Step 1: verificare lo stato della connessione

```
ladybug@ladybug:~$ sudo apt update
```

Per chi è in root@name_pc ed è entrato nel suo account studente@nome_pc potreste avere problemi del tipo:

```
studente1@ladybug:~$ sudo apt update
[sudo] password di studente1:
studente1 is not in the sudoers file.
This incident has been reported to the administrator.
studente1@ladybug:~$
```

SOLUZIONE

Step 0: dare permessi sudo al tuo account

DALL'ACCOUNT ROOT:

```
root@ladybug:~# usermod -aG sudo studente1
```

FATTO?!

Bene, allora possiamo iniziare...

COSA FAREMO OGGI

- Installare Conda e creare nuovi ambienti
- Scaricare reads Illumina e ONT
- Valutare qualità delle reads
- Sfruttare reads ONT per fare un assemblaggio
- Valutare l'assemblaggio
- (Polishing assemblaggio)

MA PRIMA...

Controlliamo che abbiate tutti i programmi necessari!

```
studente1@ladybug:~$ sudo apt -y install curl bzip2 wget
```

> brew install wget curl

CONDA

curl -fL -o Miniconda.sh https://repo.anaconda.com/miniconda/Miniconda3-latest-MacOSX-arm64.sh

...o se avete Intel Macs (x86_64)

curl -fL -o Miniconda.sh https://repo.anaconda.com/miniconda/Miniconda3-latest-MacOSX-x86_64.sh

curl -fL -o Miniconda.sh https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh

Quello che abbiamo fatto è stato scaricare dal web con il comando “curl” che sta per “client URL” uno script (evidenziato in giallo) che permette l’installazione di Miniconda. Abbiamo rinominato lo script (che in questo caso è l’output di curl) usando la flag “-o” in “**Miniconda.sh**”

Ed ora eseguendo lo script partirà l’installazione di miniconda...

CONDA

Eseguendo questo comando:

zsh Miniconda.sh -b

Eseguendo questo comando:

bash Miniconda.sh -b

I. Dovrebbe iniziare l'installazione, vi chiederà inanzitutto di visualizzare i termini di servizio, premete ENTER.
II. Successivamente vi stamperà a schermo lo User License Agreement, premete "q".
III. A questo punto vi chiederà di accettare i termini di servizio, scrivete "yes" e premete ENTER.
IV. Infine vi chiederà se vi va bene la cartella in cui siete come luogo dell'installazione. Premete ENTER.
V. Ad installazione completata vi chiederà se volete "far partire conda all'avvio". Scrivete "yes" e premete ENTER.

Thank you for installing Miniconda3!

Una volta riavviata la vostra shell dovrebbe apparirvi un "(base)" di fianco al vostro nome.

```
(base) studente1@ladybug:~$
```

CONDA

Conda serve a tenere separati gli ambienti di lavoro per evitare che conflitti tra versioni o dipendenze “contaminino” gli altri progetti.

```
(base) studente1@ladybug:~$
```

Attualmente vi trovate nell’ambiente chiamato “base”

Ora configureremo dei canali (conda-forge e bioconda) che useremo per scaricare i nostri programmi:

```
(base) studente1@ladybug:~$ conda config --add channels conda-forge
(base) studente1@ladybug:~$ conda config --add channels bioconda
(base) studente1@ladybug:~$
```

Ed ora creiamo il nostro primo ambiente Conda!

-ad un certo punto vi chiederà di accettare altri termini di servizio, premete ENTER

Per creare un nuovo ambiente:

```
(base) studente1@ladybug:~$ conda create -n Quality_control
```

Vi chiederà se siete sicuri di volerlo creare, basta premere ENTER.

Per attivare l'ambiente creato:

```
(base) studente1@ladybug:~$ conda activate Quality_control
(Quality_control) studente1@ladybug:~$
```

È case-sensitive (attenzione a maiuscole e minuscole)

Per elencare tutti gli ambienti che abbiamo creato:

```
(Quality_control) studente1@ladybug:~$ conda env list

# conda environments:
#
base                  /home/studente1/miniconda3
Quality_control     *  /home/studente1/miniconda3/envs/Quality_control
```

"*****" ci indica in che ambiente ci troviamo

Per installare un programma o pacchetto:

```
(Quality_control) studente1@ladybug:~$ conda install fastqc
```

Vi chiederà se siete sicuri di volerlo installare, basta premere ENTER.

Per disattivare un ambiente:

```
(Quality_control) studente1@ladybug:~$ conda deactivate
(base) studente1@ladybug:~$
```

Vogliamo creare due ambienti:

Quality_control

- fastqc
- nanoplot

Assembly

- flye
- bandage

Possibili problemi di compatibilità con Bandage

Assembly
- flye
- bandage

Questo sarebbe l'ambiente Conda che vorremmo creare, ma l'ultima versione di Flye (v.2.9) potrebbe andare in conflitto con Bandage, se avete problemi create un terzo ambiente chiamato Bandage ed installate dentro solo Bandage.

conda deactivate → conda create -n Bandage → conda activate Bandage → conda install bandage

Se avete avuto problemi con Bandage avrete tre ambienti conda:

Quality_control
- fastqc
- nanoplot

Assembly
- Flye

Bandage
- Bandage

Possibili problemi con installazione programmi in Conda

Nel caso voi stiate provando ad installare un programma e si blocca a “Solving environment” senza mai proseguire c’è un altro modo per installare i programmi: usando un file .yml.

```
(base) studente1@ladybug:~$ conda env create -f Quality_control.yml
```

I file .yml che troverete su moodle sono:

- **Quality_control.yml**: contiene le info per creare un ambiente conda chiamato “Quality_control” con i programmi NanoPlot e FastQC al suo interno.
- **Assembly.yml**: contiene le info per creare un ambiente conda chiamato “Assembly” con il programma Flye al suo interno.
- **Bandage.yml**: contiene le info per creare un ambiente conda chiamato “Bandage” con il programma Bandage al suo interno.

Ora dovreste avrere tutti i programmi installati, iniziamo a analizzare i nostri dati!

Ora che abbiamo i programmi necessari….

...scarichiamo i dati.

Il nostro obiettivo è assemblare il cloroplasto di *Ipomoea batatas.*

FLYE

Assembler for single-molecule sequencing reads, such as those produced by PacBio and Oxford Nanopore Technologies. It is designed for a wide range of datasets, from small bacterial projects to large mammalian-scale assemblied.

It takes raw PacBio/ONT reads as input and outputs polished contigs.

Scarichiamo le reads necessarie.

```
wget https://zenodo.org/record/3567224/files/sweet-potato-chloroplast-nanopore-reduced.fastq

wget https://zenodo.org/record/3567224/files/sweet-potato-chloroplast-illumina-reduced.fastq
```

→ **Ma se Flye usa solo PacBio o ONT, perché stiamo scaricando anche reads Illumina?**

Strategie di sequenziamento e strategie di assemblaggio devono essere pianificate molto in anticipo. Le scelte ricadono sull'uso di long reads e/o short reads principalmente in base ai costi e alle caratteristiche dell'organismo che ci si trova davanti.

Short reads only

Assemblaggio fatto solo con short reads con alto coverage.

Non viene praticamente più usata questa strategia dal 2015

Short reads + long reads

Assemblaggio fatto con short reads con alto coverage + long reads a basso coverage

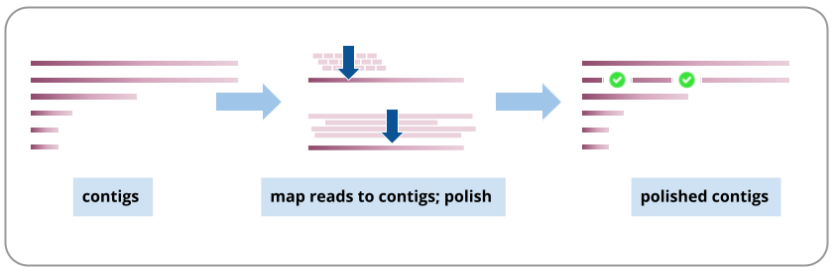
Non viene praticamente più usata perché gli standard attuali richiedono di meglio.

Strategie obsolete MA

In alcune applicazioni si usano ancora: in metagenomica MetaMDBG o per genomi molto ripetitivi e complessi come in alcune piante, si usano ancora approcci ibridi, e assemblatori ibridi come Wengan e HERA.

Long reads high coverage + short reads for polishing

Assemblaggio fatto solo con long reads + short reads per correggere gli errori delle long reads.



L'assemblaggio fatto con long reads provenienti dalla stessa piattaforma permettono di avere dei contigs molto grandi.
Il polishing inoltre permette di avere un assemblaggio più affidabile.

Strategie attuali ancora in uso

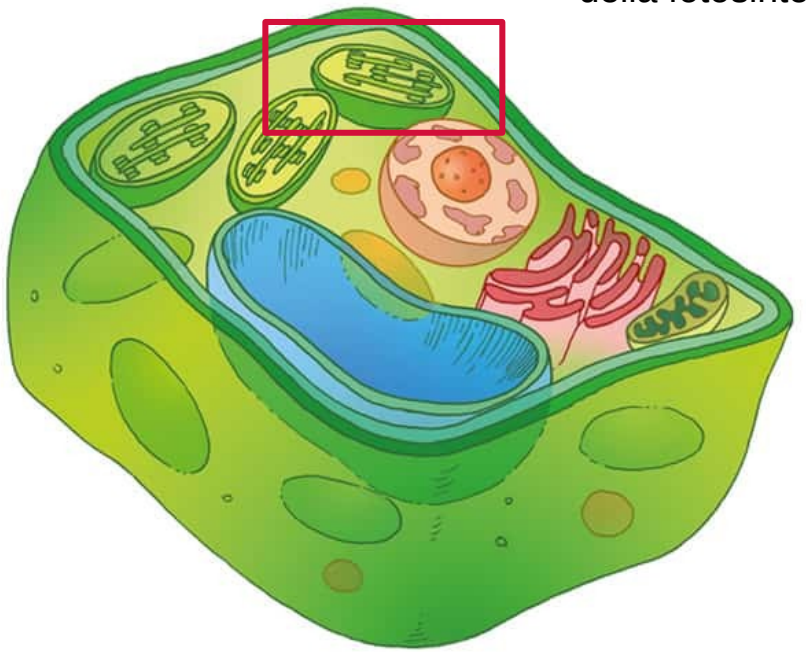
Long reads + Illumina Hi-C for scaffolding

Standard attuali, non serve il polishing, si arriva ad assemblaggi chromosome scale e la differenziazione degli aplotipi

Custom assembly

Combinazione di diversi approcci usato per casi particolari (di solito in organismi con genomi molto complessi).

Il nostro obiettivo è assemblare il **cloroplasto** di *Ipomoea batatas.*



I cloroplasti sono organelli presenti nelle cellule delle piante e delle alghe, responsabili della fotosintesi e derivano da cianobatterio inglobato da una cellula eucariotica.

ENDOSIMBIOSI

Caratteristiche:

- Possiedono un proprio genoma (DNA circolare, simile a quello batterico).

- Piccolo e fortemente ridotto rispetto all'antenato cianobatterico (≈120–170 kb).

- Altamente riarrangiato, ma nel complesso piuttosto stabile tra specie.

- Spesso presenta una regione invertita ripetuta (IR) che contribuisce alla stabilità strutturale.

...perchè un cloroplasto?

È facile da isolare e assemblare rispetto al genoma nucleare. Ottimo modello per esercitarsi con l'assemblaggio di genomi (come con Flye).

QUALITY CONTROL ILLUMINA

Le tecnologie di sequenziamento riescono a generare una quantità di sequenze enorme in un singolo esperimento. Ma visto che nessuna tecnologia è perfetta ed ogni strumento genera diversi tipi di errori è necessario capire, identificare ed escludere gli errori che possono causare interpretazioni errate nelle analisi successive. Il controllo di qualità è quindi uno step essenziale. **Soprattutto in un assemblaggio!**

Iniziamo con il valutare le reads Illumina

Entriamo nell'ambiente conda Quality_control → Creiamo una cartella chiamata Illumina_fastqc → Usiamo fastqc

```
(Quality_control) studente1@ladybug:~$ mkdir Illumina_fastqc
(Quality_control) studente1@ladybug:~$ fastqc sweet-potato-chloroplast-illumina-reduced.fastq -o Illumina_fastq
```

★ Sweet-potato-chloropl….etc è troppo lungo da scrivere? Usa l'autocompletamento con TAB

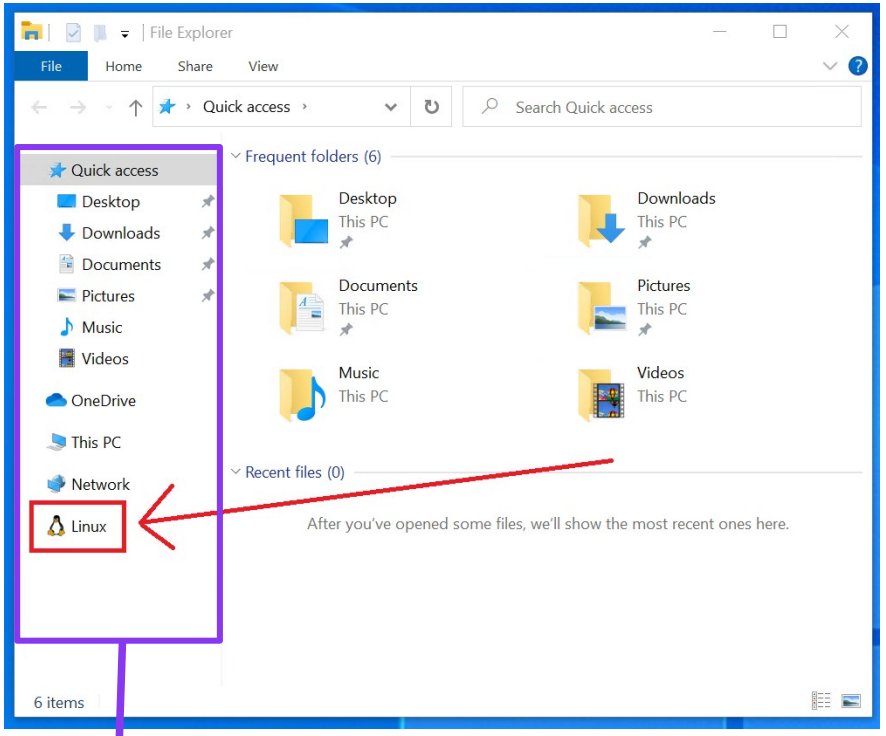
Dentro la cartella che abbiamo creato ci sarà l'output di fastqc:

```
sweet-potato-chloroplast-illumina-reduced_fastqc.html
```

```
sweet-potato-chloroplast-illumina-reduced_fastqc.zip
```

Bello, ha finito, come vediamo l'output?

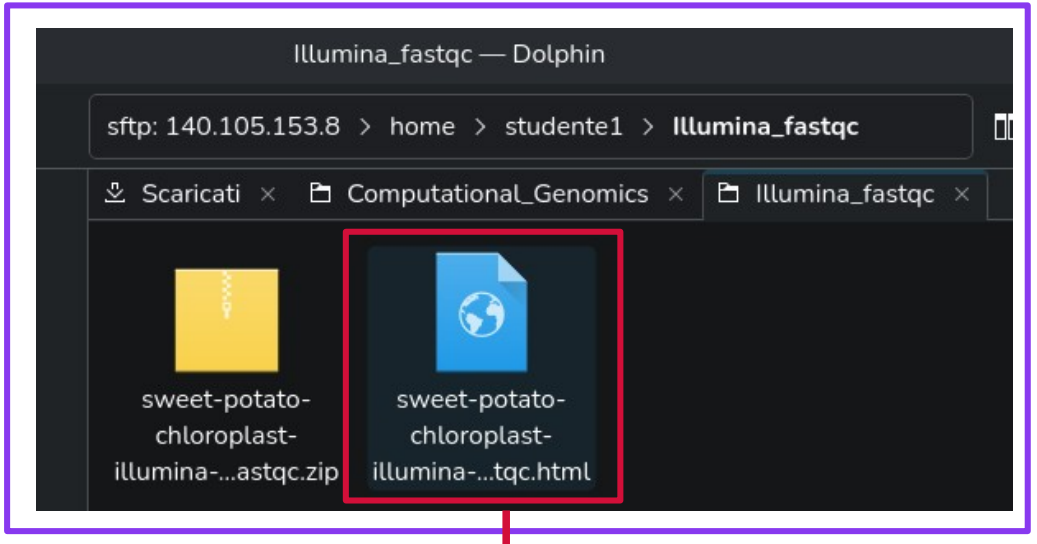
Utenti Windows:



Aprite tutte le cartelle al suo interno
e troverete la cartella Illumina_fastqc

Utenti Apple:

In teoria voi avete un terminale e siete dentro al vostro computer, in una cartella specifica, quindi dovrebbe essere più semplice.



Aprite il file .html. Vi porterà in una pagina web con tutte le informazioni sulle vostre reads. Come valutate le reads Illumina?

FastQC report

Summary

- Basic Statistics
- Per base sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

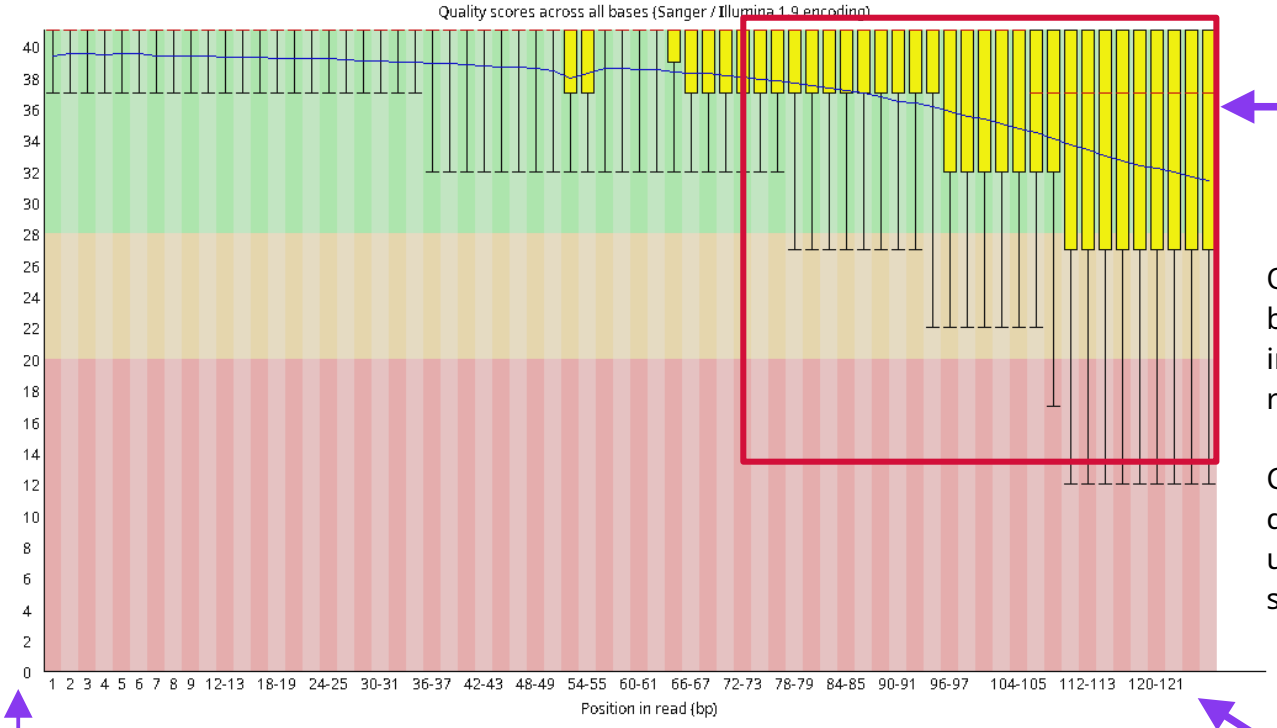
Basic Statistics

Measure	Value
Filename	sweet-potato-chloroplast-illumina-reduced.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	62500
Total Bases	7.8 Mbp
Sequences flagged as poor quality	0
Sequence length	101-127
%GC	38

FastQC report

Con FastQC possiamo usare le informazioni sulla qualità per visualizzarle base per base.

Per base sequence quality



Per ogni posizione ci sta un boxplot:
Median quality value: riga rossa
Inter-quartile (25-75%) range: box giallo
Mean quality value: riga blu

Ovviamente più alta la qualità più si è sicuri che la base call sia corretta, nel background il grafico è diviso in verde (qualità molto alta), arancione (qualità ragionevole) e rosso (qualità molto bassa).

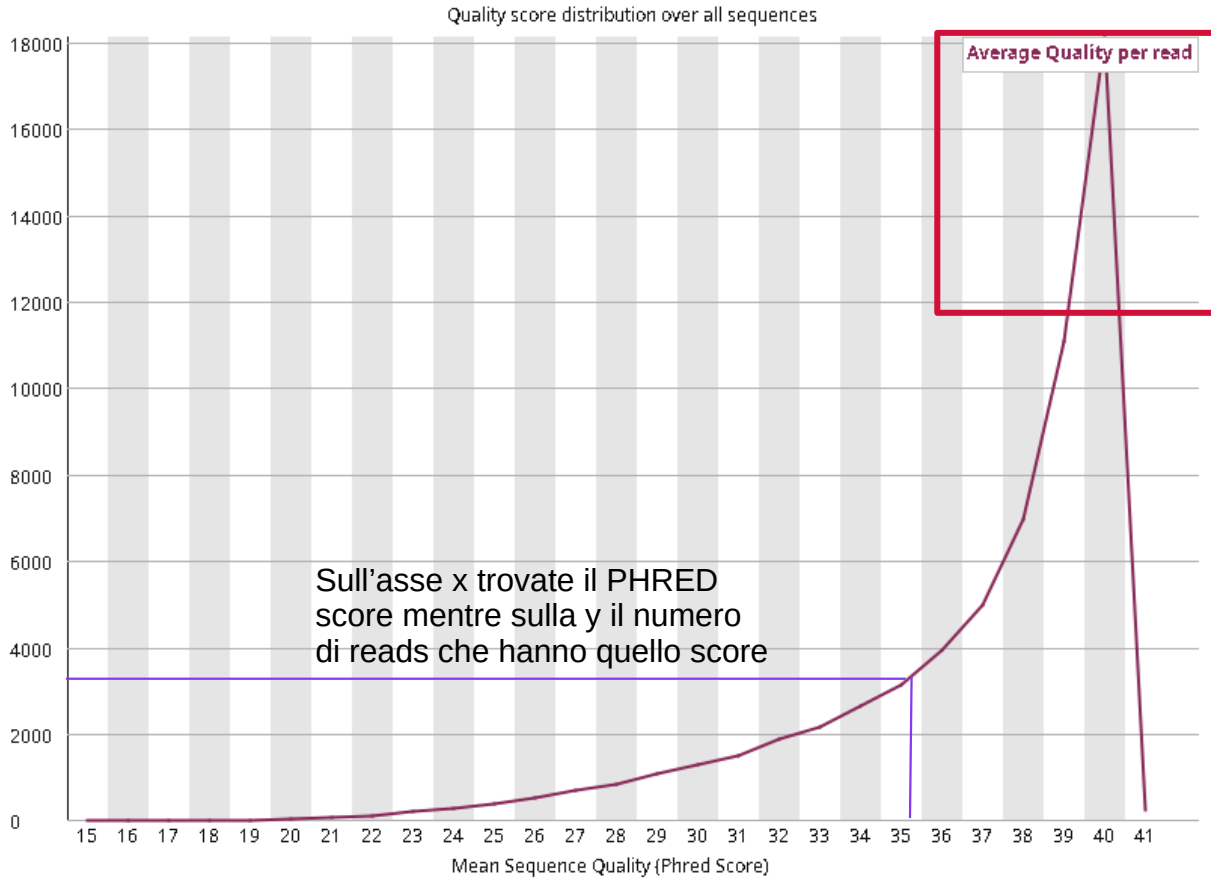
Con reads Illumina è normale che la qualità diminuisca alla fine della sequenza, anche se ultimamente miglioramenti nella chimica del sequenziamento ha migliorato la situazione.

Qualità
(PHRED score)

Sull'asse delle x c'è la posizione dei nt, in questo caso le reads arrivano fino a 121nt.

FastQC report

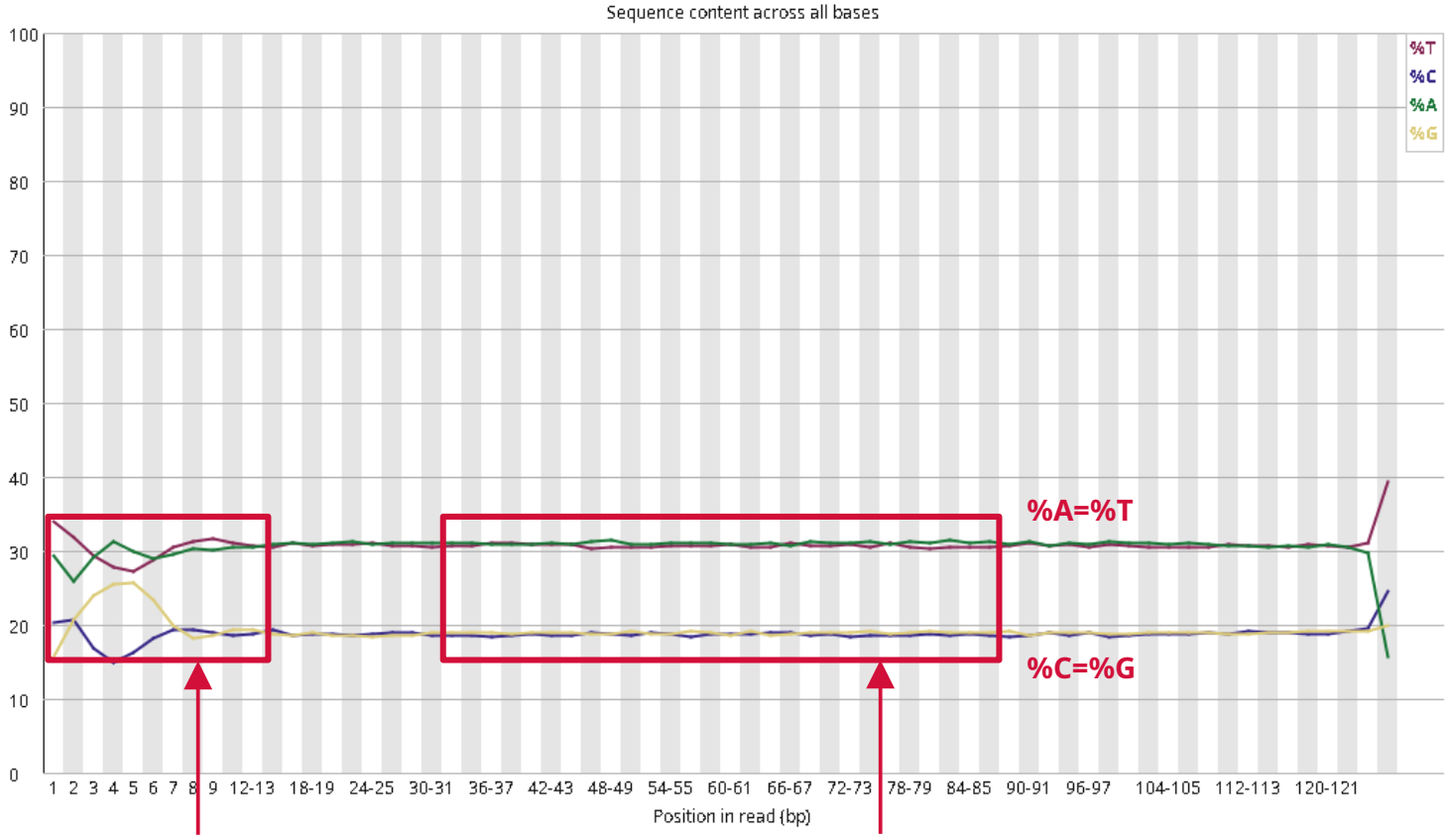
Per sequence quality scores



Per avere una buona distribuzione si deve avere un picco stretto nella parte in alto a destra del grafico (indica che ci sono molte sequenze con una qualità alta).

FastQC report

Per base sequence content

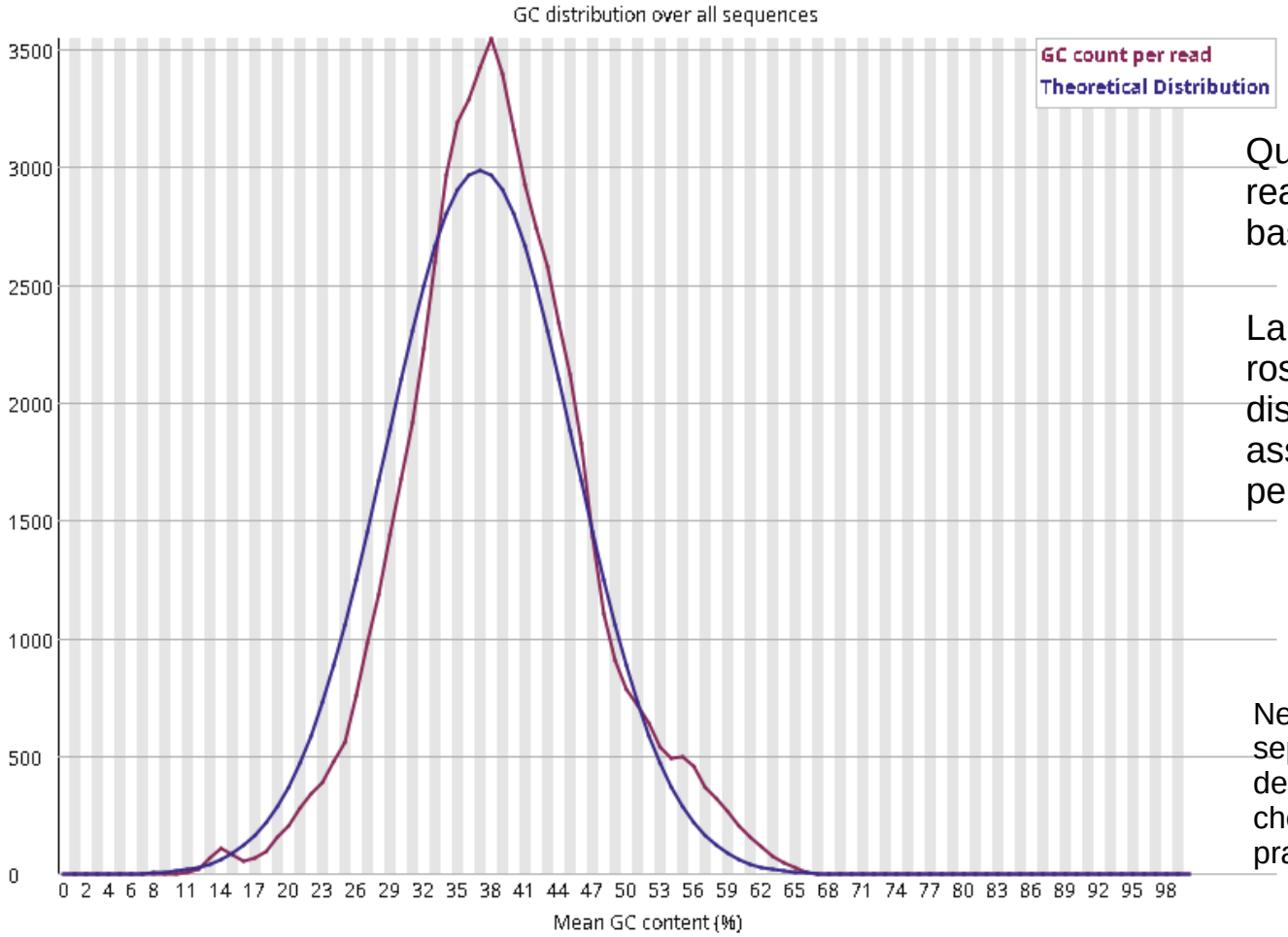


Questo grafico mostra la percentuale di ogni nucleotide in ogni posizione della sequenza.

In una libreria davvero random ci aspettiamo che la proporzione tra le coppie A e T e G e C sia uguale, quindi le linee in questo plot dovrebbero essere parallele l'una all'altra. Ma alcuni tipi di librerie produrranno sempre bias nella composizione delle seuqneze, di solito nei primi nucleotidi.

FastQC report

Per sequence GC content



Questo grafico mostra il numero di reads (asse y) VS la percentuale di basi GC per read (asse x).

La quantità di **GC per read reale** (in rosso) viene comparata ad una distribuzione **teorica** (in blu) creata assumendo un contenuto di GC uniforme per tutte le reads.

Nel caso ci fosse stato un ulteriore picco molto separato dalla distribuzione teorica si può dedurre che il campione contiene delle reads che hanno un contenuto in GC differente, in pratica una contaminazione del campione.

QUALITY CONTROL NANOPORE

Per valutare le nostre reads nanopore utilizzeremo NanoPlot, uno strumento simile a FastQC ma specifico per long reads.

```
(Quality_control) studente1@ladybug:~$ NanoPlot --fastq sweet-potato-chloroplast-nanopore-reduced.fastq -o Nanoplot_output
(Quality_control) studente1@ladybug:~$
```

Qui viene specificato tipo di file che stiamo usando come input

Qui viene specificato il nome della cartella di output dove verranno salvati tutti i risultati.

★

Come potete vedere paragonando NanoPlot a fastqc diversi strumenti vogliono diversi parametri in un diverso ordine. Nessuno ricorda esattamente l'ordine e il tipo di flag adatte per ogni tool, per questo vengono sempre in aiuto i manuali (per ogni tool c'è un manuale online o anche un manuale "interno" che può essere stampato a schermo usando la flag "-h" o "--help")

```
(Quality_control) studente1@ladybug:~$ cd Nanoplot_output/
(Quality_control) studente1@ladybug:~/Nanoplot_output$ ls
LengthvsQualityScatterPlot_dot.html   Non_weightedLogTransformed_HistogramReadlength.html
LengthvsQualityScatterPlot_dot.png    Non_weightedLogTransformed_HistogramReadlength.png
LengthvsQualityScatterPlot_kde.html   WeightedHistogramReadlength.html
LengthvsQualityScatterPlot_kde.png    WeightedHistogramReadlength.png
NanoPlot_20251006_0806.log            WeightedLogTransformed_HistogramReadlength.html
NanoPlot-report.html                  WeightedLogTransformed_HistogramReadlength.png
NanoStats.txt                         Yield_By_Length.html
Non_weightedHistogramReadlength.html  Yield_By_Length.png
Non_weightedHistogramReadlength.png
```

All'interno della cartella di output troverete diversi file; sono tutti i grafici riassuntivi di Nanoplot. Per avere uno sguardo completo su tutti apriremo **"Nanoplot-report.html"**

NanoPlot reports

Summary statistics

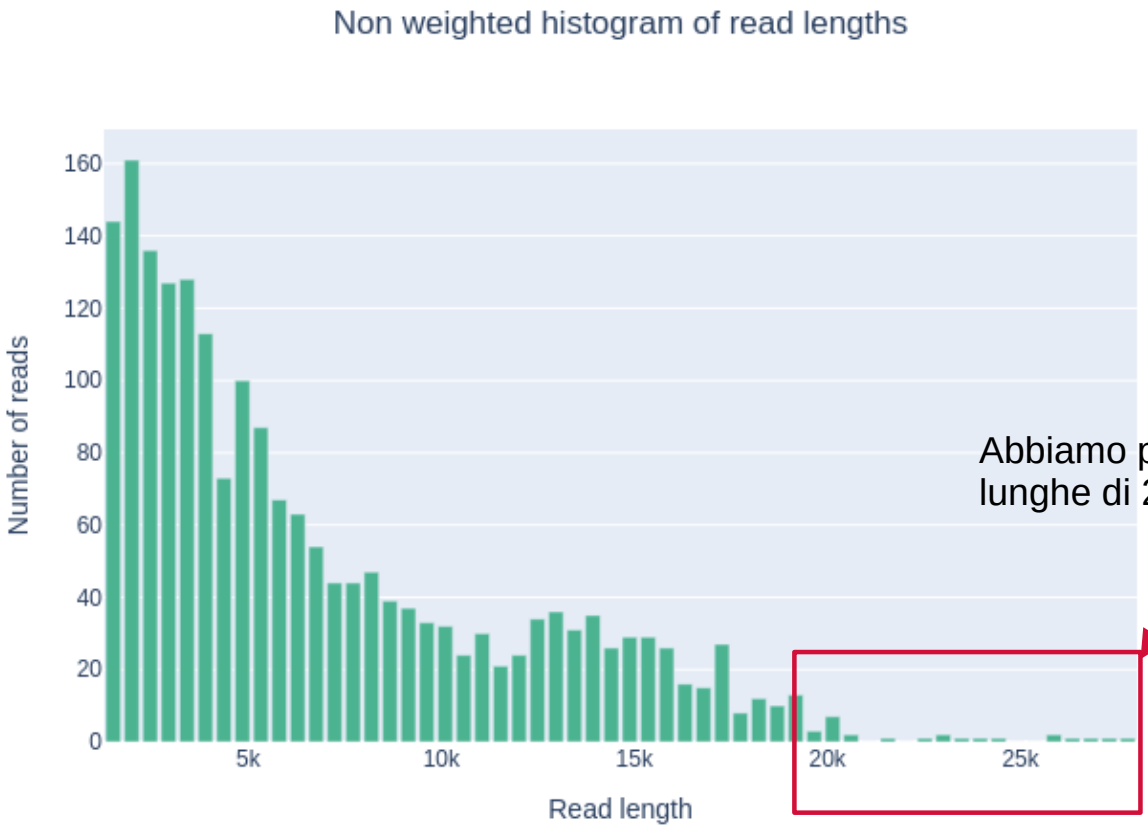
General summary	
Mean read length	6,664.9
Mean read quality	10.1
Median read length	4,942.0
Median read quality	10.6
Number of reads	2,000.0
Read length N50	10,403.0
STDEV read length	5,018.0
Total bases	13,329,799.0
Number, percentage and megabases of reads above quality cutoffs	
>Q10	1348 (67.4%) 9.4Mb
>Q15	0 (0.0%) 0.0Mb
>Q20	0 (0.0%) 0.0Mb
>Q25	0 (0.0%) 0.0Mb
>Q30	0 (0.0%) 0.0Mb
Top 5 highest mean basecall quality scores and their read lengths	
1	12.7 (1307)
2	12.6 (1538)
3	12.6 (3718)
4	12.6 (12406)
5	12.6 (3121)
Top 5 longest reads and their mean basecall quality score	
1	27779 (10.2)
2	27156 (10.9)
3	26598 (11.2)
4	26208 (11.2)
5	25656 (11.1)

Come valutiamo l'output delle reads Nanopore?

Dipende dallo scopo della tua analisi ma di solito:

1) **Sequencing quality**: lo score di qualità che ti dice con quanta probabilità una base sia corretta.
2) **Read lengths**: la tua analisi potrebbe necessitare reads di una certa lunghezza specifica
3) **Sequencing depth:** il numero di reads che "coprono" una certa posizione. (Coverage)

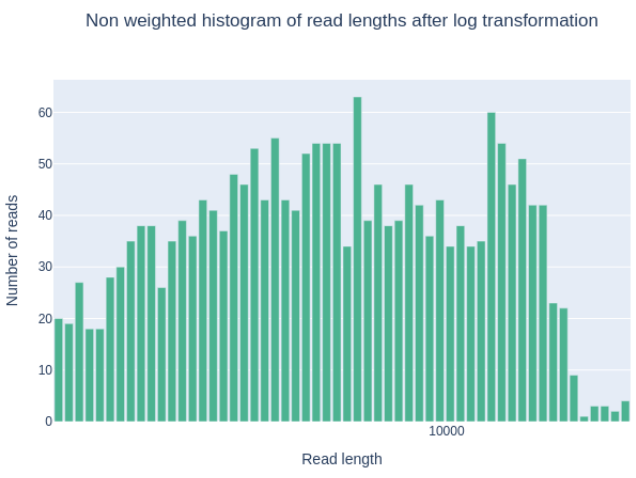
NanoPlot report



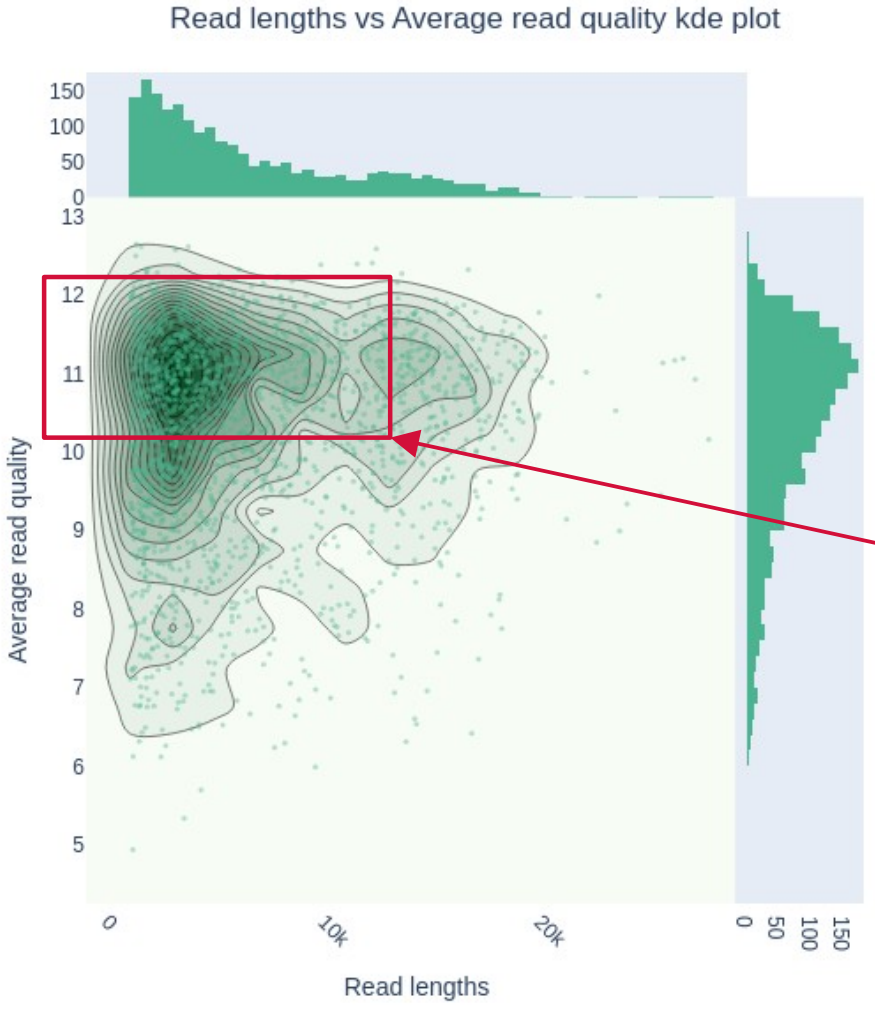
Questi plot mostrano la distribuzione della lunghezza dei frammenti del nostro file. Rispetto alla situazione con le illumina (erano tutte da 100 a 121), le long reads hanno una lunghezza molto più bariabile.

Abbiamo poche reads più lunghe di 20kb.

Questo è lo stesso grafico ma in versione logaritmica.



NanoPlot report



Questo plot invece mostra la distribuzione dei frammenti rispetto alla loro qualità. Questo modo di rappresentare permette di visualizzare entrambe le informazioni e mostrare possibili problemi.

Vediamo qua che la maggior parte delle reads è ad una qualità che va da Q10 a Q12, e che la loro lunghezza è principalmente sono a 10k.

Ci sarebbe molto altro da dire rispetto al quality control delle reads, esistono diversi tool che permettono di selezionare solo un determinato subset di reads o di tagliare parti che potrebbero portare ad errori, ma non è lo scopo di questa lezione.

Ora che abbiamo valutato la qualità delle nostre reads, passiamo all'assemblaggio.

ASSEMBLY

Ora usciamo dall'ambiente Quality_control, entriamo in Assembly e, se non l'avete già fatto, installate Flye:

conda deactivate → conda activate Assembly → conda install Flye

Una volta dentro Assembly usiamo Flye, un assemblatore che usa long reads e sfrutta un algoritmo "Overlap layout consensus"

```
$ flye --nano-raw sweet-potato-chloroplast-nanopore-reduced.fastq --genome-size 160000 -o Assembly
```

Flag per indicare che sono reads nanopore

Stima della grandezza finale del genoma

Nome della cartella di output

Quando avrà finito sul vostro terminale comparirà un breve resoconto di come è andato l'assemblaggio...

```
[2025-10-07 09:43:15] INFO: Assembly statistics:

        Total length:   202821
        Fragments:      4
        Fragments N50:  118131
        Largest frg:    118131
        Scaffolds:      0
        Mean coverage:  77

[2025-10-07 09:43:15] INFO: Final assembly: /home/ladybug/Assembly/assembly.fasta
```

...e vi indicherà il file di output →

ASSEMBLY

Una volta terminato il processo dentro la nostra cartella di output ci saranno diversi file:

```
(Assembly) studente1@ladybug:~$ cd Assembly/
(Assembly) studente1@ladybug:~/Assembly$ ls
00-assembly   20-repeat     40-polishing    assembly_graph.gfa  assembly_info.txt  params.json
10-consensus  30-contigger  assembly.fasta  assembly_graph.gv   flye.log
```

Quelli che ci interessano sono:

- **Assembly.fasta:** assemblaggio finale formato da contigs e possibili scaffolds
- **Assembly_graph.gfa:** grafo con tutte le informazioni che ci servono per interpretare l'assemblaggio
- **Assembly_info.txt:** informazioni extra riguardo ad ogni contig come lunghezza o coverage.

Per ora visualizzeremo solo il grafo con Bandage →

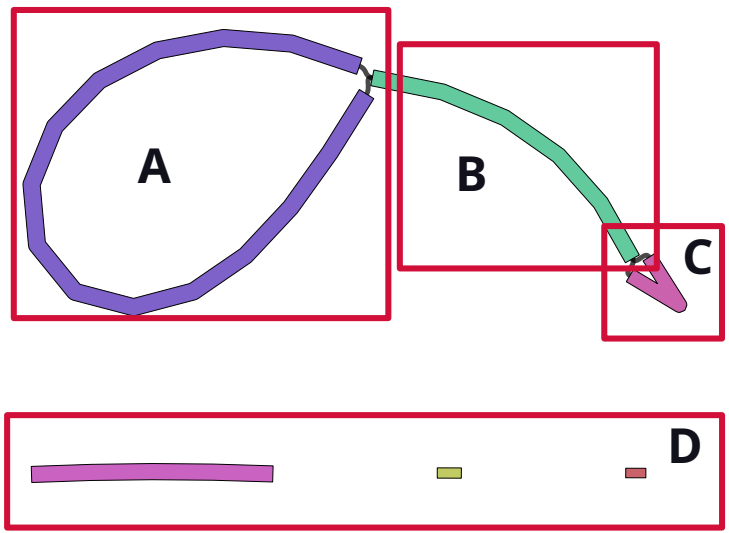
BANDAGE

Entriamo nell'ambiente conda Bandage (o se siete riusciti a scaricarlo dentro all'ambiente Assembly fatelo da lì) ed usiamo Bandage, un software che permette di visualizzare e analizzare i grafi di assemblaggi genomici (file .gfa) .

```
(Bandage) studente1@ladybug:~/Assembly$ Bandage image assembly_graph.gfa grafo_cloroplasto.svg
```

Io ho salvato l'immagine in formato .svg, ma Bandage crea anche immagini .png, scegliete voi quale preferite.

In questo caso Bandage vuole solo due file: il primo deve essere sempre l'input e il secondo sempre l'output.



Come interpretiamo il grafo?

C'è una lunga regione a singola copia (A) connessa ad una inverted repeat (B) connessa ulteriormente ad una regione a singola copia breve (C.). Nel grafo, ogni loop allle estremità è a singola copia mentre la parte centrale è la ripetizione inversa, collassata in un unica sequenza, che ha circa il doppio del coverage.
Le barre al di sotto (D) sono le sequenze che Flye è riuscito ad assemblare in dei contig abbastanza lunghi ma che non sapeva dove posizionare all'interno del grafo.

Questo assemblaggio necessita di essere corretto (POLISHING). È per questo che vengono usate le short-reads Illumina.