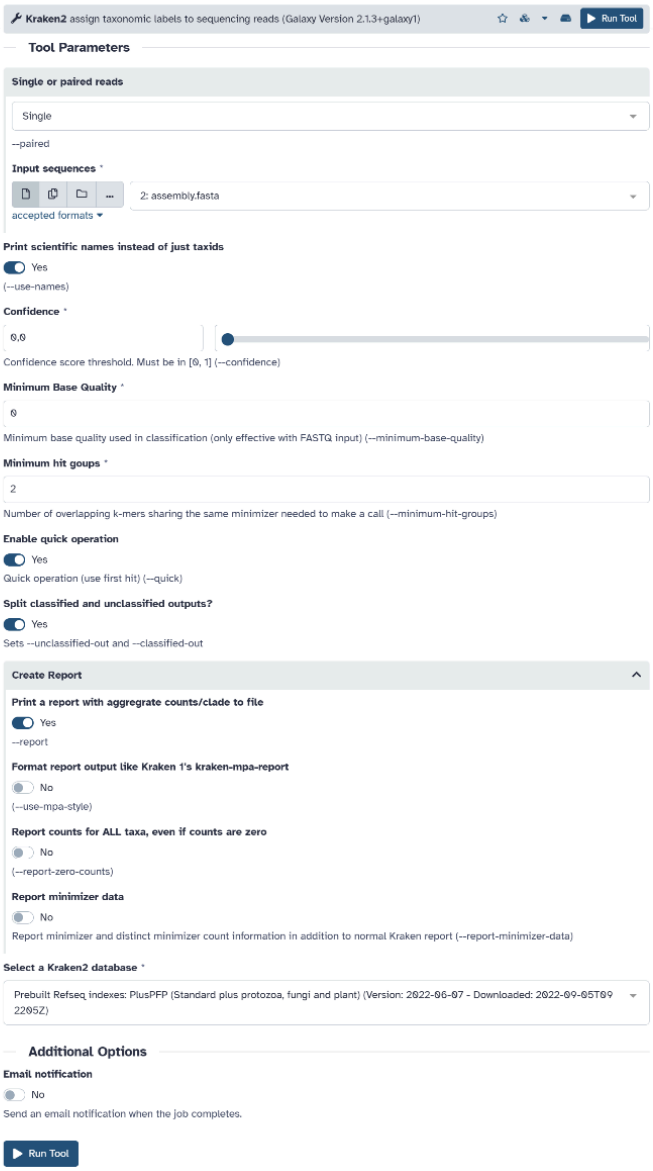
Kraken2

Select the following parameters:

- **Input sequences** = the assembly
- **Print scientific names instead of just taxids** = YES
- **Confidence** = you can use the default (0) or raise it up to 0.8
- **Enable quick operation** = Yes (is recommended for this practice, in actual use it is not suggested).
- **Split classified and unclassified outputs?** = Yes (separates the fasta into 2, those that match the database (matched) and those segments/reads that don't (unmatched)

En la sección **Create report**:

- **Print a report with aggregate counts/clade to file** = Yes. Returns a table with the names of the genera or species on which the program is able to match.
- **Select a Kraken2 database** = Prebuilt Refseq indexes: PlusPFP (standard plus protozoa, fungi and plant)



Alternative annotation.

Another useful tool to annotate mitogenomes is MITOS2 annotation tool as we show in the next step-by-step galaxy guide:

MITOS2

Proceed to upload the file "cox1.fasta" to galaxy as we did before.
Once uploaded, select "mitos2".

Parameters:

- **Sequence** = cox1.fasta
- **Genetic code** = Invertebrate (5)
- **Reference data** = Refseq89 Metazoa
- **Treat sequence as linear** = yes (in case of obtaining a single circular mitochondrial genome, use the "NO" option).
- **Outputs** = select all items.

Sequence *

3: cox1.fasta

accepted formats

A single sequence in fasta formated sequence (--input)

Genetic code *

Invertebrate (5)

(--code)

Reference data *

RefSeq89 Metazoa

Contact the administrator of this Galaxy instance if you miss reference data (--refseqver)

Treat sequence as linear

Yes

(--linear)

Outputs - optional

BED × mito × GFF file × SEQ × nucleotide FASTA × protein FASTA × geneorder × Protein prediction plot × ncRNA prediction plot × ncRNA structure plots - svg × Missing genes ×

switch to column select

Advanced options

Feature types - optional

Protein coding genes × tRNAs × rRNAs ×

switch to column select

Feature types that should be predicted by MITOS (--noprot,--notrna,--norrna)

Final overlap (nt) *

50

Maximum number of nucleotides by which genes of different types may overlap (--finovl)

Annotate only the best copy of each feature

No

(--best)

Fragment overlap *

20

Maximum allowed overlap of proteins in the query (in percent of the shorter query range) for two hits to be counted as fragments of the same gene (--fragovl)

Fragment quality factor *

10,0

Maximum factor by which fragments of the same protein may differ in their quality (--fragfac)

Advanced options for protein coding gene prediction

Advanced options for ncRNA gene prediction

Additional Options

Email notification

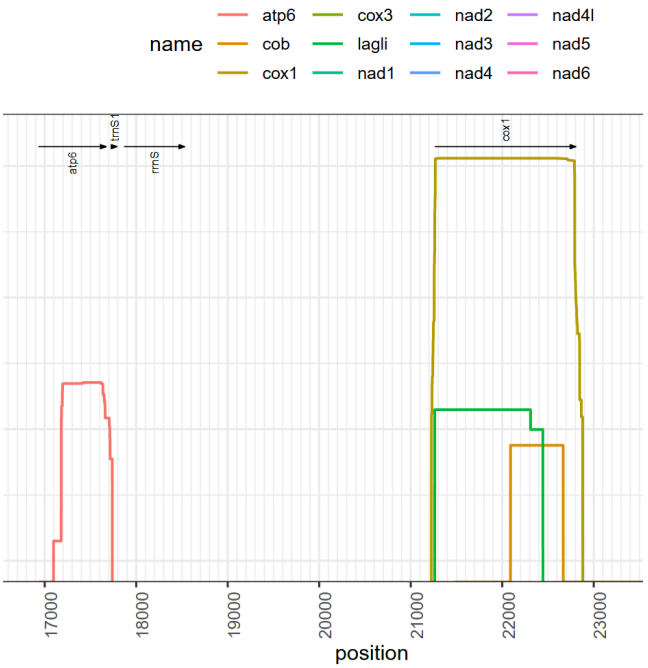
No

Send an email notification when the job completes.

Run Tool

Questions:

1. Within the results you will find the "Protein prediction plot" graph, describe the observed results. What does the multi-colored stacked histogram graph mean (see example image)?



2. Within the "missing genes" file the undetected genes are shown, which are they? are they protein coding or non-coding?
3. The missing genes file also shows "*duplicated genes*", which ones are reported in this file?
4. Check within the MITOS file (in BED format) if the score of these duplicated genes is the same. Compare the results obtained in this file with those shown in the "ncRNA prediction plot" and "protein prediction plot" files.
5. Based on your biological criteria and the knowledge you have learned during the course, would you choose one of the genes over the other? would you eliminate any of them?
6. Insert the images of the three-dimensional structures of a tRNA and an rRNA.