

UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INFORMATICA



Corso di Laurea Magistrale in Informatica
Curriculum Data Science & Machine Learning

Insegnamento di Strumenti Formali per la Bioinformatica

ANNO ACCADEMICO 2023/2024

**ASSEMBLAGGIO E ANNOTAZIONE DEL
GENOMA DEL SARS-COV-2**

Candidata:
Marianna Gambardella
Mat. 0522501698

Docenti:
Clelia De Felice
Rosalba Zizza
Rocco Zaccagnino

Indice

1	Introduzione	3
1.1	Che cos'è il Sars-CoV-2	3
1.2	Obiettivi e struttura	4
2	Tecnologie utilizzate	5
2.1	SPAdes	5
2.2	Bandage	6
2.3	Needle	7
2.3.1	Algoritmo Needleman-Wunsch per l'allineamento globale	7
2.4	Prokka	8
2.4.1	Algoritmo Prodigal	9
2.4.2	BLAST	9
3	Assemblaggio	11
3.1	Primo passo: Recuperare i dati di sequenziamento	11
3.2	Secondo passo: Importare i dati in Galaxy	13
3.3	Terzo passo: Assemblaggio attraverso SPAdes	16
3.4	Analisi dei risultati	19
4	Allineamento	21
5	Annotazione	23
6	Conclusioni	28

1 Introduzione

1.1 Che cos'è il Sars-CoV-2

Il Sars-CoV-2[1], dall'inglese *"Severe Acute Respiratory Syndrome CoronaVirus 2"*, è un virus respiratorio appartenente alla grande famiglia dei **coronavirus** e identificato alla fine del 2019, dopo essere stato segnalato in Cina nella città di Wuhan. Questa tipologia di virus risulta comunemente diffusa nel mondo animale, ma può evolversi e finire per infettare l'essere umano attraverso il fenomeno del *salto di specie*; il Sars-CoV-2 è, a oggi, il settimo coronavirus riconosciuto e diventato patogeno per l'uomo.

Il genoma dei coronavirus è costituito da un singolo filamento di RNA di grandi dimensioni; non esistono virus a RNA con genomi più grandi. La loro struttura è caratterizzata da una morfologia rotondeggiante e dalla presenza di piccoli spuntoni sulla superficie, che rappresentano la proteina **spike**, la quale, nel caso specifico del Sars-CoV-2, riveste un ruolo fondamentale nella trasmissione del virus permettendogli di introdursi nell'organismo legandosi alle cellule epiteliali del tratto respiratorio. La sequenza virale di questo virus risulta molto simile a quella del virus che causò la pandemia di SARS intorno al 2003.

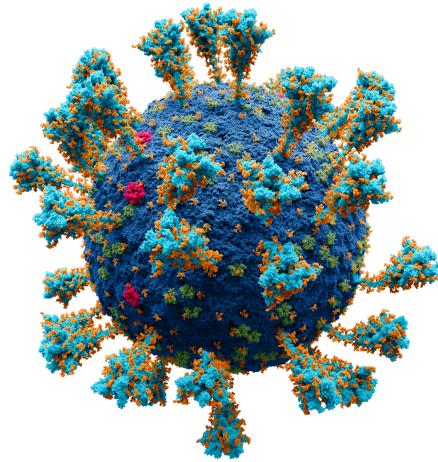


Figure 1.1: Struttura del Sars-CoV-2

La malattia scatenata dal virus prende il nome di **Covid-19** ("Co" indica corona, "vi" virus, "d" disease e 19 si riferisce all'anno di comparsa) e, a differenza dei precedenti coronavirus, è caratterizzata da un periodo di incubazione (asintomatico e contagioso) che va da 2 a 14 giorni; inoltre, la sintomatologia, che può essere in alcuni casi persino assente, dipende dalla gravità della malattia e può manifestarsi sottoforma di febbre, tosse, mal di gola, debolezza, affaticamento e dolore muscolare, nei casi meno gravi, e di polmonite, sindrome da stress respiratorio acuto con complicazioni potenzialmente mortali, nei casi più gravi. Sintomi più rari includono anche cefalea, brividi, mialgia, astenia, vomito e/o diarrea, perdita improvvisa o diminuzione dell'olfatto e perdita o alterazione del gusto.

1.2 Obiettivi e struttura

Il presente elaborato ha l'obiettivo di descrivere il processo di assemblaggio e di annotazione del Sars-CoV-2 e di allineamento con il virus Sars-CoV, per verificarne la somiglianza genomica. Per lo svolgimento è stata seguita la guida dell'assignment di Phillip Compeau[2] nel contesto del corso da lui tenuto in *Great Ideas in Computational Biology*[2], nella primavera del 2021 presso la *Carnegie Mellon University*.

Nel capitolo 2 saranno presentati i tool impiegati nell'analisi e gli algoritmi principali su cui essi sono basati. Nei capitoli 3, 4 e 5 saranno invece descritte le fasi di svolgimento del progetto, corredate da indicazioni dettagliate e immagini esplicative delle schermate ottenute dall'esecuzione e opportunamente commentate attraverso le risposte alle domande poste da Phillip Compeau.

2 Tecnologie utilizzate

Per evitare di installare localmente i tool descritti di seguito, è stata utilizzata la piattaforma australiana **Galaxy**, un progetto *open-source* che fornisce un’interfaccia web *user-friendly* e intuitiva e che permette di eseguire su cloud vari software bioinformatici in maniera semplice ed efficiente.

2.1 SPAdes

L’assemblatore *SPAdes*[3] è un software open-source per l’assemblaggio di genomi. La sua prima versione è stata pubblicata nel 2012 e sviluppata presso il ”Center for Algorithmic Biotechnology” dell’Università Statale di San Pietroburgo, in Russia; infatti il nome sta per ”*St. Petersburg genome Assembler*”. Il tool opera su genomi di piccole dimensioni, prende in input *short reads* come quelle generate dal sequenziamento di *Illumina* e implementa un approccio basato sul **grafo di de Bruijn**.

Il processo di assemblaggio è così strutturato:

Stage 1: Costruzione dell’**assembly graph** (multidimensionale), ovvero un grafo risultante dalla combinazione di molteplici grafi di de Bruijn (ciascuno con *k-mer* di dimensioni differenti), ripuliti da eventuali *bubble* e semplificati comprimendo tutti i cammini massimali di tipo *non-branching* (sequenze lineari di nodi e archi, senza biforazioni) in archi singoli. SPAdes utilizza tipicamente tre valori diversi di *k*, uno alto, uno intermedio e uno basso (di default $k = 21$, $k = 33$ e $k = 55$) per facilitare la gestione delle variazioni di *coverage* lungo le sequenze del genoma. Inoltre, in questo primo stadio vengono anche create delle strutture dati per conservare le informazioni sul mapping tra le read originali e l’assembly graph, in modo da poter tenere traccia di ogni operazione effettuata.

Stage 2: Perfezionamento dei cosiddetti *k-bimer*, ciascuno definito come la tripla $(\alpha|\beta, d)$, in cui α e β sono *k-mer* e d è un intero che stima la distanza tra specifiche istanze di α e β nel genoma. Ciò viene fatto attraverso una serie di trasformazioni, sia analizzando gli histogrammi delle distanze, sia esaminando i cammini nel grafo al fine di migliorare l’accuratezza dell’assemblaggio finale. In Figura 2.1 è mostrato il processo dettagliato.

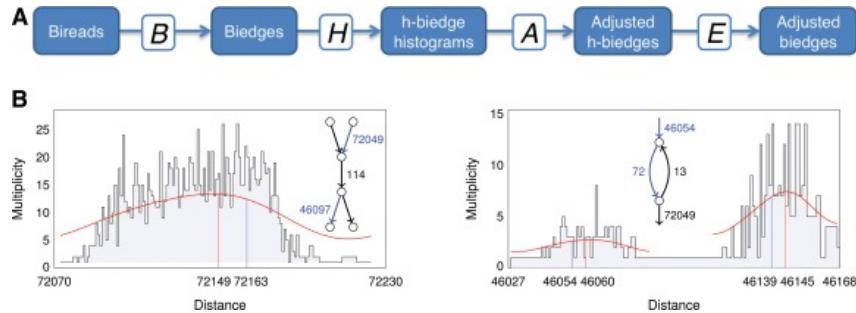


Figure 2.1: Stage 2 of SPAdes. (A) Breads are decomposed into pairs of *k-mers* with estimated genomic distances (B-transformation). These are tabulated into histograms of estimated genomic distances between pairs of h-edges (H-transformation), and peaks in the histograms and paths in the graph are used to reveal the actual genomic distances between h-edges (A-transformation). This may be converted back to genomic distances between *k-mers* on pairs of h-paths (E-transformation, used for presentation purposes but not needed in the implementation). (B) The h-biedge histogram $(\alpha|\beta, *)$ corresponding to the exact h-biedge $(\alpha|\beta, 72163)$ in the assembly graph. $\text{path}(\alpha)$ is an h-path (condensed edge representing 72049 edges) in the upper right, and $\text{path}(\beta)$ is an h-path (representing 46097 edges) at the lower left. The histogram collects all distance estimates between α and β derived from breads. The h-biedge histogram was smoothed using the Fast Fourier Transform (red curve). The peak in the smoothed histogram (marked red) well approximates the actual distance (marked blue). (C) The h-biedge histogram $(\alpha|\beta, *)$ estimates the distance between h-edges α and β ($|\text{path}(\alpha)| = 46054$, $|\text{path}(\beta)| = 72$). Because of the directed cycle formed by the two h-paths of lengths 72 and 13, there may be multiple walks through the graph between α and β . The h-biedge histogram has been divided into clusters with centers at 46060 and 46145. Thus SPAdes transforms the entire histogram into two h-biedges: $(\alpha|\beta, 46054)$ and $(\alpha|\beta, 46139)$.

Stage 3: Costruzione del **paired assembly graph**, un'estensione dell'assembly graph iniziale attraverso l'inclusione delle informazioni sulle distanze risultanti dallo studio precedente e ottenute considerando le *paired-end reads* (coppie di read appartenenti ad entrambe le estremità del frammento di DNA, sia quello *forward*, sia quello *reverse*).

In Figura 2.2 è mostrato ed esplicato un esempio preciso.

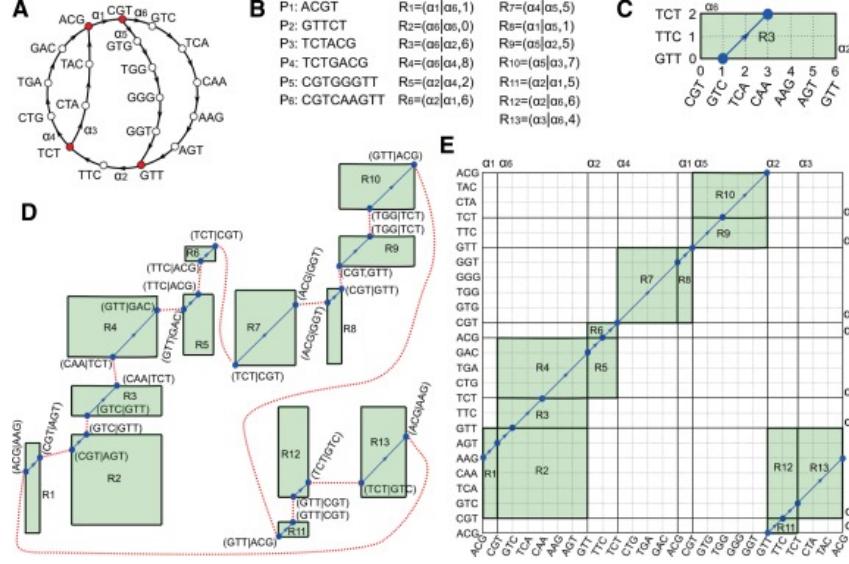


Figure 2.2: Construction of the paired assembly graph for bireads sampled from a circular 24 bp genome Genome = ACGTCAAGTTCTGACGTGGGTTCT (single reads referred to as Reads). The de Bruijn graph DB(Reads, 4) has four hubs (ACG, CGT, GTT, and TCT) (A) and six h-paths P_1, \dots, P_6 , with lengths 1, ..., 6 respectively (B). The h-edge of path P_i , denoted α_i , is its first edge. The cycle C in DB(Reads, 4) that spells Genome passes through the h-paths in order $P_1, P_6, P_2, P_4, P_1, P_5, P_2, P_3$ (P_1 and P_2 represent repeats). (B) Reads are paired with separation $d = 5$, yielding estimated distances D between various h-edges α_i and α_j , denoted as the h-bedge $(\alpha_i|\alpha_j, D)$. The 13 h-bedges constructed from all bireads are listed as R_1, \dots, R_{13} . (C) The rectangular diagram of h-bedge $(\alpha_6|\alpha_2, 6)$ is a rectangle (R_3) with sides P_6 and P_2 and 45° line segment $y = x + (d - 4) = x - 1$, from (1, 0) to (3, 2). Point (1, 0) is labeled by bivertex (GTC|GTT) formed by vertex 1 (GTC) in path P_6 and vertex 0 (GTT) in path P_2 . Point (3, 2) is labeled by bivertex (CAA|TCT) formed by vertex 3 (CAA) in path P_3 and vertex 2 (TCT) in path P_2 . (D) Vertices to glue together from different rectangle diagrams are indicated by dotted red lines. (E) Rectangles glued into a 24x24 grid, yielding a cycle (blue path) through the genome.

Stage 4: Generazione delle sequenze continue (**contig**), derivate dal paired assembly graph e corrispondenti all'output finale del processo di assemblaggio, che rappresentano la ricostruzione delle regioni del genoma.

Occorre precisare che SPAdes è anche in grado di identificare e correggere gli eventuali errori presenti nelle reads, in fase preliminare, per migliorarne la qualità prima di procedere con l'assemblaggio.

2.2 Bandage

Bandage[4] è una GUI sviluppata per la visualizzazione degli *assembly graph* generati dai tool per l'assemblaggio dei genomi, come SPAdes. Utilizza infatti algoritmi per il layout di grafi in grado di posizionare in maniera automatica ed efficiente i nodi, ma è anche dotato di un sistema di personalizzazione che permette all'utente di riarrangiari e ridefinirne colori e forme, pur preservando la struttura originale.

Ciascun nodo rappresenta un *contig* e può essere etichettato in base all'ID, alla lunghezza della sequenza o al grado di *coverage*. Visualizzare le sequenze continue insieme alle connessioni tra loro fornisce un modo sia per valutare l'assemblaggio ed eventuali regioni critiche per poterlo perfezionare, sia per confrontare visivamente gli assemblaggi

ottenuti da diversi tool.

In Figura 2.3 sono riportati alcuni esempi di grafi tramite Bandage.

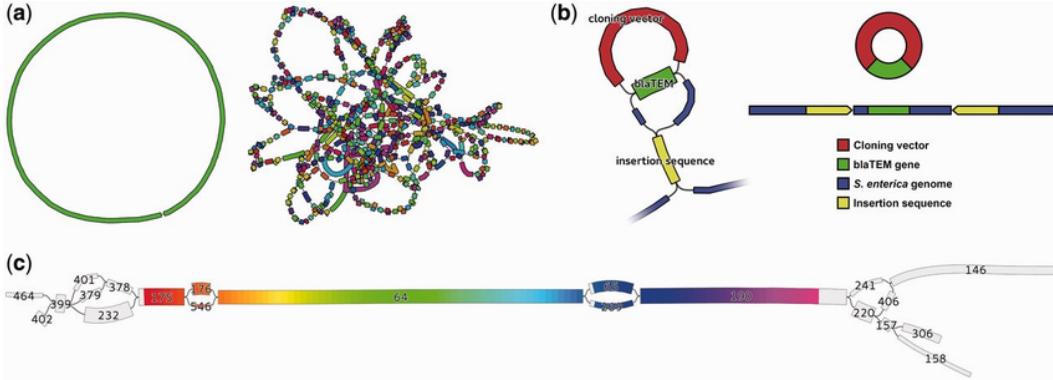


Figure 2.3: Examples of Bandage visualization (a) Left, ideal bacterial assembly (single contig); right, poor assembly with many short contigs. (b) Left, zoomed-in view of *Salmonella* assembly; repeated sequences (blaTEM and insertion sequence) appear as single nodes with multiple inputs and outputs. Node widths are scaled by read coverage (depth). Right, underlying gene structure deduced from Bandage visualization. (c) 16S rRNA region of a bacterial genome assembly graph, highlighted by Bandage's integrated BLAST search. Nodes are labelled with their ID numbers and their widths are scaled by coverage.

2.3 Needle

Needle[5] è un tool che appartiene al package **EMBOSS** (*European Molecular Biology Open Software Suite*) e che si occupa dell'allineamento di coppie di sequenze utilizzando l'algoritmo **Needleman-Wunsch** per trovare l'allineamento ottimale.

2.3.1 Algoritmo Needleman-Wunsch per l'allineamento globale

L'algoritmo *Needleman-Wunsch* appartiene alla classe di algoritmi in grado di calcolare il miglior allineamento attraverso un numero di passi pari a mn , dove m ed n sono le rispettive lunghezze delle due sequenze; implementa infatti la tecnica della programmazione dinamica esplorando ogni possibile allineamento per individuare quello migliore.

Date due sequenze da allineare

$$A = a_1 a_2 \dots a_{i-1} a_i \dots a_m$$

$$B = b_1 b_2 \dots b_{j-1} b_j \dots b_n$$

segue una descrizione dell'algoritmo:

1. Viene definita una matrice $D(i, j)$ con $m + 1$ righe e $n + 1$ colonne, nella cui cella $D(i, j)$ è memorizzato il valore del punteggio dell'allineamento globale tra le sequenze $A_i = A[1, i]$ e $B_j = B[1, j]$, $\forall i = 0, 1, \dots, m$ e $\forall j = 0, 1, \dots, n$;

2. Ogni cella della matrice è riempita nel seguente modo:

$$D(i, j) = \max \begin{cases} D(i - 1, j - 1) + \delta(a_i, b_j) \\ D(i - 1, j) + \delta(a_i, -) \\ D(i, j - 1) + \delta(-, b_j) \end{cases}$$

dove $\delta(a_i, b_j) = k$ se $a_i = b_j$, $\delta(a_i, b_j) = s$ se $a_i \neq b_j$, $\delta(a_i, -) = d$ e $\delta(-, b_j) = d$, indicando con k il punteggio per un **match**, con s il punteggio per un **mismatch** e con d il costo per i **gap**. Per questi ultimi, inseriti al fine di ottimizzare l'allineamento, si distinguono due tipologie di costi:

- **Gap open penalty:** costo sottratto dal punteggio per ogni gap introdotto;
- **Gap extension penalty:** costo associato ai residui di gap esistenti, che rappresenta la penalizzazione applicata ai gap estesi; tale valore è tipicamente 5-10 volte più basso del precedente, in quanto è preferibile avere pochi gap estesi, piuttosto che molti gap brevi.

3. $D(m, n)$ è detta *cella ottima* in quanto corrisponde al punteggio dell'allineamento globale ottimale.

Trattandosi di un algoritmo per l'allineamento globale, si occupa di allineare le due sequenze considerando le loro intere lunghezze; inoltre, alle porzioni delle sequenze che escono al di fuori dei limiti della regione allineata non viene applicato alcun tipo di penalty.

2.4 Prokka

Prokka[6] è un tool sviluppato per l'annotazione di genomi procariotici (virus, batteri e archei). Si tratta di un software che prende in input l'assemblaggio risultante di un genoma e che coordina l'esecuzione di altri tool in base al tipo di genoma da annotare (fig. 5.4).

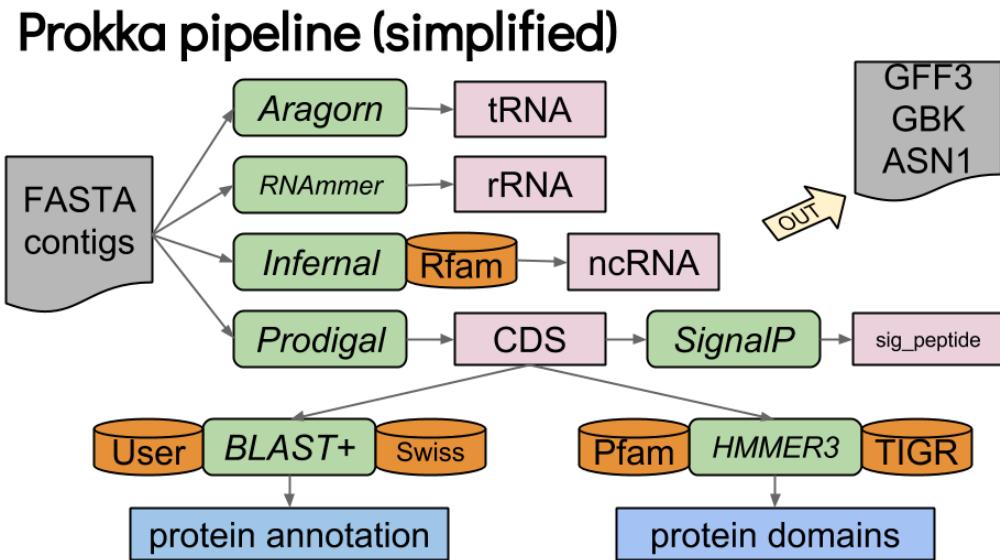


Figure 2.4: Descrizione dell'utilizzo del tool Prokka per l'annotazione di genomi

Il processo prevede l'individuazione dei cosiddetti *putative genes*, ovvero geni predetti a partire dalle sequenze del genoma in base alla somiglianza riscontrata con geni esistenti. Quest'ultima operazione può essere effettuata in vari modi, uno di questi consiste nell'identificare le regioni che hanno il potenziale di essere tradotte in proteine, ovvero le sequenze che iniziano con un **codone di inizio** (ATG) e terminano con un **codone di stop** (TAG, TAA, o TGA); queste ultime sono dette **Open Reading Frames** (ORF) e hanno lunghezza variabile, tuttavia non è detto che codifichino sempre proteine.

Il paragrafo successivo descrive l'algoritmo **Prodigal**, utilizzato da Prokka per la predizione dei geni codificanti.

2.4.1 Algoritmo Prodigal

Prodigal[7] (*Prokaryotic Dynamic Programming Genefinding Algorithm*) è un algoritmo di programmazione dinamica che è in grado di individuare i geni che codificano proteine all'interno di una data sequenza genomica.

Segue una descrizione semplificata dei passi principali compiuti dall'algoritmo:

- La fase preliminare dell'algoritmo è caratterizzata da un processo di training, in cui viene attraversata l'intera sequenza genomica in input e analizzata ciascuna ORF per determinare il *bias* in relazione alle basi G e C, in tutte e tre le posizioni di ogni codone (se una specifica posizione mostra un'elevata occorrenza di G o C, allora è presente un rispettivo bias); infatti, in genomi con un'alta concentrazione di G e C, a causa del numero ridotto di A e T, ci sono molti meno codoni di stop e lunghe sequenze ORF si presentano semplicemente in maniera random e nella maggior parte dei casi non sono nemmeno codificanti. Successivamente, attraverso la tecnica di programmazione dinamica, vengono costruiti dei modelli di geni in base al bias riscontrato per identificare le ORF potenzialmente codificanti.
- Considerando poi i cosiddetti *dicodoni* (o *hexamer*), ossia sequenze formate da due codoni adiacenti, e le statistiche sulla loro probabilità di occorrenza all'interno dei modelli creati nella fase precedente, viene assegnato un primo punteggio ai potenziali geni; tale punteggio viene progressivamente raffinato dalle restanti fasi dell'algoritmo, applicando opportune penalità e basandosi su soglie prestabilite, fino a quando l'insieme delle ORF col punteggio più alto non resta costante nelle ripetute iterazioni. Questo approccio consente all'algoritmo di produrre un output estremamente accurato.

2.4.2 BLAST

Una volta individuate le regioni potenzialmente codificanti, Prokka effettua una ricerca di queste ultime all'interno di database proteici, come *UniProt*, per identificare somiglianze e conseguentemente assegnare specifiche funzioni putative ai geni predetti. Tale ricerca viene eseguita utilizzando l'algoritmo *BLAST*[8], il cui funzionamento può essere così descritto:

1. La query (una sequenza proteica o una sequenza di nucleotidi) viene scomposta in frammenti più piccoli che si sovrappongono, ovvero "parole" di lunghezza W (**W-mer**).

2. Per ciascun W-mer così individuato viene generata una lista di parole simili, memorizzata in un database o in una hash table. A ogni elemento viene poi assegnato un punteggio in base al grado di somiglianza con la sottosequenza della query e a una certa soglia prestabilita T (le parole con punteggio inferiore a tale soglia vengono eliminate); il punteggio più alto corrisponde naturalmente al match completo. I primi due step sono ripetuti per ciascun W-mer della sequenza di query.
3. Ogni parola della lista creata viene cercata all'interno delle sequenze contenute nel database di riferimento e, non appena si verifica un match, la ricerca viene estesa in entrambe le direzioni per individuare i cosiddetti **High Scoring Pairs** (HSP), ovvero i segmenti di allineamento locale caratterizzati da un alto punteggio di similarità (considerando una certa soglia X).
4. Si restituiscono i match con i punteggi totali più alti.

Il processo è illustrato anche in Figura 2.5.

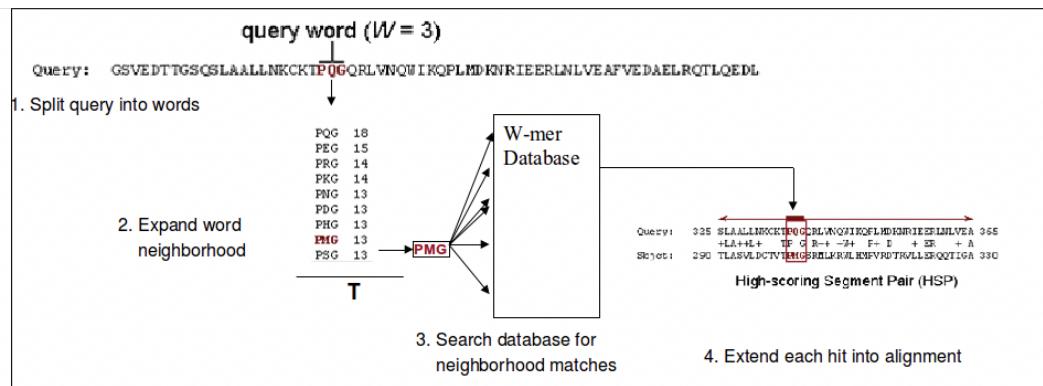


Figure 2.5: Algoritmo BLAST per la ricerca

3 Assemblaggio

Il seguente capitolo si occupa di descrivere il processo di assemblaggio effettuato utilizzando il tool SPAdes, all'interno del servizio Galaxy. Nel corso della descrizione, sono inoltre riportate le risposte alle domande poste da Phillip Compeau.

3.1 Primo passo: Recuperare i dati di sequenziamento

Per poter procedere con l'assemblaggio del virus occorre prima di tutto procurarsi i dati di sequenziamento necessari; questi ultimi possono essere raccolti dalla piattaforma del *National Center for Biotechnology Information*, nello specifico nella sezione *Sequence Read Archive*, utilizzando l'identificativo opportuno.

In Figura 3.1 è mostrata la schermata di accesso al dataset di interesse. E' possibile notare che il sequenziamento considerato è costituito da circa 719,800 reads (**Spots**) e che si tratta di *paired-end short reads* ottenute tramite **Illumina**, considerando entrambe le estremità del frammento (**PAIRED**).

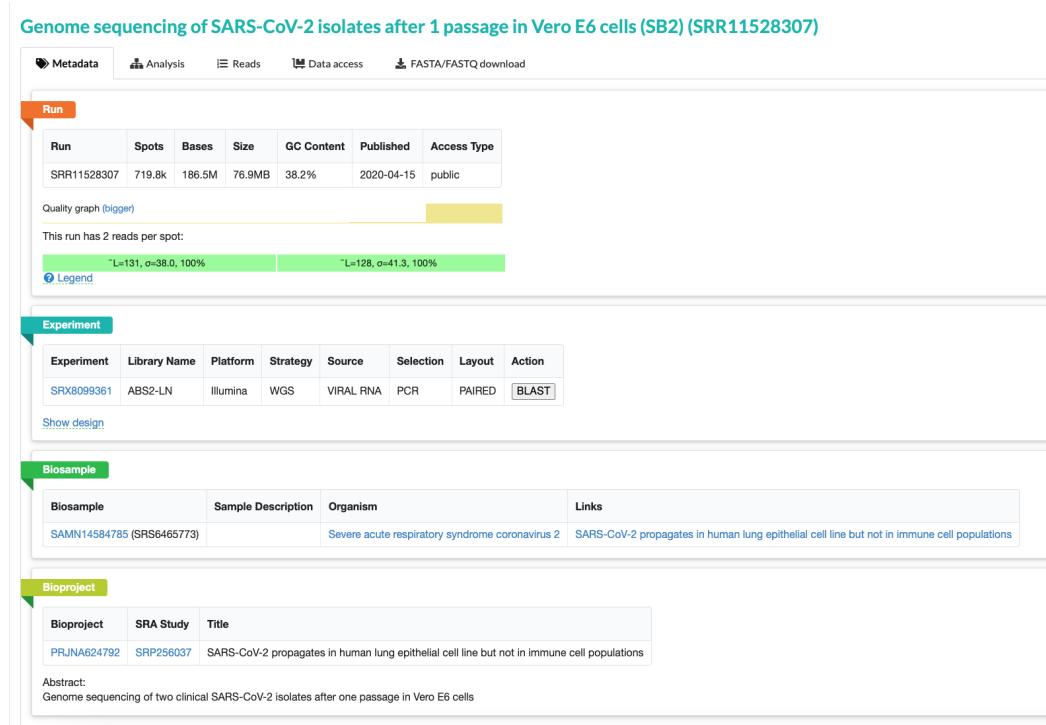


Figure 3.1: Schermata dal sito web del National Center for Biotechnology Information relativa ai dati di sequenziamento

Domanda 1: Quante basi nucleotidiche si trovano all'interno di questo file? Qual è la percentuale di reads che sono G oppure C?

Osservando la schermata è possibile dedurre che il file contiene 186,500,000 basi nucleotidiche e una percentuale di G e C pari al 38.2%.

Domanda 2: Sappiamo che il genoma del Sars-CoV dell'epidemia del 2003 è costituito da approssimativamente 30,000 nucleotidi. Se il genoma del Sars-CoV-2 ha una lunghezza pari a 30,000 nucleotidi, qual è la coverage delle reads di questo dataset?

Per calcolare la *coverage* è possibile far riferimento alla seguente formula:

$$Coverage = \frac{nl}{L}$$

dove n sta per il numero di reads, l per la lunghezza delle reads e L per la lunghezza totale del genoma. Dunque, considerando che è specificata una lunghezza media pari a 128 per la prima read della coppia e a 131 per la seconda, effettuando la media delle tra le due si ottiene 129.5, che occorre moltiplicare ulteriormente per 2 trattandosi di paired-end reads. Pertanto risulta:

$$Coverage = \frac{719,800 \times (2 \times 129.5)}{30000} = 6,214.3$$

Ciò significa che, in media, ciascun nucleotide del genoma è coperto da 6,214.3 reads secondo il sequenziamento in esame.

Cliccando sulla tab relativa alle reads, che ne mostra dieci paia per volta, si può osservare che queste sono espresse nel formato **FASTA**; sono infatti introdotte da un header preceduto dal simbolo ">" (fig. 3.2).

Genome sequencing of SARS-CoV-2 isolates after 1 passage in Vero E6 cells (SB2) (SRR11528307)

The screenshot shows a web interface for managing SRA datasets. At the top, there are tabs for 'Metadata', 'Analysis', 'Reads' (which is selected), 'Data access', and 'FASTA/FASTQ download'. Below the tabs is a 'Filter' section with a search input and a 'Find' button. A tooltip 'What can the filter be applied to?' is visible. The main area displays a list of reads with the following details:

- Reads: 719,807 reads**
- Page: 1 / 71,981**
- quality scores** and **advanced options** buttons
- Reads (separated)**
- List of reads numbered 1 to 10, each with a name and member count (e.g., 1 SRR11528307.1, name: MN01288:4:000H32WJK:1:1110, member: 5).
- Sequence snippets for reads 1 through 10, showing the first few bases.

Figure 3.2: Schermata dal sito web del National Center for Biotechnology Information che mostra la finestra delle reads

3.2 Secondo passo: Importare i dati in Galaxy

Il dataset può dunque essere importato in Galaxy. Per farlo basterà cliccare su **”Download and Extract Reads in FASTQ Format from NCBI SRA”**, sotto **”Get Data”**, assicurarsi che il menu a tendina in corrispondenza di **”select input type”** mostri **”SSR accession”** e digitare l’accession ID del dataset (*SRR11528307*) nel campo corrispondente. Il processo di importazione avrà inizio dopo aver cliccato su **”Run Tool”** e, una volta completato, sarà osservabile cliccando sull’icona simboleggiata da un occhio nella History del progetto a destra della pagina.

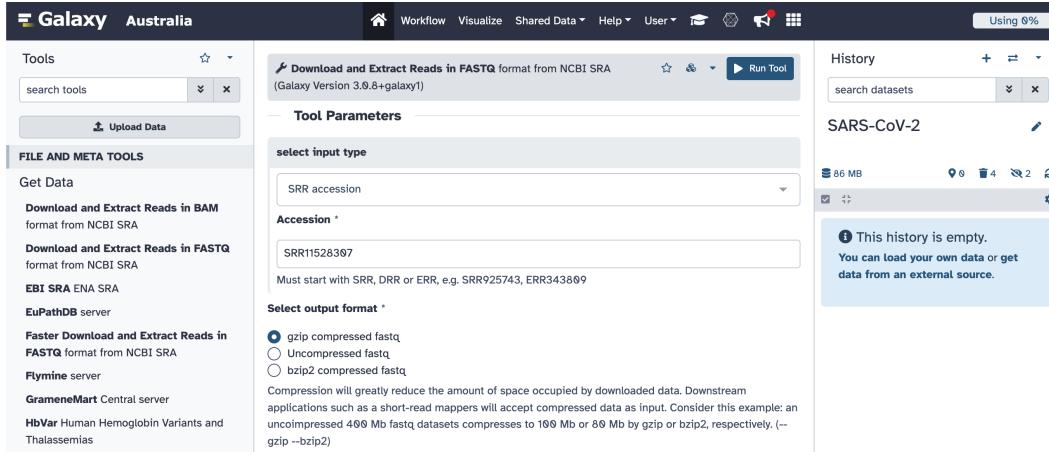


Figure 3.3: Schermata dalla piattaforma Galaxy che mostra l'importazione dei dati di sequenziamento

Il file così importato è nel formato **FASTQ**, un'estensione del formato FASTA che rappresenta ogni read con quattro righe così strutturate:

- Un header contenente il simbolo iniziale "@" che termina con "/1" o "/2", a seconda della read nella coppia;
- La sequenza primaria della read;
- Un header contenente il simbolo "+", che segna la fine della sequenza nucleotidica (potrebbero essere presenti ulteriori informazioni, assenti nel caso in esame);
- Una sequenza di simboli ASCII che rappresentano i **punteggi di qualità Phred**, uno per ciascuna base.

Dopo il processo di sequenziamento, a ogni base è associata una stima Q della probabilità della sua correttezza. Il calcolo viene effettuato su scala logaritmica ($Q = -10 \log_{10}(p)$, dove p è la probabilità che la base sia errata) e viene assegnato il carattere ASCII i -esimo, corrispondente a $Q + 33$, all' i -esima base. Più alto è il valore di Q , più bassa è la probabilità che la base non sia corretta: in generale, una base è ritenuta corretta quando $Q \geq 50$, mentre è ritenuta buona quando $Q \geq 30$, dove 50 e 30 sono considerate le probabilità massime di errore nei rispettivi casi (base buona e base corretta).

Domanda 3: In che modo potrebbero essere utili i punteggi Phred quando si applica l'approccio del grafo di de Bruijn per l'assemblaggio del genoma?

I punteggi di qualità Phred possono essere utilizzati nella fase di pre-processing per effettuare il *trimming* delle reads: viene fissata una soglia minima di Q in base alla quale ogni read è sostituita dalla più lunga sottostringa in cui ciascuna base ha un punteggio pari almeno a Q , se sufficientemente lunga, altrimenti viene rimossa. Nel caso specifico del grafo di de Bruijn e durante la fase di esecuzione dell'assemblaggio, ritengo che potrebbe risultare utile considerare i punteggi Phred nella scelta della lunghezza dei k-mer, in modo da evitare di scegliere valori di k che generino sequenze caratterizzate unicamente da un numero elevato di basi con punteggi bassi, aumentando così l'affidabilità dell'assemblaggio risultante.

Domanda 4: Quali sono i punteggi di qualità della read "8595/1"? Sono buoni? Ci sono dei nucleotidi che ti preoccupano?

I punteggi di qualità della read "8595/1" sono i seguenti:

```
@MN01288:4:000H32WJK:1:11101:10036:8595/1
TTTATATACTGCTCATCTTCCAAGTTCTGGAGATCGATGAGAGATTCAATTCTGGCACCTCATTGAGGC GGTCATTTCTTTGAATG
+
F/FFFFFFFFFFFFF/FFFF=FFFFFAFFFAAFFFFFFF/FFFFFFFFFFFFF/FFFFFFFFFFFAFAF//FFFFFFF/F/AFFF//A
```

Figure 3.4: Read "8595/1"

Utilizzando la tabella di conversione ASCII per i valori di Q (fig. 3.5), si può notare che la maggior parte dei punteggi è pari a 37 (dieci nucleotidi presentano un punteggio pari a 32) e può essere pertanto considerata buona; tuttavia sono presenti alcuni valori più bassi, in particolare un 28 e nove 14, corrispondenti a specifici nucleotidi, che risultano preoccupanti.

ASCII_BASE=33 Illumina, Ion Torrent, PacBio and Sanger											
Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII
0	1.00000	33 !	11	0.07943	44 ,	22	0.00631	55 7	33	0.00050	66 B
1	0.79433	34 "	12	0.06310	45 -	23	0.00501	56 8	34	0.00040	67 C
2	0.63096	35 #	13	0.05012	46 .	24	0.00398	57 9	35	0.00032	68 D
3	0.50119	36 \$	14	0.03981	47 /	25	0.00316	58 :	36	0.00025	69 E
4	0.39811	37 %	15	0.03162	48 0	26	0.00251	59 ;	37	0.00020	70 F
5	0.31623	38 &	16	0.02512	49 1	27	0.00200	60 <	38	0.00016	71 G
6	0.25119	39 '	17	0.01995	50 2	28	0.00158	61 =	39	0.00013	72 H
7	0.19953	40 (18	0.01585	51 3	29	0.00126	62 >	40	0.00010	73 I
8	0.15849	41)	19	0.01259	52 4	30	0.00100	63 ?	41	0.00008	74 J
9	0.12589	42 *	20	0.01000	53 5	31	0.00079	64 @	42	0.00006	75 K
10	0.10000	43 +	21	0.00794	54 6	32	0.00063	65 A			

Figure 3.5: The current Phred score table. Each row contains a quality score Q, the corresponding probability of error, and the ASCII symbol used to label a nucleotide having this quality score. Source: <https://bit.ly/2NrZvS7>.

Ritornando alla pagina del Sequence Read Archive del dataset, nella finestra **”Metadata”**, cliccando sulla scritta **”bigger”** accanto a **”Quality graph”**, viene mostrato un istogramma dei punteggi di qualità Phred dell’intero dataset (fig. 3.6).

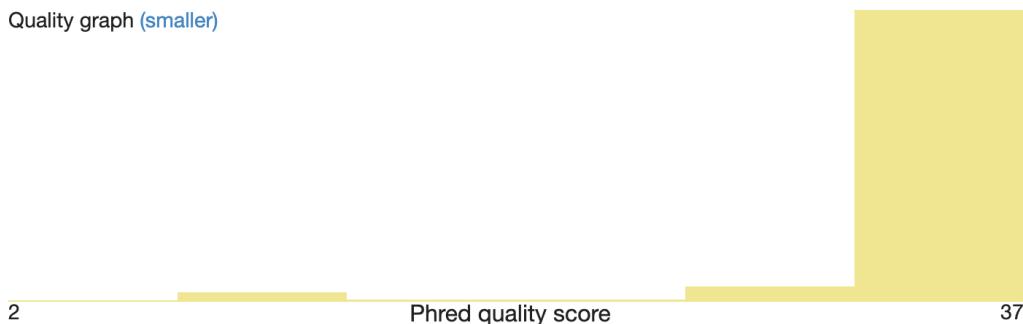


Figure 3.6: Istogramma dei punteggi di qualità Phred del dataset

Domanda 5: *Interpreta l’istogramma. I punteggi di qualità sono buoni? Cosa te lo fa pensare?*

Dall’istogramma si può osservare che la più alta concentrazione dei punteggi corrisponde al valore pari a 37, dunque si può affermare che i punteggi di qualità sono nel complesso buoni ($37 \geq 30$).

3.3 Terzo passo: Assemblaggio attraverso SPAdes

E’ possibile ora procedere con l’assemblaggio vero e proprio del genoma, utilizzando SPAdes. Cliccando su **”Assembly”**, sotto **”Genomics Analysis”**, verrà mostrata una serie di assemblatori, tra cui occorrerà selezionare SPAdes. Nello specifico è stata utilizzata la versione 3.12.0+galaxy1, in quanto quella più recente mostrava un’interfaccia e delle opzioni differenti rispetto a quanto documentato dall’assignment di Phillip Compeau. A questo punto, prima di procedere, devono essere selezionate le opzioni giuste per l’assemblaggio:

- Il campo **”Single cell?”** non deve essere selezionato, poichè non si tratta di un progetto su batteri di cui si possiede solo il DNA di una singola cella;
- Il campo **”Run only assembly?”** non deve essere selezionato, in quanto vogliamo che SPAdes non si occupi soltanto dell’assemblaggio, ma anche di cercare e correggere eventuali errori, come ad esempio reads con basi con punteggi di qualità particolarmente bassi;
- Il campo **”Careful correction?”** deve essere selezionato;
- Il campo **”Automatically choose k-mer values”** deve essere selezionato, infatti non avendo alcuna informazione per stabilire a priori i valori dei k-mer, risulta opportuno affidarsi al sistema di machine learning di SPAdes;

- Il campo **”Coverage cutoff”** non deve essere selezionato, non essendo necessario per l’assemblaggio di un virus;
- Il campo **”Libraries are ionTORRENT reads?”** non deve essere selezionato, poichè si tratta di reads generate da Illumina;
- Nel campo **”Library type”** deve essere selezionata la voce **”Paired end/single reads”**, poichè quelle in esame sono paired-end reads;
- Nel campo **”Orientation”** deve essere selezionata la voce **”fr”**, che sta per *forward*, infatti la convenzione vuole che il DNA sia letto sempre nella direzione da 5’ a 3’.

Dopodichè sarà possibile selezionare il file su cui effettuare l’assemblaggio, assicurandosi che il campo **”Select file format”** sia impostato su **”interleaved files”**; infatti il file importato è unico e contiene tutte le coppie di read in maniera consecutiva. Infine basterà selezionare il campo **”Output final assembly graph (contigs)?”** e deselectare il campo **”Output final assembly graph (scaffold)?”** e cliccare su **”Run Tool”** per dare inizio all’assemblaggio.

Le Figure 3.7, 3.8, 3.9 e 3.10 mostrano le schermate di impostazione del tool.

SPAdes genome assembler for regular and single-cell projects (Galaxy Version 3.12.0+galaxy1)

Tool Parameters

Single-cell?

No
This option is required for MDA (single-cell) data. (--sc)

Run only assembly? (without read error correction)

No
(--only-assembler)

Careful correction?

Yes
Tries to reduce number of mismatches and short indels. Also runs MismatchCorrector – a post processing tool, which uses BWA tool (comes with SPAdes). (--careful)

Automatically choose k-mer values

Yes
k-mer choices can be chosen by SPAdes instead of being entered manually

Figure 3.7: Schermata dei parametri dell’assemblatore SPAdes in Galaxy 1/4

SPAdes genome assembler for regular and single-cell projects
(Galaxy Version 3.12.0+galaxy1)

Coverage Cutoff

Off

Libraries are IonTorrent reads?

No

Libraries

It is not possible to specify only mate-pair libraries. Scaffolds are not produced if neither a paired-end nor a mate-pair library is provided.

1: Libraries

Library type *

Paired-end / Single reads

Orientation *

-> <- (fr)

Figure 3.8: Schermata dei parametri dell’assemblatore SPAdes in Galaxy 2/4

SPAdes genome assembler for regular and single-cell projects
(Galaxy Version 3.12.0+galaxy1)

Files

1: Files

Select file format

Interleaved files

Interleaved paired reads *

35: SRR11528307 (fastq-dump)

FASTQ format

+ Insert Files

+ Insert Libraries

Figure 3.9: Schermata dei parametri dell’assemblatore SPAdes in Galaxy 3/4

SPAdes genome assembler for regular and single-cell projects
(Galaxy Version 3.12.0+galaxy1)

Output final assembly graph (contigs)?

Yes
Will output the final assembly graph (contigs) in fastg format for visualisation

Output final assembly graph with scaffolds?

No
Will output the final assembly graph with scaffold information in gfa format for visualisation

Additional Options

Email notification

No
Send an email notification when the job completes.

Run Tool

Figure 3.10: Schermata dei parametri dell’assemblatore SPAdes in Galaxy 4/4

3.4 Analisi dei risultati

L’assemblaggio così eseguito ha prodotto i seguenti file:



Figure 3.11: Schermata dei file di output dell’assemblaggio tramite SPAdes in Galaxy

In particolare, il file **contigs** mostra nel dettaglio il risultato dell’assemblaggio, mentre il file **contig stats** permette di visualizzare una tabella che ne sintetizza gli attributi principali (fig. 3.12).

name	length	coverage
#name	length	coverage
NODE_1	29600	2807.600820
NODE_2	147	27.500000

Figure 3.12: Schermata di visualizzazione del file "contig stats" prodotto da SPAdes in Galaxy

Domanda 6: *Quanti contigs ha prodotto l'assemblaggio? Che lunghezza hanno? Cosa pensi si intenda per "coverage" in questo contesto?*

Osservando la tabella riportata, si nota che sono stati generati due contigs, di lunghezza pari rispettivamente a 29600 e 147 e con corrispondente coverage uguale a 2807.600820 e 27.500000. Nel contesto presente, quest'ultima dovrebbe corrispondere alla coverage relativa ai k-mer usati da SPAdes per l'assemblaggio, indicando il numero di occorrenze di ogni k-mer del contig di appartenenza all'interno delle reads.

Come risultato dell'assemblaggio è stato anche prodotto l'assembly graph. Per visualizzare questo grafo, occorre utilizzare il tool Bandage, cliccando su "**Bandage Image**", sotto "**Assembly**", selezionare poi nel menu a tendina relativo al campo "**Graphical Fragment Assembly**" l'assembly graph prodotto, e cliccare su "**Run Tool**".

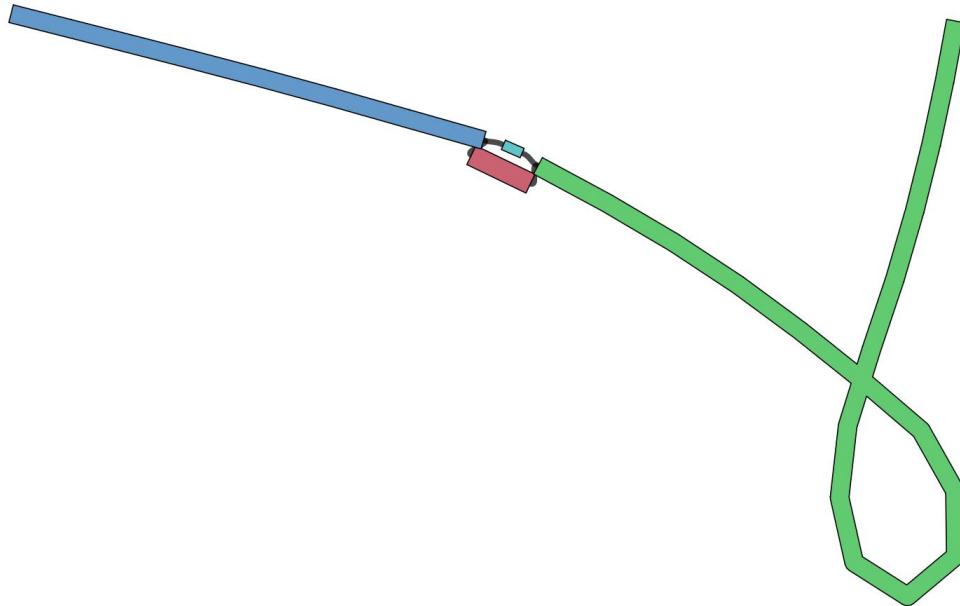


Figure 3.13: Assembly Graph prodotto dall'assemblaggio del genoma attraverso il tool SPAdes in Gslaxy

4 Allineamento

In questo capitolo sarà descritto il processo di allineamento di due genomi, quello del Sars-CoV-2 e quello del Sars-CoV dell'epidemia del 2003 per verificare quanto questi differiscano tra loro. Anche in questo caso saranno presentate le risposte alle domande di Phillip Compeau.

Per allineare i due genomi verrà utilizzato l'algoritmo *Needleman-Wunsch per l'allineamento globale* descritto nel capitolo 1. Prima di tutto occorre reperire il genoma del Sars-CoV e per farlo di attinge nuovamente alla piattaforma web del National Center for Biotechnology Information, direttamente tramite Galaxy. Cliccando **"NCBI Accession Download"**, sotto **"Get Data"**, basterà poi selezionare la voce **"Direct Entry"** nel menu a tendina **"Select source for IDs"**, specificare l'ID *NC_004718.3* nel campo **"ID List"** e premere **"Run Tool"**.

Si può dunque procedere con l'allineamento: cliccando dapprima su **"EMBOSS"**, sotto **"Genomics Toolkits"** e poi su **needle**. A questo punto bisognerà selezionare il file dei contigs prodotto da SPAdes nel campo **"Sequence 1"** e il file del genoma appena importato nel campo **"Sequence 2"** e cliccare ancora **"Run Tool"** (fig.4.1). Occorre precisare che sarà utilizzato soltanto il contig più lungo.



Figure 4.1: Schermata dei parametri di allineamento del tool Needle in Galaxy

Una volta terminata l'esecuzione, il tool avrà generato un unico file di output che mostra le sequenze allineate e una sintesi degli attributi dell'allineamento (fig. 4.2). Ogni *match* è rappresentato dal simbolo "|", mentre ogni *mismatch* dal simbolo "."; i gap invece non sono rappresentati da alcun simbolo.

```
#####
# Program: needle
# Rundate: Sun 11 Feb 2024 17:50:37
# Commandline: needle
#   -asequence /mnt/user-data-volA/data11/b/d/a/dataset_bdab2215-081a-4f0a-a3b6-a87d9d7e334e.dat
#   -bsequence /mnt/user-data-volA/data11/8/7/e/dataset_87e8f1c1-9744-498a-a5e5-607574e77ed3.dat
#   -outfile /mnt/tmp/job_working_directory/008/177/8177465/outputs/dataset_6a10cfda-34a1-4bae-
98d6-53e8cceeb25b7.dat
#     -gapopen 10.0
#     -gapextend 0.5
#     -brief yes
#     -aformat3 srspair
#     -auto
# Align_format: srspair
# Report_file: /mnt/tmp/job_working_directory/008/177/8177465/outputs/dataset_6a10cfda-34a1-4bae-
98d6-53e8cceeb25b7.dat
#####
=====
#
# Aligned_sequences: 2
# 1: NODE_1_length_29600_cov_2807.600820
# 2: NC_004718.3
# Matrix: EDNAFULL
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 30448
# Identity: 23875/30448 (78.4%)
# Similarity: 23875/30448 (78.4%)
# Gaps: 1545/30448 ( 5.1%)
# Score: 94510.0
#
=====
```

Figure 4.2: Schermata di visualizzazione del file dell'allineamento ottenuto con Needle in Galaxy

Domanda 7: *Quanti simboli risultano allineati dall'allineamento? Quanti gap ci sono?*

Osservando il valore degli attributi **Identity** e **Gaps**, si può affermare che risultano allineati 23875 simboli e ci sono 1545 gap.

Domanda 8: Scorrendo l'allineamento, noti qualche regione che appare più variabile di altre?

Sì, esaminando visivamente l'allineamento emergono regioni più variabili, ovvero caratterizzate da un numero maggiore di *mismatch* e *indel*. Un esempio è riportato in Figura 4.3.

Figure 4.3: Una delle regione maggiormente variabili nell'allineamento

5 Annotazione

Infine, ci occuperemo dell'annotazione del genoma, identificando i geni putativi e confrontandoli poi con un database di geni noti per individuare quelli che si allineano meglio. Anche in questo caso saranno presentate le risposte alle ultime domande di Phillip Compeau.

Cliccando su **”Annotation”**, sotto **”Genomics Analysis”** occorre cercare e selezionare il tool Prokka. Si dovrà poi selezionare nel menu a tendina relativo a **”Contigs to annotate”** il file dei contigs ottenuto tramite SPAdes (fig. 5.1), scegliere l’opzione **”Viruses”** nel campo **”Kingdom”** e cliccare su **”Run Tool”** (fig. 5.2) per dare inizio al processo. Prokka utilizzerà soltanto il contig più lungo, in quanto non supporta sequenze di lunghezza inferiore a 200.

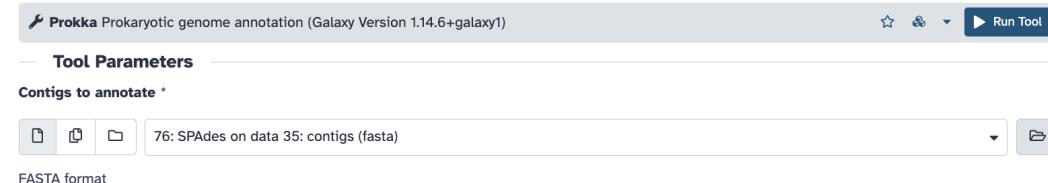


Figure 5.1: Schermata dei parametri del tool Prokka in Galaxy 1/2

Prokka Prokaryotic genome annotation (Galaxy Version 1.14.6+galaxy1) Run Tool

(--strain)

Plasmid name or identifier - optional

(--plasmid)

Kingdom

Viruses ▼

(--kingdom)

Figure 5.2: Schermata dei parametri del tool Prokka in Galaxy 2/2

L'annotazione del genoma così eseguita ha prodotto i seguenti file di output:



Figure 5.3: Schermata dei file di output dell'annotazione tramite Prokka in Galaxy

Tra questi, il file .gff è quello più interessante in quanto contiene le informazioni relative alle regioni identificate da Prokka come geni putativi (fig. 5.4).

Seqid	Source	Type	Start	End	Score	Strand	Phase	Attributes
##gff-version 3								
##sequence-region NODE_1_length_29600_cov_2807.600820 129600								
NODE_1_length_29600_cov_2807.600820	Prodigal:002006	CDS	118	13335	.	+		⑨ ID=AMFCHINL_00001;Name=1a;inference=ab initio prediction:Prodigal:002006,similar to AA sequence:UniProtKB:P0C6U8;locus_tag=AMFCHINL_00001;product=Replicase polyprotein 1a
NODE_1_length_29600_cov_2807.600820	Prodigal:002006	CDS	13620	21407	.	+		⑨ ID=AMFCHINL_00002;Name=rep;gene=rep;inference=ab initio prediction:Prodigal:002006,similar to AA sequence:UniProtKB:P0C6X7;locus_tag=AMFCHINL_00002;product=Replicase polyprotein 1ab
NODE_1_length_29600_cov_2807.600820	Prodigal:002006	CDS	21415	25236	.	+		⑨ ID=AMFCHINL_00003;Name=S;gene=S;inference=ab initio prediction:Prodigal:002006,similar to AA sequence:UniProtKB:P59594;locus_tag=AMFCHINL_00003;product=Spike glycoprotein
NODE_1_length_29600_cov_2807.600820	Prodigal:002006	CDS	25245	26072	.	+		⑨ ID=AMFCHINL_00004;Name=3agene=3a;inference=ab initio prediction:Prodigal:002006,similar to AA sequence:UniProtKB:P59632;locus_tag=AMFCHINL_00004;product=Protein 3a
NODE_1_length_29600_cov_2807.600820	Prodigal:002006	CDS	26375	27043	.	+		⑨ ID=AMFCHINL_00005;Name=M;gene=M;inference=ab initio prediction:Prodigal:002006,similar to AA sequence:UniProtKB:P59596;locus_tag=AMFCHINL_00005;product=Membrane protein
NODE_1_length_29600_cov_2807.600820	Prodigal:002006	CDS	27054	27239	.	+		⑨ ID=AMFCHINL_00006;inference=ab initio prediction:Prodigal:002006;locus_tag=AMFCHINL_00006;product=hypothetical protein
NODE_1_length_29600_cov_2807.600820	Prodigal:002006	CDS	27246	27611	.	+		⑨ ID=AMFCHINL_00007;Name=7agene=7a;inference=ab initio prediction:Prodigal:002006,similar to AA sequence:UniProtKB:P59635;locus_tag=AMFCHINL_00007;product=Protein 7a
NODE_1_length_29600_cov_2807.600820	Prodigal:002006	CDS	27746	28111	.	+		⑨ ID=AMFCHINL_00008;inference=ab initio prediction:Prodigal:002006;locus_tag=AMFCHINL_00008;product=hypothetical protein
NODE_1_length_29600_cov_2807.600820	Prodigal:002006	CDS	28126	29385	.	+		⑨ ID=AMFCHINL_00009;Name=N;gene=N;inference=ab initio prediction:Prodigal:002006,similar to AA sequence:UniProtKB:P59595;locus_tag=AMFCHINL_00009;product=Nucleoprotein

Figure 5.4: Schermata del file .gff generato come output dell’annotazione tramite Prokka in Galaxy

Per visualizzare il risultato dell’annotazione in maniera più significativa dal punto di vista biologico utilizzeremo il tool **JBrowse**, una piattaforma web dinamica e interattiva per la visualizzazione e l’analisi di genomi. Per trovarlo in Galaxy, basterà cliccare su **”Graph/Display Data”**, sotto **”Statistics and Visualization”** e selezionarlo nell’elenco. Dopodichè sarà necessario effettuare i seguenti passi:

1. Selezionare l’opzione **”Use a genome from history”** nel menu a tendina denominato **”Reference genome to display”** e selezionare il file .fna generato da Prokka in **”Select the reference genome”**;
2. Cliccare su **”Insert Track Group”**;
3. Cliccare su **”Insert Annotation Track”**;
4. Selezionare il file .gff generato da Prokka in **”GFF/GFF3/BED Track Data”**;
5. Cliccare su **”Run Tool”** per visualizzare l’annotazione.

La selezione dei parametri è illustrata anche in Figura 5.5 e 5.6.

Figure 5.5: Schermata dei parametri del tool JBrowse in Galaxy 1/2

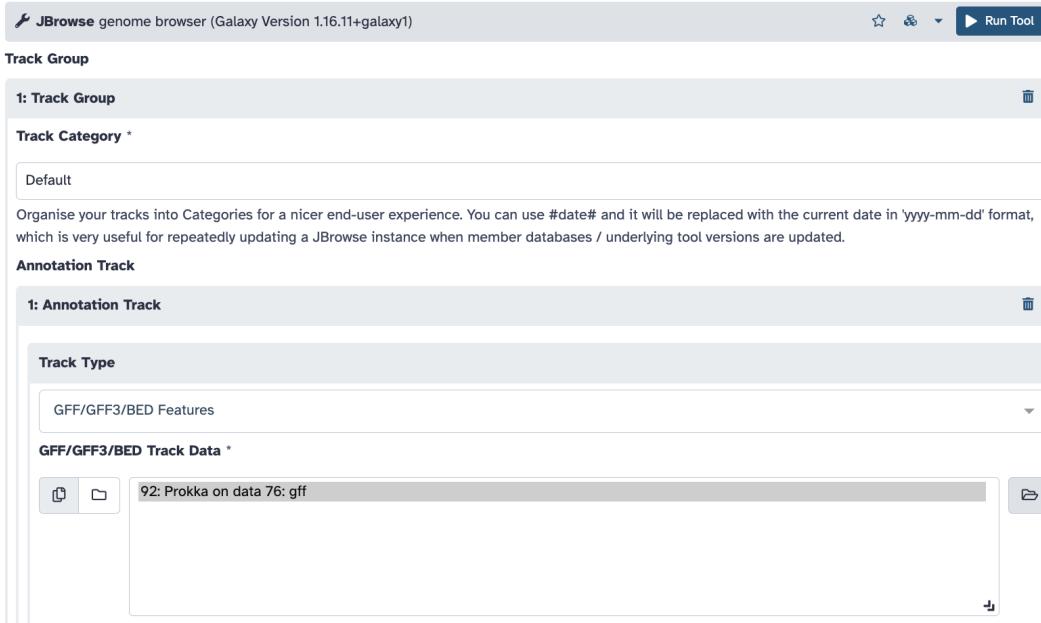


Figure 5.6: Schermata dei parametri del tool JBrowse in Galaxy 2/2

Il file generato da JBrowse è un HTML che, una volta aperto, può essere visualizzato direttamente all'interno di Galaxy. Spuntando il campo **"Prokka on data XX: gff"** viene mostrata l'annotazione del genoma del Sars-CoV-2 (fig. 5.7). Si può notare che tutte le frecce puntano nella stessa direzione e ciò indica che i geni si trovano tutti sullo stesso filamento del genoma, come aspettato trattandosi di un virus a RNA (il suo genoma ha un unico filamento). Inoltre, cliccando sui singoli geni è possibile ottenere informazioni come la lunghezza o le eventuali proteine del database con cui è stata riscontrata una somiglianza: cliccando, ad esempio, sulla prima proteina (*"Replicase polyprotein 1a"*) si può notare che è stato scoperta una somiglianza con la proteina con l'Uniprot ID pari a P0C6U8, come mostrato in Figura 5.8, e quest'ultima corrisponde allo stesso gene del Sars-CoV.

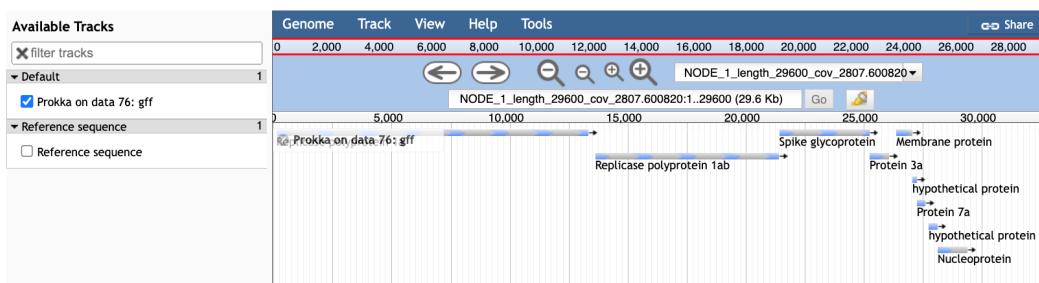


Figure 5.7: Schermata di visualizzazione dell'annotazione del genoma con JBrowse in Galaxy

The screenshot shows a JBrowse interface for a protein named "Replicase polyprotein 1a". The top navigation bar includes File, Track, View, Help, and Tools. A zoomed-in genomic track labeled "CDS 1a" is visible at the top. Below the track, the "Primary Data" section displays the following details:

- Name:** Replicase polyprotein 1a
- Type:** CDS
- Position:** NODE_1_length_29600_cov_2807.600820:118..13335 (+ strand)
- Length:** 13,218 bp

The "Attributes" section lists various identifiers and annotations:

- gene: 1a
- id: AMFCHINL_00001
- inference: ab initio prediction:Prodigal:002006 similar to AA sequence:UniProtKB:P0C6U8
- locus_tag: AMFCHINL_00001
- phase: 0
- product: Replicase polyprotein 1a
- seq_id: NODE_1_length_29600_cov_2807.600820
- source: Prodigal:002006
- uniqueID: offset-125133

The "Region sequence" section shows the DNA sequence for the CDS region:

```

>NODE_1_length_29600_cov_2807.600820
NODE_1_length_29600_cov_2807.600820:118..13335 (+ strand)
class=CDS length=13218
ATGGAGAGCTTGTCCCTGTTCAACAGAGAAAACACAGTCCAACACTCAGTTGCCTGTTTACAG
GTTGGCGACCTGCTCTAGTGCGTTGGAGACTCCGTGGAGGAGTCTTATCAGAGGCACGTCAA
CATCTAAAGATGGCACITGTGCGTTAGTAGAAGTTGAAAAAGGCCCTTTGCCCTAACATTGAACAG
CCCTATGTGTTCATCAAACGTTGGATGCTCGAACATGCCACCTCATGGTCATGTTATGGTTGAGCTG

```

Figure 5.8: Schermata delle informazioni sulla proteina "Replicase polyprotein 1a" in JBrowse in Galaxy

Nella schermata di JBrowse, selezionando inoltre anche il campo **"Reference sequence"** verrà mostrata in parallelo la sequenza del genoma di riferimento:

The screenshot shows a JBrowse interface with the "Available Tracks" sidebar expanded. It includes sections for "Default" tracks (Prokka on data 76: gff) and "Reference sequence" tracks (Prokka on data 76: gff). The main view displays a genomic track with a blue header and a protein sequence track below it. The protein sequence is labeled "Protein 7a". The sequence is color-coded by amino acid (e.g., R=red, H=green, K=blue, etc.). Above the protein sequence, a red box highlights a specific region of the sequence.

Figure 5.9: Schermata di visualizzazione dell'annotazione del genoma e della sequenza di riferimento con JBrowse in Galaxy

Domanda 9: Quanti geni sono stati identificati come putativi? Qual è il più lungo e quale il più corto?

Sono stati individuati 9 geni putativi. Per determinare il più lungo e il più corto è possibile consultare il file .tsv, in cui sono raggruppati i geni e le lunghezze corrispondenti. (fig.)

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
locus_tag	ftype	length_bp	gene	EC_number	COG	product
AMFCHINL_00001	CDS	13218	1a			Replicase polyprotein 1a
AMFCHINL_00002	CDS	7788	rep			Replicase polyprotein 1ab
AMFCHINL_00003	CDS	3822	S			Spike glycoprotein
AMFCHINL_00004	CDS	828	3a			Protein 3a
AMFCHINL_00005	CDS	669	M			Membrane protein
AMFCHINL_00006	CDS	186				hypothetical protein
AMFCHINL_00007	CDS	366	7a			Protein 7a
AMFCHINL_00008	CDS	366				hypothetical protein
AMFCHINL_00009	CDS	1260	N			Nucleoprotein

Figure 5.10: Schermata del file .tsv generato come output dell'annotazione tramite Prokka in Galaxy

Dunque il più lungo è la proteina comune al Sars-CoV, *Replicase polyprotein 1a*, il più corto noto è *Membrane protein*, mentre quello non noto è etichettato "*hypothetical protein*".

Domanda 10: Perchè pensi che due geni siano stati etichettati come "ipotetiche proteine"?

Tale etichetta probabilmente suggerisce che non sia stata riscontrata una significativa somiglianza con alcuna sequenza proteica nel database considerato dal tool Prokka.

6 Conclusioni

Nel corso dello svolgimento del progetto è stato eseguito con successo l'assemblaggio e l'annotazione del genoma del Sars-CoV-2, permettendone la comprensione della struttura e delle caratteristiche. Il virus è stato inoltre allineato con il virus responsabile dell'epidemia del 2003, il Sars-CoV, evidenziando le regioni più variabili e quelle più conservative.

L'utilizzo dei tool impiegati e gli output ottenuti possono essere ancora esplorati per ulteriori analisi. In particolare, i geni individuati possono essere investigati singolarmente per esaminare e comprendere le variazioni del virus riscontrate negli ultimi anni, durante la sua diffusione a livello globale.

References

- [1] M. della Salute. Cosa sono sars-cov-2 e covid-19. [Online]. Available: <https://www.salute.gov.it/portale/nuovocoronavirus/dettaglioFaqNuovoCoronavirus.jsp?lingua=italiano&id=257>
- [2] P. Compeau. Sars-cov-2 software assignment: Genome assembly and annotation. [Online]. Available: <https://compeau.cbd.cmu.edu/online-education/sars-cov-2-software-assignments/covid-19-genome-assembly-assignment/>
- [3] A. Bankevich, S. Nurk, D. Antipov, A. A. Gurevich, M. Dvorkin, A. S. Kulikov, V. M. Lesin, S. I. Nikolenko, S. Pham, A. D. Prjibelski, A. V. Pyshkin, A. V. Sirotnik, N. Vyahhi, G. Tesler, M. A. Alekseyev, and P. A. Pevzner, “Spades: A new genome assembly algorithm and its applications to single-cell sequencing,” *Journal of Computational Biology*, vol. 19, no. 5, 2012.
- [4] R. R. Wick, M. B. Schultz, J. Zobel, and K. E. Holt, “Bandage: interactive visualization of de novo genome assemblies,” *Bioinformatics*, vol. 31, no. 20, 06 2015.
- [5] A. Bleasby. (1999) Needle documentation. [Online]. Available: <https://galaxy-iuc.github.io/emboss-5.0-docs/needle.html>
- [6] T. Seemann, “Prokka: rapid prokaryotic genome annotation,” *Bioinformatics*, vol. 30, no. 14, 03 2014.
- [7] C. G. L. P. e. a. Hyatt, D., “Prodigal: prokaryotic gene recognition and translation initiation site identification,” *BMC Bioinformatics*, vol. 11, no. 119, 03 2010.
- [8] M. T. McGinnis S, “BLAST: at the core of a powerful and diverse set of sequence analysis tools,” *Nucleic Acids Res*, vol. 32, 07 2004.