UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

"IL GRAFO DI DE BRUIJN E SUA APPLICAZIONE ALL'ASSEMBLY DI GENOMI"

Relatore: Prof.ssa CINZIA PIZZI

Laureando: GUAN ENCI

ANNO ACCADEMICO 2023 – 2024

Data di laurea 19/07/2024

Abstract

Il sequenziamento del genoma è una tecnologia fondamentale che ha rivoluzionato la biologia molecolare, consentendo lo studio dettagliato dei genomi di organismi complessi. Dal completamento del sequenziamento del primo genoma umano nel 2003, le tecnologie di sequenziamento hanno continuato a evolversi, permettendo progressi significativi nella genomica. L'assemblaggio dei genomi è una fase cruciale di questo processo, in quanto consente di ricostruire sequenze genomiche complete a partire dai frammenti di DNA ottenuti dalle moderne tecnologie di sequenziamento. Questo elaborato fornisce in primo luogo una breve descrizione sulle procedure di sequenziamento e assemblaggio genomico, e successivamente una focalizzazione sul grafo di De Bruijn e l'implementazione pratica utilizzando l'assemblatore SPAdes.

Indice

1 Introduzione

La sequenza del DNA rappresenta l'ordine preciso dei nucleotidi – adenina (A), timina (T), citosina (C) e guanina (G) – lungo una molecola di DNA. Questa sequenza è fondamentale perché contiene l'informazione genetica necessaria per lo sviluppo, il funzionamento e la riproduzione di tutti gli organismi viventi. Decifrare questa sequenza, attraverso un processo noto come “DNA Sequencing”, è essenziale per comprendere i meccanismi biologici di base, identificare le cause genetiche delle malattie e sviluppare terapie mirate.

Il "genoma assembly" è un processo computazionale in cui i frammenti di DNA sequenziati vengono assemblati per formare la sequenza completa del genoma di un organismo. Poiché le tecniche di sequenziamento attuali possono gestire solo frammenti di DNA di breve lunghezza alla volta, è necessario suddividere l'intero genoma in piccoli frammenti chiamati anche “read” e sequenziarli individualmente. Dopo questa fase possiamo capire l’ordine dei nucleotidi che costituiscono ogni frammento dell'organismo e ricostruire la sequenza originale del genoma tramite gli algoritmi basati sulle sovrapposizioni tra le letture. La Figura 1 [13] mostra una panoramica sul sequenziamento ed assemblaggio di un genome.

Le procedure di assemblaggio genomico sono complesse e presentano numerose sfide, specialmente con le tecnologie di “Next Generation Sequencing” (NGS), che producono letture corte e numerose. Per affrontare le sfide nell’assemblaggio si utilizzano i grafi di assemblaggio, che sono efficaci per rappresentare le letture di sequenziamento e le loro sovrapposizioni, facilitando l'identificazione e la gestione delle ripetizioni genomiche.

L'obiettivo principale di questa tesi è fornire una panoramica delle tecniche di sequenziamento e di assemblaggio genomico, con un focus particolare sull'uso del grafo di De Bruijn e un’applicazione di questo metodo utilizzando l'assemblatore SPAdes con aiuto dello strumento di valutazione QUAST.

La tesi è strutturata come segue: nel secondo capitolo vengono introdotte le tecniche e le pipeline utilizzate nell'assemblaggio genomico, comprese le tecnologie di sequenziamento di seconda e terza generazione e le differenze tra de novo assembly e

reference-based assembly. Nel terzo capitolo viene approfondita la definizione del grafo di De Bruijn , i metodi di assemblaggio basati su di esso e le principali sfide presenti. Il quarto capitolo introduce l'assemblatore SPAdes e le metriche utilizzate nelle statistiche QUAST per la valutazione della qualità. Nel quinto capitolo vengono riportati i risultati ottenuti dall'assemblaggio tramite SPAdes e i commenti su di essi.

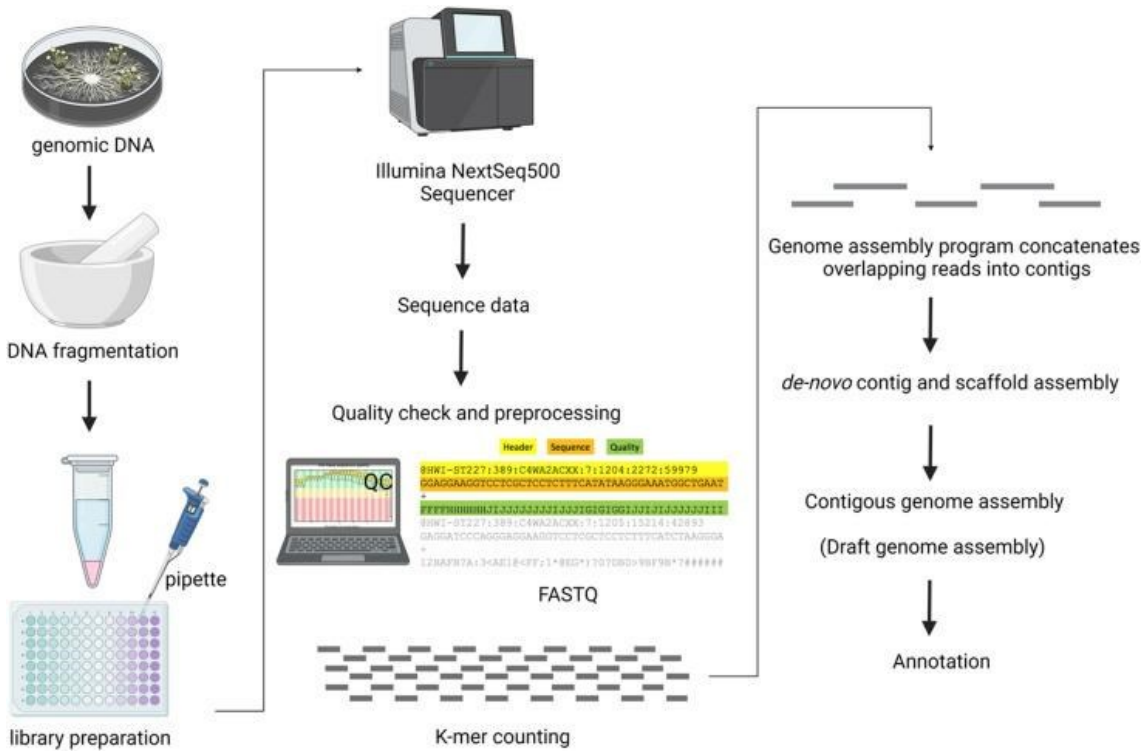


Figura 1: Rappresentazione schematica del processo di sequenziamento e assemblaggio [13]

2 Genome assembly

L’assemblaggio genomico è un passo cruciale nel sequenziamento del DNA fornendo una rappresentazione completa del genoma di un organismo. Tuttavia, con le tecnologie attuali non è possibile leggere un intero genoma, quindi è richiesto prima una procedura di sequenziamento che frammenta il genoma in tanti piccoli frammenti che poi vengono assemblati sfruttando le regioni condivise tra le letture.

2.1 Tecnologie di sequenziamento

Le tecnologie di sequenziamento sviluppate negli anni sono caratterizzate in tre generazioni. Il primo metodo, introdotto negli anni Settanta da Sanger [14], utilizzava ddNTP (dideossinucleotidi fluorescenti) per interrompere la sintesi della catena. I filamenti risultanti venivano poi sottoposti a elettroforesi capillare per determinare la sequenza. Sebbene il metodo di Sanger abbia decodificato il primo genoma umano ed è stato il più utilizzato per quasi 30 anni, era lento e costoso, rendendolo poco adatto per genomi complessi.

Nel 2005 sono state introdotte le tecnologie di "*next generation sequencing*" utilizzate per la lettura di short read, caratterizzate da quattro caratteristiche comuni rispetto al Sanger:

1) Capacità di generare milioni di short read (250-800 bps) in parallelo

2) Elevata velocità di sequenziamento

3) Basso costo

4) Non richiede l'uso di elettroforesi. Il flusso di lavoro di base per le piattaforme di seconda generazione include la preparazione e l'amplificazione di librerie (preparate da campioni di DNA/RNA), l'espansione clonale che produce le copie di ciascun frammento, il sequenziamento e l'analisi.

Ogni piattaforma NGS ha le sue specificità, la procedura comune è che la preparazione di una libreria NGS inizia con la frammentazione del materiale di partenza, successivamente gli adattatori di sequenza vengono collegati ai frammenti per consentire l'arricchimento di tali frammenti.

Le due aziende più note di piattaforme di sequenziamento di seconda generazione sono Illumina e ThermoFisher che si basano entrambi sull'approccio *"sequencing by synthesis (SBS)"*.

Il processo di sequenziamento di Illumina inizia con la frammentazione del DNA (Figura 2 [15]) in segmenti di solito lunghi 200-500 pb (paia di basi), ai quali vengono aggiunti gli adattatori alle estremità. Gli adattatori sono brevi sequenze di DNA sintetico che contengono sequenze complementari agli oligonucleotidi sulla superficie della flow cell. Questo permette ai frammenti di DNA di legarsi alla flow cell, una piastra di vetro ricoperta di oligonucleotidi.

Ogni frammento di DNA legato si piega formando un ponte a doppio filamento. Successivamente, i filamenti vengono denaturati, separandoli in singoli filamenti tramite riscaldamento o trattamenti chimici. Questo processo di denaturazione consente di ottenere modelli a singolo filamento, che sono poi ancorati alla flow cell. Tale ciclo si ripete continuamente, generando milioni di cluster densi di DNA a doppio filamento in ogni canale della flow cell (Figura 3 [15]).

Il sequenziamento avviene aggiungendo singoli deossinucleotidi trifosfato (dNTP) complementari al modello di DNA sulla flow cell, fungendo da "reversibile terminator" (Figura 4 [15]). I coloranti fluorescenti sono identificati tramite eccitazione laser e imaging. Successivamente, i coloranti vengono rimossi per consentire il ciclo successivo di incorporazione.

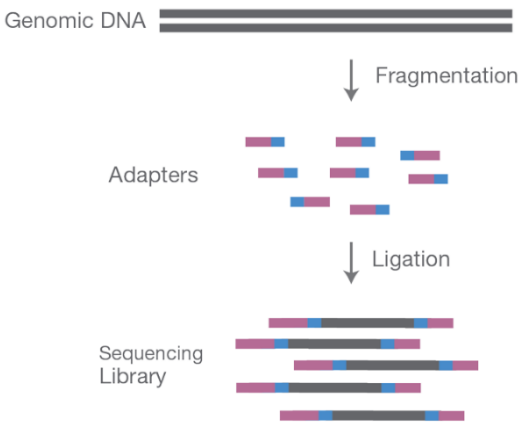


Figura 2: Preparazione della libreria NGS [15]

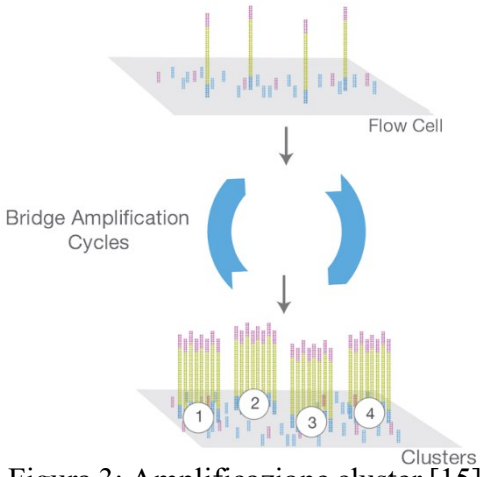


Figura 3: Amplificazione cluster [15]

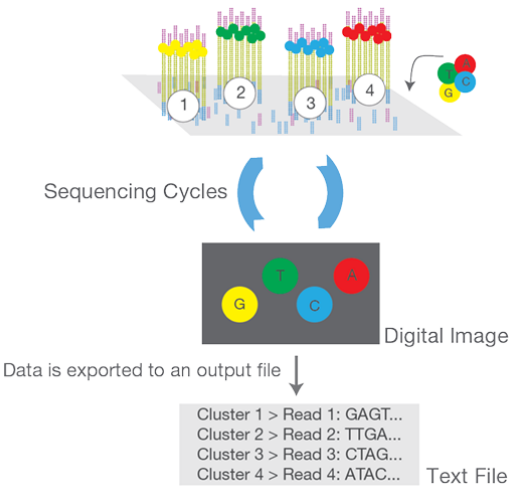


Figura 4: Sequenziamento tramite nucleotidi fluorescenti [15]

Le piattaforme di ThermoFisher si basano sulla tecnologia "Ion Torrent", che riesce a rilevare le variazioni di pH causate dal rilascio di ioni idrogeno (H^+) durante la sintesi del DNA. Nella tecnologia Ion Torrent, ogni frammento viene attaccato alla propria "bead", delle microparticelle sferiche fatte di materiale polimerico, con dimensioni tra 1 e 10 mm. I frammenti vengono clonati fino a coprire le bead tramite PCR in emulsione. Questo processo automatizzato copre milioni di bead con milioni di frammenti diversi. Ogni bead viene depositata in un micro-pozzetto sul chip dell'array del sensore a semiconduttore, denominato chip CMOS (complementary metal-oxide-semiconductor). Ogni volta che la polimerasi incorpora un nucleotide complementare nella catena in crescita, viene rilasciato uno ione idrogeno, modificando il pH della soluzione nel pozzetto. Uno strato sensibile agli ioni sotto il pozzetto misura la variazione del pH e la converte in tensione. Questa variazione di tensione viene registrata e utilizzata per la chiamata delle basi.

La principale fonte di errore di Ion Torrent si verifica durante l'allungamento di modelli omopolimerici, che sono sequenze di DNA o RNA costituite da un singolo tipo di nucleotide ripetuto più volte. Ad esempio, una sequenza composta esclusivamente da adenine (A), come AAAAA, è un omopolimero di adenina. Durante il sequenziamento di un modello omopolimerico, si verificheranno molte incorporazioni della stessa base su ciascun filamento, generando il rilascio di una maggiore concentrazione di H+ in un singolo flusso. Tuttavia, la variazione di pH potrebbe avere un rapporto non lineare

rispetto al numero di nucleotidi incorporati; quindi, può essere difficile distinguere il numero esatto di nucleotidi.

Nella Tabella 1 [6] sono illustrate le caratteristiche di alcune delle piattaforme delle due aziende relative all'anno 2021. Possiamo osservare che le piattaforme di Illumina possono essere applicate a diverse tipologie di sequenziamento come WGS (whole genome sequencing), WES (whole exome sequencing) e TS (targeted sequencing). Le letture ottenute sono paired-end con lunghezza compresa tra 150 bp e 300 bp. Per quanto riguarda Ion Torrent, il campo di applicazione è più limitato e le letture sono generalmente single-end con lunghezza tra 200 bp e 400 bp. Il Phred score Q = -10*log10(P), dove P rappresenta la probabilità di errore di una base nucleotidica. Un punteggio di qualità di 30 significa che c'è una probabilità di errore di 1 su 1.000 (0,1%), ovvero una precisione del 99,9%.

È evidente che tutte le piattaforme di Illumina producono letture di alta qualità, poiché la percentuale di letture che soddisfano il Q30 è molto elevata (70% ~ 80%). Al contrario, le letture di Ion Torrent hanno una precisione mediamente superiore al 99%, corrispondente a un punteggio di qualità Q20.

Company	Illumina								ThermoFisher		
System Platform	iSeq	Miniseq	MiSeq	NextSeq550	NextSeq 1000&2000	NovaSeq 6000	MiSeqDx	NextSeq550 Dx	GeneStudio S5	Genexus	Ion PGM-Dx
Sequencing Principle	Sequence by Synthesis										
Detection	Fluorescent								Ion		
Applications	Small WGS, TS, Small RNA sequencing		Small WGS, TS, ChIP-Seq, Small RNA sequencing	TS, small WGS, exome and transcriptome sequencing		TS, WGS, WES, transcriptome and epigenome sequencing	TS, Small WGS	TS, exome and transcriptome sequencing	TS, epigenetic, exome, and transciptome sequencing	TS	TS
Maximum Read length (bases)	2 × 150 bp		2 × 300 bp	2 × 150 bp	2 × 150 bp	2 × 250 bp	2× 300 bp	2× 150 bp	600 bp	400 bp	200 bp
Flow cells/device	1					2	1				
Output (per flow cell)	1.2 Gb	7.5 Gb	15 Gb	120 Gb	330 Gb	3000 Gb	≥5 Gb	≥90 Gb	15 Gb	24 Gb	1 Gb
Sequencing Run time	9.5-19 hr	5-24 hr	4-56 hr	11-29 hr	11-48 hr	13-44 hr	24 hr	≤35 hr	4.5-21.5 hr	14-31 hr	4.4 hr
Accuracy/Quality Score	Q30≥ 80% (2 × 150 bp)		Q30≥ 70% (2 × 300 bp)	Q30≥ 75% (2 × 150 bp)		Q30≥ 75% (2 × 250 bp)	Accuracy >99.66%, Q30> 80%	Accuracy ≥99.98%, Q30≥75%	Accuracy ≥99%	Accuracy ≥99%	Accuracy ≥99%
Equipment Cost (USD)	$19,900	$49,500	$99,000	$275,000	$335,000	on request					
Dimensions	42.5 × 30.5 × 33 cm	45.6 × 48 × 51.8 cm	68.6 × 56.5 × 52.3 cm	53.3 × 63.5 × 58.4 cm	92 × 120 × 118 cm	80 × 94.5 × 165.6 cm	68.6 × 56.5 × 52.3 cm	54 × 69 × 58 cm	54.2 x 80.6 x 50.9 cm	81.5 × 106.5 × 167.8 cm	61 × 53 × 51 cm
Weight	15.9 kg	45 kg	57.2 kg	83 kg	141 kg	481 kg	54.4 kg	84.4 kg	63.5 kg	204.1 kg	30 kg
Advantages	High accuracy with good depth of coverage										
Disadvantages	Long running time with phasing difficulties										

WGS = whole genome sequencing, WES = whole exome sequencing, TS = targeted sequencing

Tabella 1: Piattaforme di sequenziamento a lettura breve e varie caratteristiche [16]

Le tecnologie di seconda generazione generalmente richiedono l'amplificazione tramite PCR, una procedura costosa e lunga. Di conseguenza, anche il processo di assemblaggio diventa oneroso a causa dell'abbondanza di short read. Per affrontare tali problematiche, sono state introdotte le *tecnologie di terza generazione*, che eliminano la necessità di PCR e sono in grado di produrre letture lunghe. Un approccio ampiamente utilizzato è la tecnologia SMRT (Single Molecule Real-Time), che sintetizza il DNA in modo continuo. Durante il processo, viene rilevato il tempo trascorso tra l'incorporazione di ciascun nucleotide, consentendo la registrazione in tempo reale della sequenza di DNA.

Ci sono due strumenti che adottano questo approccio:

1) Pacific BioScience: utilizza etichette fluorescenti per rilevare i nucleotidi in tempo reale, senza amplificarli. La struttura comprende celle SMRT, ognuna contenente una nanostruttura chiamata zero mode waveguides (ZMWs). Queste celle contengono una DNA polimerasi e il frammento target, riuscendo a rilevare il nucleotide aggiunto durante la sintesi in tempo reale. Lo strumento ha un tempo di preparazione di circa 4-6 ore, produce letture lunghe di circa 10 kbp, con un tasso di errore del 13% dovuto a inserzioni e cancellazioni.

2) Oxford Nanopore è un dispositivo mobile connesso al computer portatile tramite cavo USB 3.0. Il processo coinvolge il passaggio del DNA attraverso un nanoporo, misurando le variazioni di flusso ionico per identificare le basi nucleotidiche. Questa tecnologia ha il vantaggio di essere meno costosa e di dimensioni più ridotte. Il campione viene caricato sul dispositivo e i dati appaiono sul display in tempo reale senza necessità di attesa. Riesce a generare letture anche più lunghe di 150 kbp, con un tasso di errore di circa il 12%, di cui il 3% è dovuto al mismatch, il 4% all'inserimento e il 5% alla cancellazione [7].

2.2 De novo vs reference-based assembly

Nel De novo assembly, le sequenze genomiche vengono assemblate senza l'ausilio di un genoma di riferimento. Questo approccio è spesso utilizzato quando non è disponibile un genoma di riferimento o si lavora con organismi poco conosciuti. Le letture del DNA ottenute tramite tecniche di sequenziamento vengono sovrapposte e combinate per formare contig, sequenze continue di DNA. Questi contig vengono poi ulteriormente collegate e ordinate per produrre uno o più scaffolds, seguiti da fasi di rifinitura per correggere errori e migliorare la qualità dell'assemblaggio.

Nel Reference-based assembly, le sequenze lette vengono allineate o mappate su un genoma di riferimento già noto. Questo approccio è utile quando si dispone di un genoma di riferimento di alta qualità e si desidera ottenere la sequenza del genoma di un organismo correlato o della stessa specie. Il processo consiste nell'allineare le letture, producendo una copertura del genoma che mostra quali regioni sono coperte dalle letture e quali potrebbero avere lacune. Queste lacune vengono colmate utilizzando informazioni

dal genoma di riferimento, generando un assemblaggio completo del genoma dell'organismo in esame.

In questa tesi ci concentriamo sul De novo assembly poiché che utilizza il grafo di De Bruijn .

2.3 Genome assembly pipeline

Negli ultimi 30 anni sono stati sviluppati molti strumenti per l'assemblaggio del DNA, ognuno con strategie differenti, ma quasi tutti seguono la seguente pipeline:

2.3.1 Costruzione dei contig

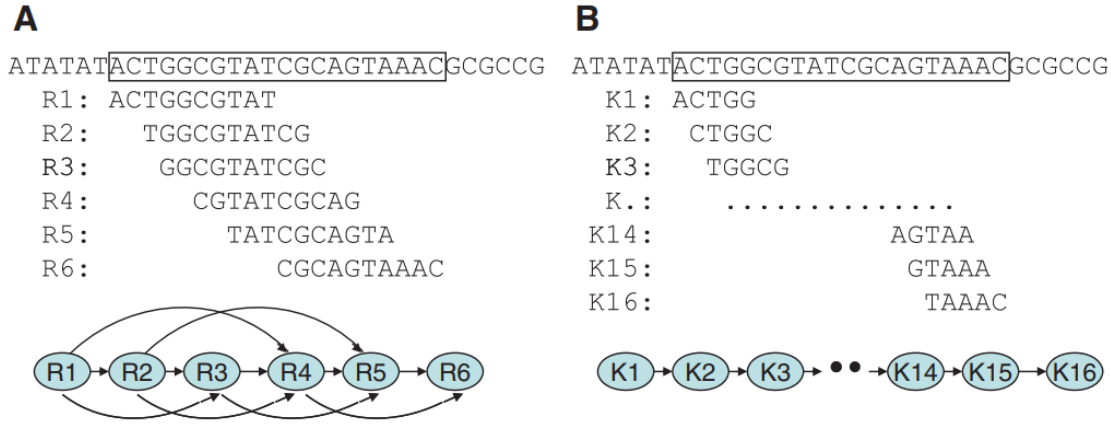
I contig sono sequenze continue di DNA abbastanza lunghe da rappresentare porzioni di genomi, ma spesso non coprono l'intera lunghezza dei cromosomi o dei genomi a causa della complessità dell'assemblaggio e della presenza di regioni ripetute o altre difficoltà. In De Novo assembly una volta costruito il grafo di assemblaggio, il percorso ottimale viene identificato nel grafo e i contig si ottengono trasformando inversamente il percorso ottimale nel grafo di De Bruijn in sequenze. Dopo aver corretto le letture dal sequenziamento, viene creato il grafo di assemblaggio. I grafi più utilizzati per rappresentare le relazioni di sovrapposizione tra le letture in De novo assembly sono “Overlap graph” e il “De Bruijn graph”.

Il metodo Overlap costruisce un grafo che rappresenta i nodi con le letture e assegna un ramo tra due nodi quando queste due si sovrappongono per una lunghezza maggiore di una lunghezza limite, il numero di nodi è uguale al numero di letture e aumenta linearmente con la profondità di sequenziamento, il numero di rami aumenterà su scala logaritmica. Nella Figura 5a [4], sono state generate 6 letture (R1–R6) di lunghezza 10 bp con il limite della lunghezza di sovrapposizione pari a 5 bp. Le letture sono ordinate lungo il genoma in base alla loro posizione iniziale e il grafo OLC ha la maggior parte dei nodi che presentava più di un arco in entrata o in uscita

Nel metodo di De Bruijn , ogni lettura viene suddivisa in sequenze della stessa lunghezza k, chiamati anche k-mer, che rappresentano i nodi del grafo e assegna un ramo tra due nodi quando questi due k-mer si sovrappongono per suffisso e prefisso, il numero di nodi e rami sono uguali alla dimensione del genoma e non dipendono alla profondità del sequenziamento. Nella Figura 5b [4], letture sono state suddivise in k -meri (K = 5),

ci sono in totale 16 k -meri diversi, la maggior parte dei quali si verifica in più di una lettura.

Considerando il consumo computazionale di tempo e memoria, l'algoritmo di sovrapposizione è più adatto per letture lunghe a bassa copertura, mentre l'algoritmo De Bruijn è più adatto per letture brevi ad alta copertura e soprattutto per assemblaggi di genomi di grandi dimensioni [4].



(a) Overlap graph (b) De Bruijn graph

Figura 5: Costruzione del grafo OLC e DBG utilizzando dati di esempio da una regione genomica di lunghezza 20 bp [4]

2.3.2 Scaffolding

Successivamente i contig vengono ordinati, orientati e collegati tramite due tipi di letture specifici, Paired-end Read e Mate-Pair Read.

Le letture pair-end sono coppie di sequenze ottenute dalle estremità opposte di un frammento di DNA, e permettono di determinare l'orientamento relativo dei due segmenti di DNA e la distanza approssimativa tra di essi. Rispetto alle letture single-end, le letture pair-end coprono frammenti di DNA più lunghi, il che aiuta a risolvere meglio regioni ripetute o complesse del genoma.

Le letture Mate-pair provengono da frammenti di DNA che sono stati "tagliati" e poi ricongiunti in laboratorio, producendo coppie di read provenienti da una distanza maggiore rispetto alle letture paired-end.

2.3.3 Gap filling

Dopo che i contig sono collegati, rimangono tra loro degli spazi chiamati “gaps” che vengono riempiti da altre letture indipendenti per completare assemblaggio. Scaffolding e gap filling possono essere eseguiti in modo iterativo fino a quando non vengono più collegati i contig o non vengono risolti ulteriori spazi vuoti.

Nella Figura 6 [5] sono mostrate i diversi passaggi di un De novo assembly, sovrapponendo le letture, i contig vengono assemblati da letture corte prima di scaffolding tramite letture di grandi inserti e i gap rimanenti vengono riempiti.

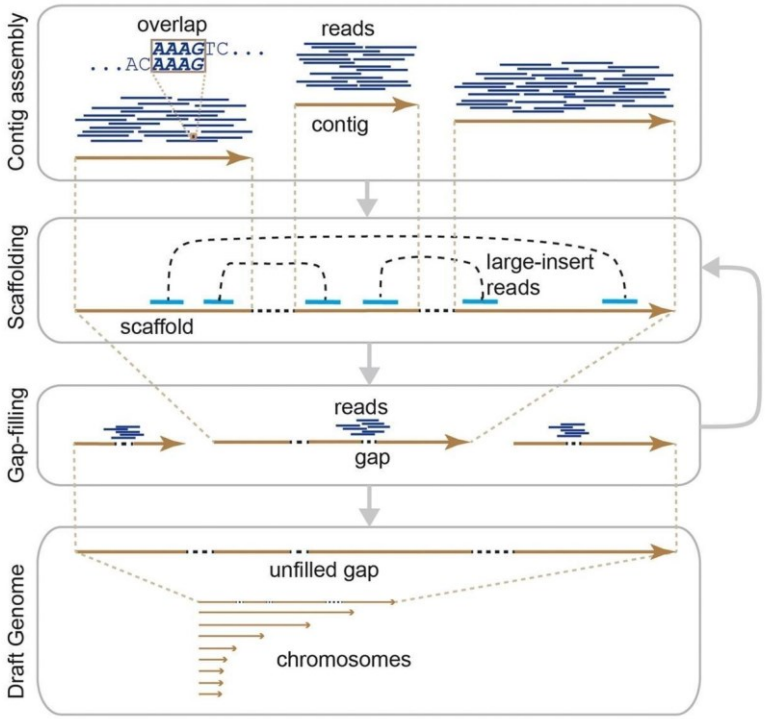


Figura 6: Flusso di lavoro di De novo assembly di un genoma intero [5]

3 De Bruijn graph

3.1 Definizione ed Origine

Nel 1946, il matematico olandese Nicolaas De Bruijn ha pubblicato un articolo intitolato *"A combinatorial problem"*, in cui utilizzando un approccio ispirato alla soluzione di Euler per il problema dei ponti di Königsberg [1]. Il problema consisteva nel determinare se fosse possibile per i cittadini di Königsberg attraversare tutti i ponti unA sola volta e tornare al punto di partenza senza dover attraversare alcun ponte due volte (Figura 7). De Bruijn sviluppò un metodo per costruire superstringhe circolari contenenti tutte le possibili sottostringhe di lunghezza k (k-meri) per un alfabeto dato.

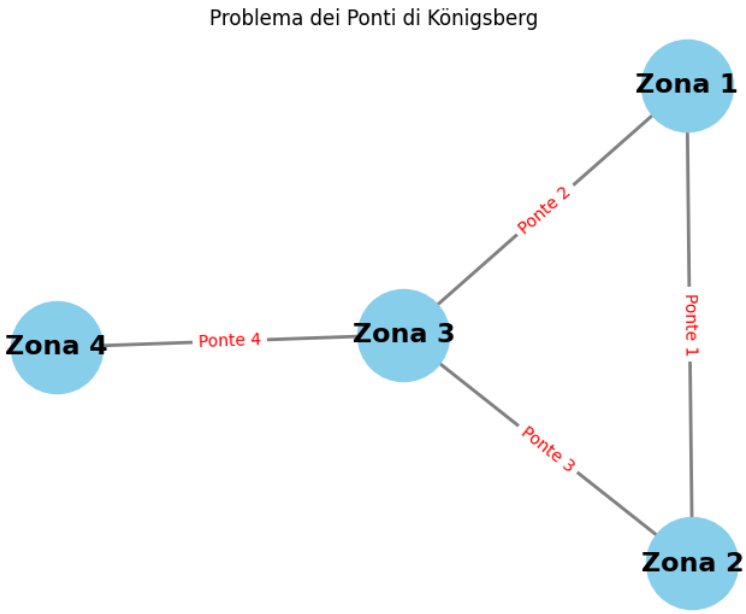


Figura 7: Una mappa della vecchia Königsberg, in cui ogni zona della città è rappresentato da un nodo, e i ponti da un ramo, fatto con Python

Un grafo di De Bruijn è un tipo di grafo orientato che rappresenta una sequenza di simboli in cui ogni sottostringa di lunghezza k è associata a un nodo nel grafo. Sia Σ un alfabeto di simboli e sia k un intero positivo, un grafo di De Bruijn B(Σ, k) è definito da insieme di nodi V = $\{v_1, v_2 \dots v_n\}$ che contiene tutte le sottostringhe di lunghezza k che possono essere create combinando i simboli dell'alfabeto Σ, e un insieme di Archi E= $\{ e_1, e_2 \dots e_m\}$, c'è un arco che va da un nodo A ad un nodo B se suffisso di lunghezza k-

1 di A è uguale al prefisso di B, ogni arco è etichettato con il simbolo aggiunto per passare da un nodo all'altro.

Nel contesto dell'assemblaggio genomico tutte le letture sono suddivise in (L-k+1) k -mers, dove L è la lunghezza della lettura e k è la dimensione del k -mer. I k -mer sono collegati mediante prefisso e suffisso sovrapposti (k -1) mer.

Esempio: sia k = 3 , e Σ = {A, C, G, T} l'alfabeto dato, i nodi saranno V= {AAA, AAC, AAG, AAT, CAA, CAC, ..., TTT}. Gli archi connetteranno i nodi in modo che ogni coppia di nodi consecutivi abbia una sovrapposizione di lunghezza k-1.

3.1.1 Grafo di De Bruijn multidimensionale

La scelta di k influenza la costruzione del grafo di De Bruijn. Valori di k piccoli comprimono più ripetizioni insieme rendendo il grafo più interconnesso e complesso, mentre i valori grandi di k potrebbero non riuscire a rilevare sovrapposizioni tra letture in particolare nelle regioni a bassa copertura ovvero il numero di volte in cui ogni nucleotide del genoma è stato sequenziato, rendendo il grafo più frammentato. Il grafo di De Bruijn multidimensionale consente di variare i valori di k nella regione in base alla copertura.

hub: un nodo che soddisfa uno delle seguenti condizioni:

1) Numero archi entranti ≠ 0
2) Numero archi uscenti ≠ 0

h-path: è un percorso che inizia e termina in nodi hub e che i nodi intermedi non sono hub (Figura 8 [18])

h-edge: è il primo arco di un h-path (α nella Figura 8 [18]), indichiamo con path(α) un h-path che ha h-edge "α". Se "a" è un arco diverso da "α", ed appartenente a path(α), allora h-edge(a) = α. Per esempio, nel grafo di Figura 8 [18], h-edge(arco 4) = α.

h-read (contig): una sequenza formata attraversando un h-path (Figura 8).

H- READ (G) : un insieme di h-read associato con tutti h-path in un grafo di de bruijn G

READ: un insieme di letture ottenuto dal sequenziamento (con alfabeto{A,G,T,C})

B(Read, k) : il grafo di De Bruijn costruito con tutti k-mer presenti in una libreria Read.

Dato un numero intero k'<k si definisce un grafo di De Bruijn multidimensionale B(READ, k', k) come unione di tutti i contig presenti grafi di De Bruijn con dimensione da k' a k :

B(READ , k', k)=DB(Read ∪ H- READ (B(READ ,k',k-1))

La Figura 9 [18] mostra un esempio di costruzione del grafo di De Bruijn B(READ, 3, 4) per genoma CATCAGATAGGA con READ che ha nove 4-mer { {ACAT, CATC, ATCA, TCAG, CAGA, AGAT, GATA, TAGG, GGAC} e tutti 3-mer presenti nel genoma. Tre dei dodici possibili 4-mer dal genoma sono mancanti dal Read ({ATAG,AGGA,GACA}).

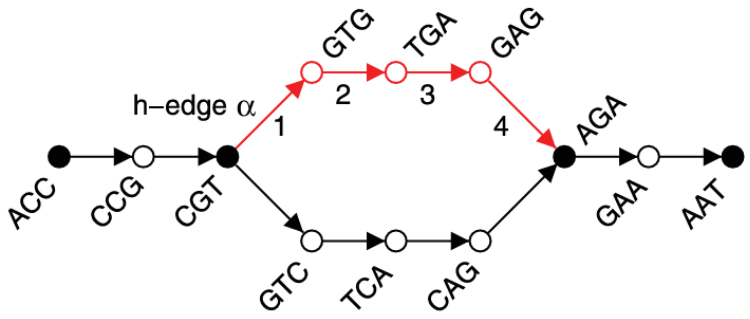


Figura 8: un grafo di De Bruijn con k=4, il percorso rosso è un h-path che attraversandolo forma un h-read: CGTGAGA

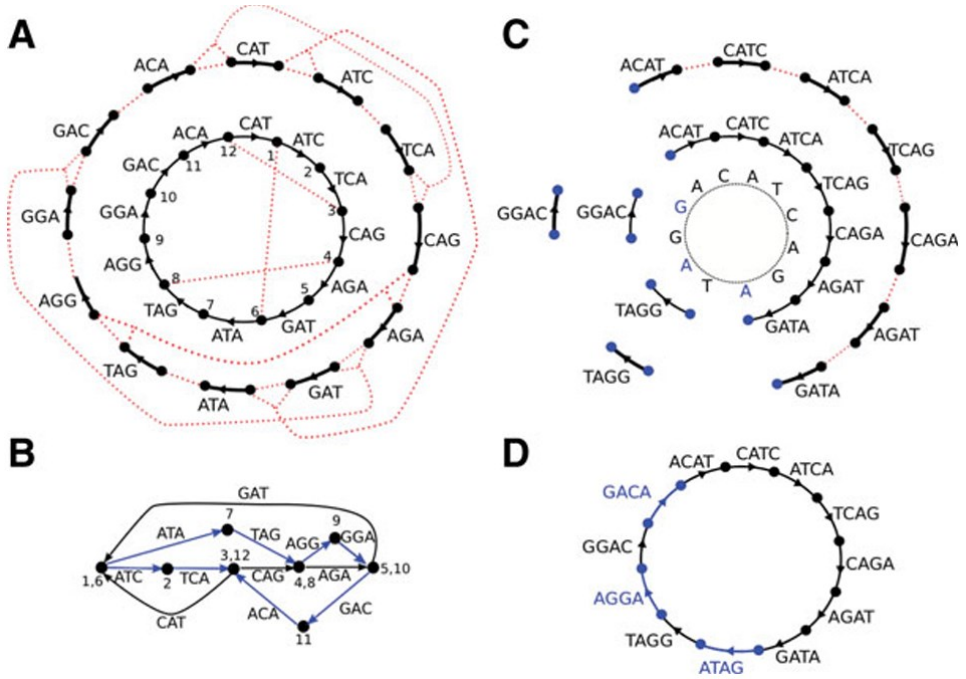


Figura 9 : esempio di costruzione di un grafo di De Bruijn multidimensionale. A) e C) rappresentando rispettivamente i grafi di De Bruijn con k = 3 e k = 4, i cerchi esterni rappresentano ogni k-mer con un arco non connesso e le linee rosse tratteggiate indicano i nodi che saranno collegati, i cerchi interno rappresentano risultato di applicazione di alcune connessioni. B) grafo di De Bruijn B(READ, 3) ottenuto connettendo tutti i nodi del READ (k=3), le tre h-path di lunghezza 2 (in colore blue) {ATAG, AGGA, GACA} saranno utilizzati per costruire il grafo multidimensionale. D) grafo di De Bruijn multidimensionale B(READ, 3, 4) è risultato di unione dei due READ connessi.

3.2 Metodi di assemblaggio con grafo di De Bruijn

Negli algoritmi di sequenziamento tradizionali di Sanger, come quello usato per assemblare genoma umano nel 2001, le letture venivano rappresentate come nodi in un grafo e gli archi rappresentavano gli allineamenti tra le letture, per ricostruire il genoma iniziale bisognava trovare un ciclo hamiltoniano seguendo gli archi in ordine. Alla fine del ciclo, la sequenza ritornava all'inizio del genoma, ogni lettura verrà inclusa una volta nell'assembly.

Con l'avvento delle tecnologie di sequenziamento di nuova generazione che producono un'enorme quantità di letture brevi, l'assemblaggio basato sui grafi di De Bruijn è diventato una tecnica dominante per il sequenziamento del DNA.

Ci sono due tipi di rappresentazione del grafo di De Bruijn in base al metodo di espressione dei nodi e degli archi: Metodo hamiltoniano ed euleriano. Nel grafo hamiltoniano di De Bruijn , il k -mer diventa un nodo e il suffisso (k -1) -mer del k -mer che si sovrapponeva al prefisso (k -1) -mer del successivo k -mer diventa un bordo. In altre parole, se il prefisso di un nodo è uguale al suffisso di un altro nodo, i due nodi sono collegati. Mentre nel grafo Euleriano di Bruijn la successione dei k -mer è un arco e il (k -1) -mer sovrapposto è un nodo.

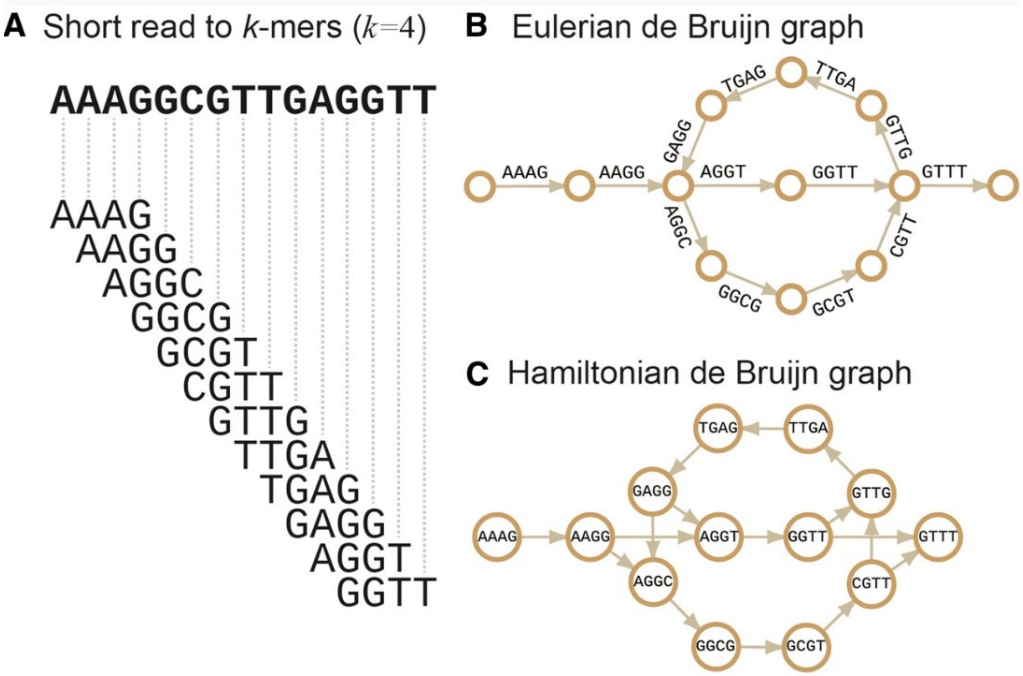


Figura 10 : a): una lettura suddiviso in k-mer di lunghezza 4; b) il grafo euleriano di De Bruijn costruito con i k-mer ottenuto, dove i rami rappresentano le sottosequenze k-mer e i nodi k-1 mer; c) grafo Hamiltoniano di De Bruijn , in cui sono usati i nodi sono per rappresentazione dei k-mer [5].

3.2.1 Metodo hamiltoniano

Ogni k-mer viene rappresentato come un nodo, viene tracciato un arco da nodo A al nodo B se il suffisso di lunghezza k-1 di A è uguale al prefisso di B, e l'arco tracciato viene etichettato con l'ordine di attraversamento, con obiettivo di costruire un ciclo hamiltoniano che attraversa ogni nodo una e sola volta.

Un ciclo hamiltoniano in un grafo orientato G (V, E) è una sequenzia di nodi v_1, v_2,..,v_n tale che:

1. ogni vertice v_i nel cammino è un vertice del grafo G.
2. ogni vertice del grafo appare esattamente una volta nella sequenza:
   $v_i \neq v_j$ per $i \neq j$
3. ogni coppia di vertici v_i, v_{i+1} è un arco del grafo G.
4. Il primo e l'ultimo vertice del ciclo sono gli stessi: $v_i = v_n$

L'approccio del ciclo hamiltoniano era fattibile per il sequenziamento del primo genoma microbico 7 nel 1995 e del genoma umano nel 2001, nonché per tutti gli altri progetti basati sul sequenziamento di Sanger, ma la ricerca di cammini hamiltoniani in un grafo è un problema NP-completo che sono i più difficili problemi nella classe NP ("problemi risolvibili non-deterministicamente in tempo polinomiale"), se si trovasse un algoritmo in grado di risolvere in tempo polinomiale un qualsiasi problema NP-completo, allora si potrebbe usarlo per risolvere in tempo polinomiale ogni problema in NP.

3.2.2 Metodo euleriano

Nel metodo Euleriano, ogni k-mer viene suddiviso in un prefisso e un suffisso di lunghezza k-1 (chiamati anche k-1 mer). Ad esempio, se ho un k-mer ATG, il prefisso sarebbe AT e il suffisso TG. Ogni k-1 mer viene salvato in un nodo, e i rami vengono etichettati con il k-mer invece che con l'ordine di attraversamento. Trovare un ciclo euleriano consente di ricostruire il genoma formando un allineamento in cui ogni k-mer successivo (proveniente da archi successivi) è spostato di una posizione.

Dato un grafo G= (V, E), dove V è un insieme finito di nodi V= {v_1, v_2…v_m} ed E= {e_1, e_2…e_n} è un insieme finito di archi, si definisce un ciclo euleriano in G una sequenza di archi e_1, e_2,…e_k tale che:

1. Ogni arco e_i nel cammino è un elemento di E.
2. Ogni arco e_i è incidente ai vertici v_i-1 e vi con $0<i<=k$.
3. Ogni arco e_i compare una e una sola volta nel cammino.

Esempio: supponiamo di avere tre letture ACGTAC, GTACGT, TACGTA, suddividiamo le letture in k-mer con k=3: ACG, CGT, GTA, TAC, GTA, TAC, ACG, CGT, TAC, ACG,

CGT, GTA, quindi i nodi del grafo sono: AC, CG, GT, TA, e gli archi: ACG, CGT, GTA, TAC (Figura 11)

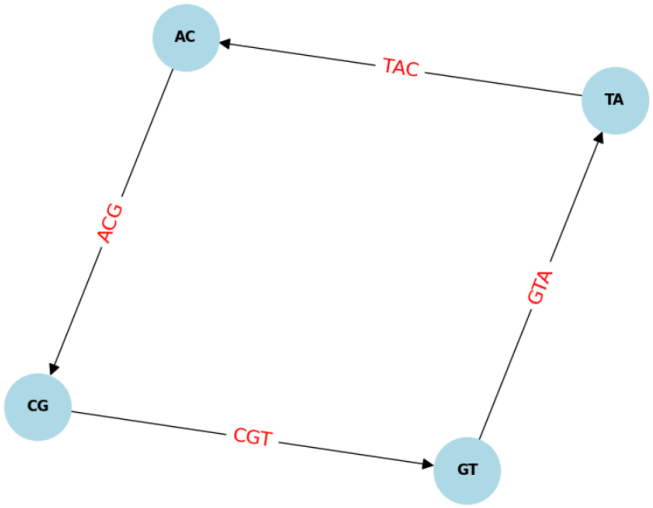


Figura 11: un grafo euleriano di De Bruijn con k=3, con letture iniziali ACGTAC, GTACGT, TACGTA, costruito con Python

Per sapere se un grafo orientato contiene un ciclo euleriano è possibile applicare il teorema di Eulero: un Grafo orientato G (V, E) contiene un ciclo se e solo se soddisfa le seguenti condizioni:

1) per ogni $v_i, v_j \in V$, esiste un cammino diretto che va da v_i a v_j , con $v_i \neq v_j$
2) per ogni $v_i \in V$, il numero di archi entranti deve essere uguale al numero di archi uscenti.

3.3 Sfide nell'assemblaggio del grafo di De Bruijn

In pratica applicare i grafi di De Bruijn non è una procedura semplice, in questa sezione discutiamo alcune delle sfide principali nell'assemblaggio de novo e gli strumenti per affrontarli.

3.3.1 Correzione degli errori

Ogni errore in una lettura crea un “bulge ” nel grafo di De Bruijn , che va a complicare gli assemblaggi. Un “bulge” inizia in un nodo del grafo da cui partono due o più archi divergenti, i quali portano a cammini alternativi che rappresentano sequenze differenti ma correlate, i cammini divergenti alla fine si riconnettono in un nodo comune del grafo, chiudendo la bolla. Nella Figura 12 [1] è presente un grafo De Bruijn che ha un bulge (percorso TGGAGTG) dovuto ad un errore di sequenziamento.

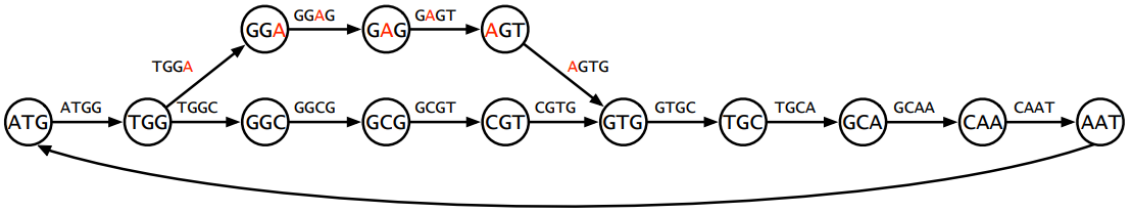


Figura 12: un grafo di De Bruijn con un bulge dovuto dall’errore di sequenziamento TGGAGTG [1]

La correzione degli errori di sequenziamento consiste nell'identificare gli errori di sequenziamento e nel distinguerli dagli alleli eterozigoti. I metodi di correzione posso essere di tre tipi:

1) <u>k-spectrum based</u>

Chiamato anche conteggio dei k-mer, il quale consiste nel contare tutte le possibili sottosequenze di lunghezza k, e si presuppone che k-mer frequenti possono rappresentare sequenze biologicamente rilevanti, mentre k-mer rari possono indicare errori di sequenziamento o mutazioni, una volta ottenuta tutti i possibili k-mers, viene determinata la frequenza di ciascun k-mer nella sequenza, utilizzando algoritmi efficienti come l'uso di tabelle hash o altre strutture di dati simili. I k-mer con frequenze molto basse vengono considerati potenziali errori, poiché le vere sequenze di DNA tendono a produrre k-mer che appaiono più frequentemente e i k-mer frequenti rappresentano sequenze più affidabili e biologicamente significative. Per correggere gli errori, i k-mer rari vengono confrontati con quelli frequenti, se un k-mer raro differisce da un k-mer frequente per un solo nucleotide, si può ipotizzare che l'errore sia stato causato da una mutazione o da un errore di sequenziamento. La correzione avviene sostituendo il nucleotide errato, in modo che il k-mer raro corrisponda al k-mer frequente più simile.

La Figura 13 [5] rappresenta un l'istogramma della profondità k -mer, dove l'asse x si riferisce alle molteplicità dei k-mer D(k), l'asse y si riferisce alle frequenze delle molteplicità f(D(k)), per esempio se ho k-mers: ["AAG", "AGC", "GCT", "CTA", "TAG", "AGC", "GCT", "CTT"] , quindi le molteplicità dei k-mers sono: "AAG": 1, "AGC": 2, "GCT": 2, "CTA": 1, "TAG": 1, "CTT": 1, .allora le frequenze f(1)=4, f(2)=2, la funzione esponenziale rossa corrisponde a k-mer errati che compaiono solo in poche letture, le curve gaussiane rappresentano i dati privi di errori nel caso ideale, e i punti neri sono i dati reali. I k -mer. In particolare, i k-mer risultanti dall'eterozigosità dovrebbero essere gestiti con maggiore attenzione perché questi k -meri sono simili ai k -meri errati, appaiono i loro picchi D'/n, dove D' è il picco principale dell'istogramma e n è plodia [5].

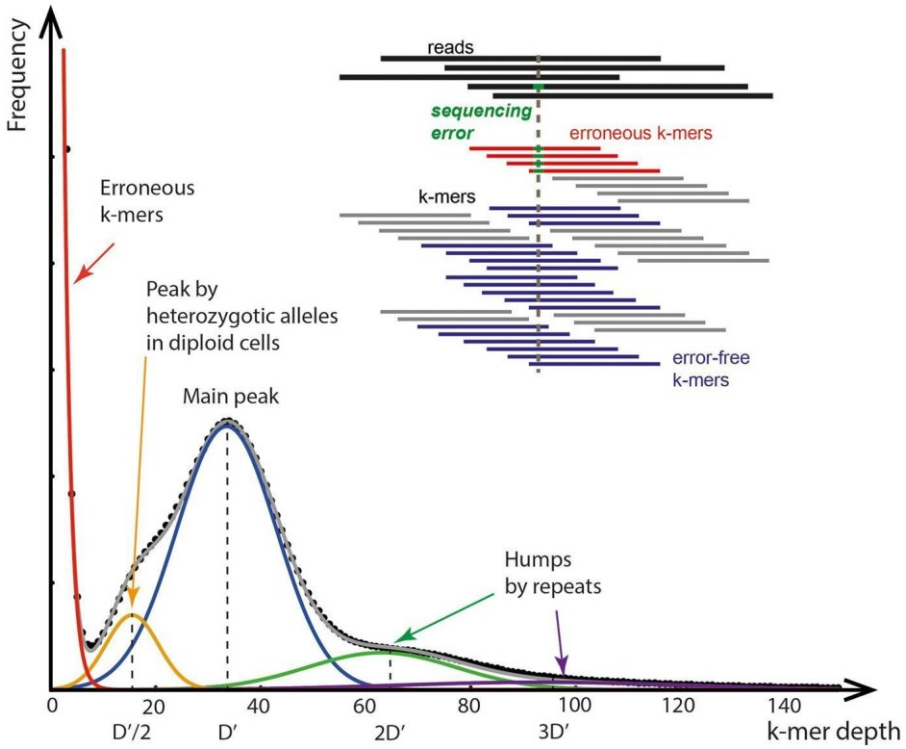


Figura 13: l'istogramma della profondità k -mer [5]

2) Albero o array di suffissi

Questo metodo prevede la costruzione di un array/albero (Figura 14b [5]) di suffissi di tutti i suffissi letti e correggono gli errori utilizzando le frequenze k -mer associate ai nodi dell'albero dei suffissi. Un trie è una struttura ad albero dove ogni nodo rappresenta un carattere e un percorso da radice alla foglia rappresenta una lettura (o suffisso), Se la frequenza di un percorso è molto bassa rispetto agli altri percorsi, potrebbe essere un errore.

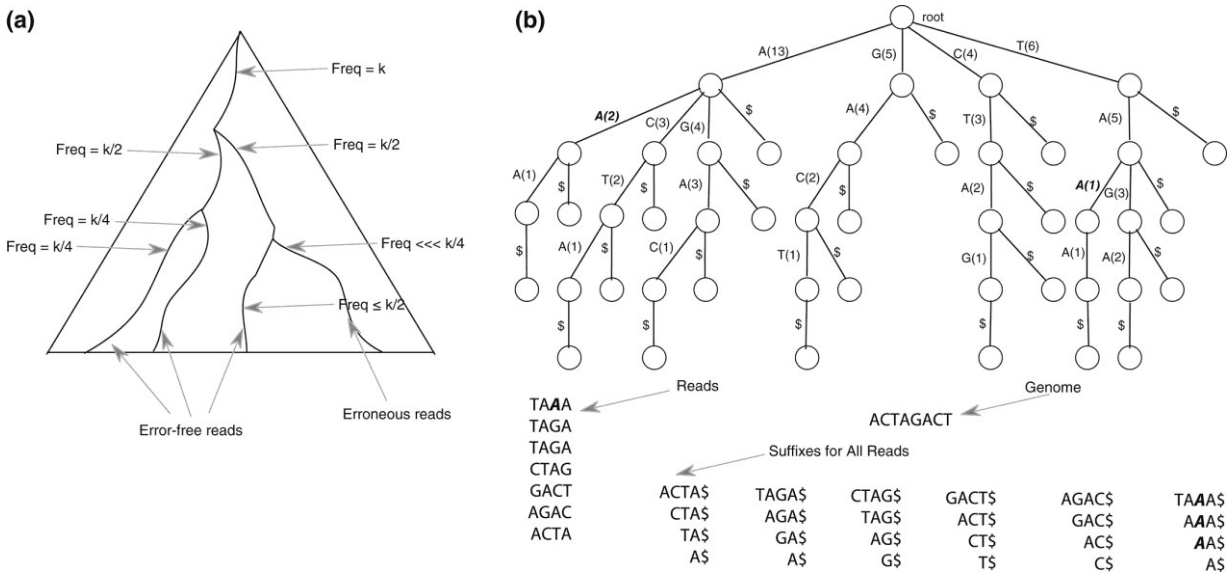


Figura 14: a) un errore sul percorso più a destra fa sì che il ramo abbia una frequenza molto bassa (<< k/2) rispetto al ramo fratello (k/2); b) esempio di un trie per un genoma molto corto con la lettura TAAA che presenta un errore nella terza posizione [5]

3) metodi basati sull'allineamento di sequenze multiple

Questo metodo utilizza allineamento multiplo delle letture che possono appartenere allo stesso locus genomico per confrontare tra loro o con un genoma di riferimento ed è possibile identificare e correggere errori di sequenziamento.

3.2.2 Gestione delle strutture ripetitive

Il problema dell'assemblaggio del genoma è complicato dalle strutture ripetitive che creano ambiguità e lacune nel processo di assemblaggio. Sebbene il percorso più breve attraverso un grafo di k-mer possa sembrare una soluzione logica, non tiene conto delle complessità introdotte dalle ripetizioni del genoma. I metodi per affrontare questo problema sono diversi: come aumentare la profondità della della lettura, poiché la profondità di lettura nelle regioni ripetute è superiore a quella di altre regioni, è possibile stimare il numero di copie della regione [5]. Oppure utilizzare la dimensione dell'inserto delle letture, in cui ciascun frammento corrisponde a una regione al di fuori delle regioni ripetitive.

3.2.3 Costi computazionali

Memoria RAM

Grafo di De Bruijn genera (l-k+1) k mers da una lettura di lunghezza l, e per memorizzare un genoma di dimensione G è necessario circa 2(k+1)G byte di memoria. Per esempio, ALLPATHS-LG (la dimensione predefinita di k -mer è 96) richiede almeno 512 GB di memoria per un assemblaggio dell'intero genoma umano [5]. La Figura 15 [5] mostra le strategie utilizzate nei rispettivi assemblatori.

RAM memory	Distributed memory on linux cluster	ABySS, Meraculous
	2 bits nucleotides	SparseAssembler
	FM-index	SGA
	Bloom filter	SGA
	Reducing depth by super-reads	MaSuRCA
	Lightweight hash	ABySS, Meraculous

Figura 15: strategie per ridurre l'occupazione di memoria RAM [5]

Tempo di Calcolo

Sia x numero di rami nel grafo, la complessità f(x) di trovare un percorso euleriano in un grafo di De Bruijn è generalmente lineare rispetto al numero di rami del grafo ($f(x)= a_0 + a_1*x_1+, a_2*x_2 +\ldots+ a_n*x_n$ quindi $f(x) \sim x^n$)

ed è più adatto per l'assemblaggio di genomi grandi con molte ripetizioni, ma può essere meno accurato nella risoluzione di ambiguità causate da sequenze ripetitive rispetto all'approccio hamiltoniano. Mentre la complessità per trovare un percorso hamiltoniano la è NP-completa, il che significa il tempo di calcolo ($f(x) \sim e^{ax}$) .

Nella Figura 16 [5] è mostrato un grafo che confronta le due funzioni di complessità dei due tipi di grafi, hamiltoniano ed Euleriano, con l'asse delle ordinate che rappresenta la complessità e asse delle ascisse che rappresenta il numero di rami.

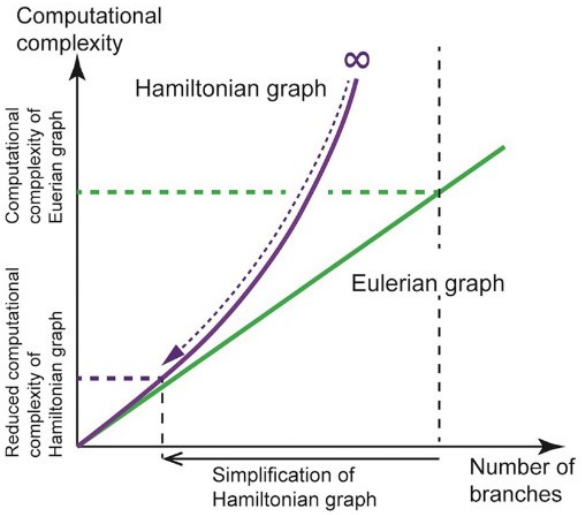


Figura 16: Complessità computazionali di Grafo euleriano e Hamiltoniano [5]

In teoria, l'approccio del grafo hamiltoniano di De Bruijn potrebbe essere più veloce di quello euleriano di De Bruijn in un grafo a bassa complessità. Quando si eliminano i percorsi ambigui o ridondanti, la complessità computazionale degli assemblatori hamiltoniani diminuisce esponenzialmente secondo la curva NP, mentre la complessità computazionale degli assemblatori euleriani diminuisce linearmente secondo la curva P. Quindi, si può dire che esiste un compromesso tra accuratezza e tempo di calcolo: il percorso euleriano è molto efficiente, sacrificando l'accuratezza nelle regioni altamente ripetitive, mentre il percorso hamiltoniano può essere utile per ottenere una maggiore accuratezza in grafi meno complessi, accettando però un aumento significativo del tempo di calcolo.

4 Tool utilizzati

4.1 Il tool SPAdes

SPAdes [18] è un assemblatore basato su grafi di De Bruijn ed è stato progettato per il sequenziamento di singole cellule. L'approccio si caratterizza per l'utilizzo di più grafi di De Bruijn (ciascuno costruito con diverse dimensioni di k-mer) per gestire le grandi variazioni di copertura nel genoma che sono una caratteristica del sequenziamento di singole cellule SCS.

4.1.1 Descrizione della pipeline

Le fasi principali della pipeline di SPAdes sono [18]:

Fase 1: costruisce un grafo di De Bruijn multidimensionale utilizzando k-mer di diverse dimensioni e rimuove gli errori.

Fase 2: estrae inizialmente i k-bimer dai biread (paired-end read), risultando in k-bimer con stime di distanza imprecise (ereditate dalle stime di distanza dei biread). L'approccio di aggiustamento dei k-bimer trasforma questo set di k-bimer (stime di distanza imprecise) in un set di k-bimer con stime di distanza esatte. L'approccio di aggiustamento consiste in quattro trasformazioni (Figura 18 [18]) :

B-transformation: decomposizione dei bireads in k-bimers con distanze genomiche stimate.

Un k-bimer (a|b, d) è costituito da due k-mer a,b che hanno una distanza genomica d tra loro che è il numero di basi (nucleotidi) che separano l'inizio del "a" dall'inizio del "b" all'interno della sequenza di DNA originale.

H-transformation: creazione di istogrammi delle distanze genomiche stimate tra coppie di h-biedge.

Ogni k-bimer formato applicando la trasformazione B ai bireads definisce una coppia di archi nel grafo de Bruijn, che chiamiamo biedge, tramite i quali si va a costruire h-biedge.

Dato un biedge (a|b,d) definiamo h-biedege(a|b,d) = (h-edge(a) | h-edege(b), D) dove D è la distanza genomica tra h-edge(a) e h-edge(b)

A-transformation: analisi degli istogrammi per rivelare distanze genomiche precise. La figura 17[18] rappresenta l'istogramma degli h-biedge (a|b,*), che contiene tutte le distanze genomiche estimate tra h-edge(a) e h-edge(b), il picco nell'istogramma con valore 72163 rappresenta la distanza genomica esatta, quindi h-biedge esatto è (a|b, 72163). A destra è presente un grafo semplificativo che rappresenta la connessione di path(a) (un arco condensato che rappresenta 72049 archi) e path(b) (che rappresenta 46097 archi) tramite un h-path più corto di lunghezza 114.

E-transformation: trasformazione delle distanze genomiche stimate tra h-path in set di k-bimer con distanze genomiche giuste.

Fase 3 :costruzione del grafo di assemblaggio accoppiato che utilizza tutti h-biedge con distanze corrette come archi [18].

Fase 4 : costruzione dei contig finali utilizzando il grafo di assemblaggio accoppiato.

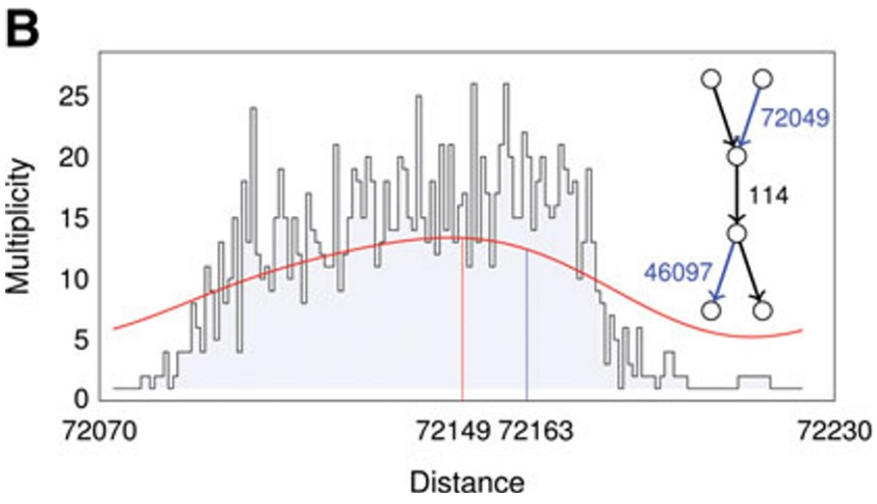


Figura 17 : l'istogramma delle distanze genomiche

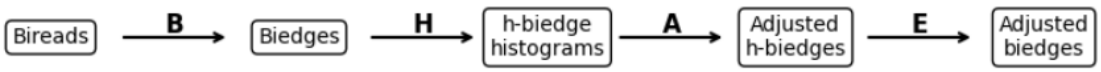


Figura 18 : le trasformazioni nell'aggiustamento dei k-bimer

4.1.2 Setting dei parametri

Per gli esperimenti è stato utilizzato SPAdes v3.15.3 presente sulla piattaforma Kbase [17], che accetta input solo la libreria di letture paired-end.

Ci sono tre parametri di base per il tipo di DNA di input: standard per il DNA isolato, cellula singola per cellule batteriche selezionate in flusso da amplificazione a spostamento multiplo (MDA) o plasmide per DNA plasmidico.

Tre parametri avanzati di input sono: (i) lunghezza minima del contig da segnalare (predefinita 500); (ii) un elenco di dimensioni di k-mer per i grafi De Bruijn ; (iii) l'opzione "solo assembly", che impedisce qualsiasi correzione di errore. Una volta completata-l'esecuzione del codice con successo, l'App crea un oggetto Assembly KBase, che apparirà nel riquadro dei dati e viene generato un riepilogo sulla valutazione della qualità QUAST.

4.2 Il tool QUAST

QUAST (Quality Assesment Tool for Genome Assemblies) è uno strumento per la valutazione della qualità degli assemblaggi genomici. Le principali metriche di valutazione forniti da Quast includono:

1) Number of contig: Il numero totale di contig prodotti dall'assemblaggio.
2) Total length: La lunghezza totale degli assemblaggi, espressa solitamente in basi nucleotidiche.
3) Largest contig: lunghezza del contig più lungo.
4) N50: è la lunghezza per cui la raccolta di tutti i contig di quella lunghezza o più lunga copre almeno metà della lunghezza totale di tutti i contig/scaffolds nell'assemblaggio.
5) L50: Numero minimo di contig o scaffolds necessari per raggiungere o superare la lunghezza N50.
6) NG50: è la lunghezza per cui la raccolta di tutti i contig di quella lunghezza o più copre almeno metà del genoma di riferimento.
7) GC (%): è il numero totale di nucleotidi G e C nell'insieme, diviso per la lunghezza totale dell'insieme.

8) Number of misassemblies: Il numero di errori nel posizionamento dei contig rispetto al genoma di riferimento, come le inversioni, le duplicazioni o altre anomalie strutturali.
9) Misassembly rate: La percentuale di contig o basi nucleotidiche assemblate in modo errato rispetto al genoma di riferimento.
10) Number of Predicted genes: è il numero di geni nell'assemblaggio trovato da GeneMarkS, GeneMark-ES, MetaGeneMark o GlimmerHMM.

5 Assemblaggio di un genoma

In questa sezione si riportano i risultati dell'assemblaggio di un genoma effettuata con un tool, SPAdes [17] basato su grafo di De Brujin. Gli esperimenti sono stati effettuati su un laptop con CPU : Intel Core i7-8565U CPU @1.80GHz e 8 GB di RAM. Per la valutazione della qualità dell'assemblaggio è stato utilizzato il tool QUAST.

5.1 Dataset utilizzato

Per gli esperimenti è stato fornito come file di input "rhodo.art.q20.PE.read" (caratteristiche nella Figura 19 [17]) scaricato dalla libreria Paired-end della piattaforma Kbase, e ha in totale 386106 letture, sequenziato con Illumina, e che ha le seguenti caratteristiche:

1) qual_min: È il valore minimo di qualità di base (Phred score) osservato tra tutte le basi sequenziate. In questo caso, è 20.0, il che indica che la qualità minima delle basi sequenziate è 20, una qualità che corrisponde a un'accuratezza di lettura del 99%.
2) qual_mean : È il valore medio di Phred score di tutte le basi sequenziate.
3) number_of_duplicates: Rappresenta il numero totale di duplicati rilevati durante l'analisi di sequenziamento.
4) qual_stdev: È la deviazione standard dei valori di qualità di base (Phred score).
5) read_length_stdev: È la deviazione standard delle lunghezze delle letture sequenziate. Qui è 0.0, il che indica che tutte le letture hanno la stessa lunghezza media di 100 basi.
6) qual_max: È il valore massimo di qualità di base (Phred score) osservato tra tutte le basi sequenziate.

7) phred_type: Indica la Phred score utilizzata per le letture (Q33).

qual_min	20.0
qual_mean	53.044
sequencing_tech	Illumina
number_of_duplicates	9573
read_length_mean	100.0
qual_stdev	10.5457
read_length_stdev	0.0
qual_max	61.0
total_bases	38610600
single_genome	1
gc_content	0.6660900000000001
read_count	386106
phred_type	33

Figura 19: caratteristiche della libreria paired-end fornito come input [17]

5.2 Risultati

La Figura 20 [17] mostrato il report fornito da QUAST sul risultato dell'assemblaggio, il quale ci suggerisce alcune caratteristiche

1) Numero di contig: Il numero totale dei contig è 233, con tutti di lunghezza superiore a 1.000 bp e 143 di lunghezza superiore a 10.000 bp. Tuttavia, non ci sono contig che superano i 100.000 bp o 1.000.000 bp. Questo suggerisce un assemblaggio frammentato, con molte sequenze corte rispetto al genoma completo.
2) Lunghezza dei Contig: Il contig più lungo misura 64.486 bp. La lunghezza totale dell'assemblaggio è di 3.855.055 bp, che rimane invariata se si considerano contig superiori a 1.000 bp. Questo indica che tutti i contig contribuiscono in modo significativo alla lunghezza totale dell'assemblaggio. La lunghezza totale dei contig superiori a 10.000 bp è 3.219.000 bp, mostrando che una parte significativa dell'assemblaggio è costituita da contig di questa lunghezza.

3) Metriche di Qualità: I valori di N50 (20.489 bp) e N75 (13.108 bp) indicano la lunghezza media dei contig che costituiscono il 50% e 75% del genoma assemblato sono rispettivamente 20.489bp e 13.108bp.
4) L50 = 58 e L75 =116, significa che 58 contig sono necessari per coprire metà del genoma e 75 contig sono necessari per coprire 75% del genoma
5) Il contenuto di GC è 66,63%, suggerendo una composizione del genoma relativamente alta in basi GC.
6) Il numero totale di N's (gap non risolti) è 784, con una densità di 20,34 N's per 100.000 bp. Questo indica la presenza di regioni non risolte nell'assemblaggio.
7) Il numero totale di geni predetti è 3.690, di cui 3.577 sono completi e 113 parziali. Ci sono 3.246 geni di lunghezza superiore a 300 bp e 499 superiori a 1500 bp. Solo 37 geni sono superiori a 3.000 bp, il che potrebbe riflettere la complessità e la frammentazione dell'assemblaggio.

Statistics without reference	**SPAdes.Assembly**
# contigs	233
# contigs (>= 0 bp)	233
# contigs (>= 1000 bp)	233
# contigs (>= 10000 bp)	143
# contigs (>= 100000 bp)	0
# contigs (>= 1000000 bp)	0
Largest contig	64 486
Total length	3 855 055
Total length (>= 0 bp)	3 855 055
Total length (>= 1000 bp)	3 855 055
Total length (>= 10000 bp)	3 219 000
Total length (>= 100000 bp)	0
Total length (>= 1000000 bp)	0
N50	20 489
N75	13 108
L50	58
L75	115
GC (%)	66.63
Mismatches	
# N's	784
# N's per 100 kbp	20.34
Predicted genes	
# predicted genes (unique)	3690
# predicted genes (>= 0 bp)	3577 + 113 part
# predicted genes (>= 300 bp)	3246 + 108 part
# predicted genes (>= 1500 bp)	499 + 15 part
# predicted genes (>= 3000 bp)	37 + 2 part

Figura 20 : statistiche Quast fornito da Kbase [17]

La Figura 21 [17] mostra un grafo della lunghezza cumulativa che indica come la lunghezza totale dell'assemblaggio cresce man mano che vengono aggiunti contig di diverse lunghezze. L'asse X rappresenta i contig ordinati in ordine decrescente di lunghezza. Ogni punto sull'asse X corrisponde a un contig. L'asse Y indica la lunghezza cumulativa totale dell'assemblaggio fino a quel contig specifico. Man mano che si sposta a destra lungo l'asse X, l'asse Y mostra la somma delle lunghezze di tutti i contig inclusi fino a quel punto. Nel nostro caso la funzione ha pendenza più graduale che indica un contributo minore delle sequenze corte alla lunghezza totale dell'assemblaggio.

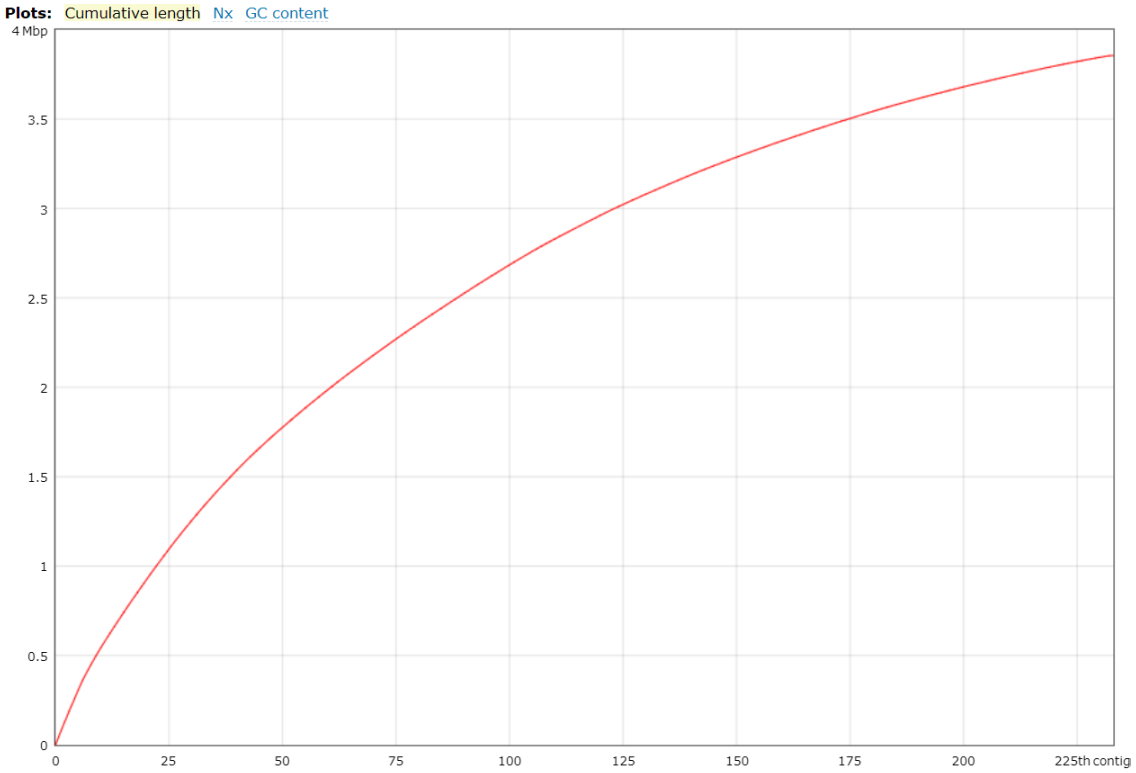


Figura 21: grafico di lunghezza cumulativa dell'assemblaggio [17]

Il grafico Nx (Figura 22 [17]) mostra vari valori di Nx (come N50, N75, ecc.) L'asse X rappresenta la percentuale cumulativa della lunghezza totale dell'assemblaggio, dal 0% al 100%. L'asse Y indica la lunghezza dei contig corrispondente a ciascuna frazione cumulativa della lunghezza totale dell'assemblaggio. Nell'esperimento effettuato la curva inizia alta e scende gradualmente, indicando una buona distribuzione delle lunghezze dei contig.

Al punto del 50% sull'asse X, la curva si interseca con l'asse Y a 20.489 bp, che è il valore di N50 , al punto del 75% sull'asse X, la curva si interseca con l'asse Y a 13.108 bp, che è il valore di N75.

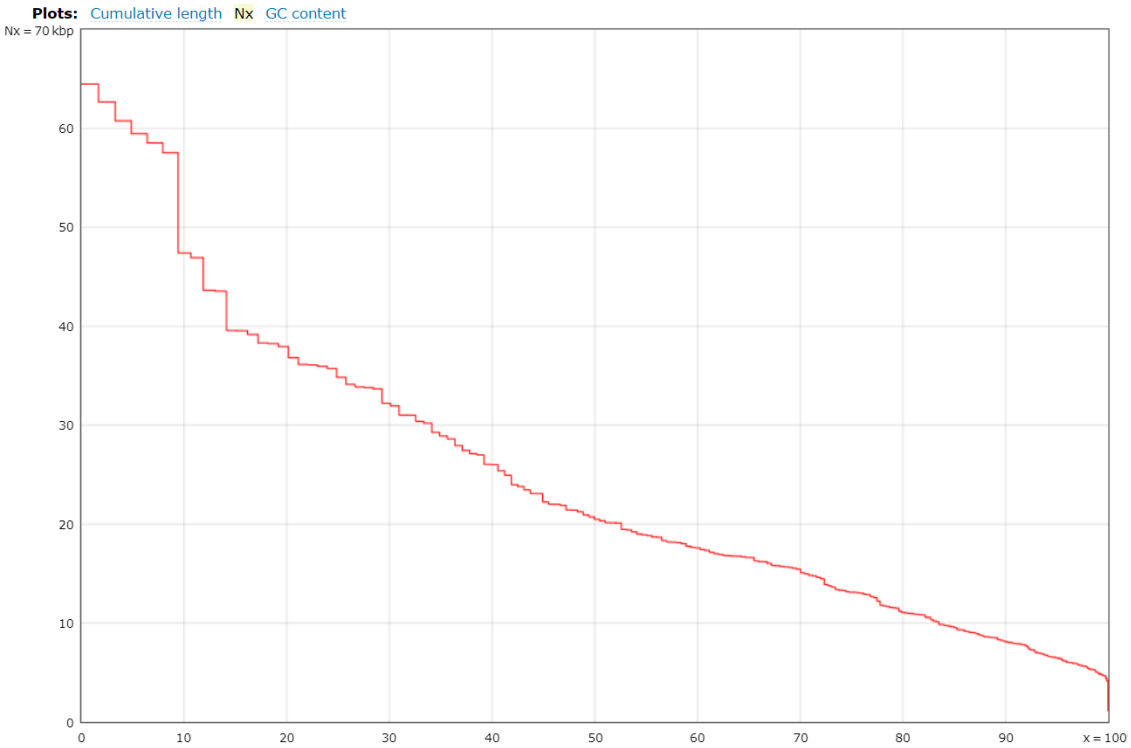


Figura 22: curva Nx fornito da Kbase utilizzando QUAST [17]

La Figura 23 [17] mostra la distribuzione del contenuto di guanina e citosina (GC) nei contig di un assemblaggio genomico. L'asse rappresenta la percentuale di basi guanina (G) e citosina (C) rispetto al totale delle basi in un contig, l'asse Y indica il numero di contig che hanno una certa percentuale di GC, tali contig sono stati suddivisi sequenze di cento basi ciascuna, senza sovrapposizioni tra loro. Nell'esperimento effettuato il picco principale è intorno al 66,63%, suggerendo che la maggior parte dei contig ha un contenuto di GC vicino a questa percentuale.

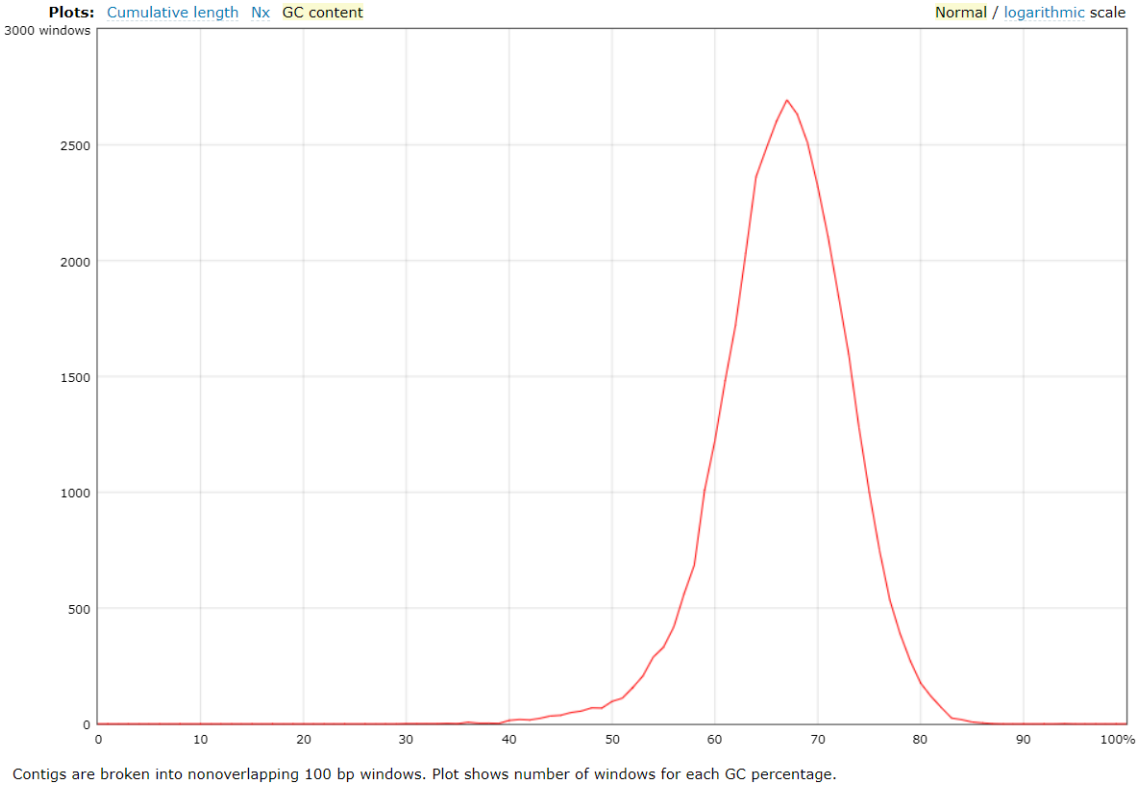


Figura 23: Il grafico del contenuto GC [17]

6 Conclusione

L'assemblaggio del genoma rimane un'area in rapido sviluppo, con continui miglioramenti nelle tecnologie di sequenziamento e negli algoritmi di assemblaggio. Con l'avanzare delle tecnologie e delle metodologie, possiamo aspettarci ulteriori progressi che ci avvicineranno sempre più alla completa comprensione dei genomi complessi e delle loro funzioni biologiche. Il futuro dell'assemblaggio genomico è promettente, con nuove tecnologie e approcci innovativi che continueranno a migliorare la nostra capacità di decifrare i complessi codici genetici degli organismi. Il grafo di De Bruijn ha rappresentato una svolta nell'assemblaggio genomico, permettendo di trasformare un problema di sovrapposizione di letture in un problema di percorso nel grafo. Questo approccio ha dimostrato di essere particolarmente efficace con le letture corte prodotte dalle tecnologie NGS e continuerà ad essere una componente chiave di questo progresso, grazie alla sua versatilità ed efficacia nella risoluzione delle complessità genomiche.

7 Ringraziamenti

Mi è doveroso dedicare queste righe del mio elaborato alle persone che hanno contribuito, con il loro instancabile supporto, alla realizzazione dello stesso.

Innanzi tutto, un ringraziamento speciale alla mia relatrice Prof.ssa. Pizzi, per la sua infinita disponibilità e tempestività ad ogni mia richiesta, per l'immensa pazienza ad ogni mio dubbio insensato e per gli indispensabili consigli.

Ringrazio profondamente anche i miei genitori, che mi hanno sempre sostenuto, appoggiando ogni mia decisione. Grazie a loro per avermi dato l'opportunità di proseguire gli studi, senza di loro non avrei raggiunto questo traguardo.

Un grazie di cuore a tutti i professori del dipartimento che mi hanno seguito in questi tre anni. Senza la loro pazienza e disponibilità, non sarei arrivato a questo livello di formazione.

Infine, ringrazio i miei amici più cari, che mi hanno fornito informazioni e supporto. È grazie a loro che ho superato i momenti più stressanti. Senza di loro, non sarei riuscito a fronteggiare tutte le pressioni incontrate.

Riferimenti bibliografici

[1] Compeau PE, Pevzner PA, Tesler G. How to apply De Bruijn graphs to genome assembly. Nat Biotechnol. 2011 Nov 8;29(11):987-91. doi: 10.1038/nbt.2023. PMID: 22068540; PMCID: PMC5531759.

[2] Rizzi, R., Beretta, S., Patterson, M. et al. Overlap graphs and De Bruijn graphs: data structures for de novo genome assembly in the big data era. Quant Biol 7, 278–292 (2019). https://doi.org/10.1007/s40484-019-0181-x

[3] Xiao Yang, Sriram P. Chockalingam, Srinivas Aluru, A survey of error-correction methods for next-generation sequencing, Briefings in Bioinformatics, Volume 14, Issue 1, January 2013, Pages 56–66, https://doi.org/10.1093/bib/bbs015

[4] Zhenyu Li, Yanxiang Chen, Desheng Mu, Jianying Yuan, Yujian Shi, Hao Zhang, Jun Gan, Nan Li, Xuesong Hu, Binghang Liu, Bicheng Yang, Wei Fan, Comparison of the two major classes of assembly algorithms: overlap–layout–consensus and de-Bruijn-graph, Briefings in Functional Genomics, Volume 11, Issue 1, January 2012, Pages 25–37, https://doi.org/10.1093/bfgp/elr035

[5] Jang-il Sohn, Jin-Wu Nam, The present and future of de novo whole-genome assembly, Briefings in Bioinformatics, Volume 19, Issue 1, January 2018, Pages 23–40, https://doi.org/10.1093/bib/bbw096

[6] Taishan Hu, Nilesh Chitnis, Dimitri Monos, Anh Dinh, Next-generation sequencing technologies: An overview, Human Immunology, Volume 82, Issue 11, 2021, Pages 801-811, ISSN 0198-8859, https://doi.org/10.1016/j.humimm.2021.02.012

[7] Kchouk M, Gibrat JF, Elloumi M (2017) Generations of Sequencing Technologies: From First to Next Generation. Biol Med (Aligarh) 9:395. doi:10.4172/0974-8369.1000395

[8] Alic, A. S., Ruzafa, D., Dopazo, J., & Blanquer, I. (2016). Objective review of de novo stand-alone error correction methods for ngs data. WIREs Computational Molecular Science, 6(2), 111-146. https://doi.org/10.1002/wcms.1239

[9] Walker, Bruce J., et al. "Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement." PloS one 9.11 (2014): e112963.

https://doi.org/10.1371/journal.pone.0112963

[10] Kelley DR, Schatz MC, Salzberg SL. Quake: quality-aware detection and correction of sequencing errors, Genome Biol, 2010, vol. 11 pg. R116

https://doi.org/10.1186/gb-2010-11-11-r116

[11] Hackl T, Hedrich R, Schultz J, Förster F. proovread: large-scale high-accuracy PacBio correction through iterative short read consensus. Bioinformatics. 2014 Nov 1;30(21):3004-11.

https://doi.org/10.1093/bioinformatics/btu392

[12] Salmela, Leena, and Eric Rivals. "LoRDEC: accurate and efficient long read error correction." Bioinformatics 30.24 (2014): 3506-3514.

https://doi.org/10.1093/bioinformatics/btu538

[13] Pallavi Mishra, Ranjeet Maurya, Himanshu Avashthi, Shikha Mittal, Muktesh Chandra, Pramod Wasudeo Ramteke, Chapter 4 - Genome assembly and annotation, Editor(s): Dev Bukhsh Singh, Rajesh Kumar Pathak, Bioinformatics, Academic Press,2022,Pages 49-66,ISBN 9780323897754,

https://doi.org/10.1016/B978-0-323-89775-4.00013-4

[14] Sanger, F., Air, G., Barrell, B. et al. Nucleotide sequence of bacteriophage fX174 DNA. Nature 265, 687–695 (1977). https://doi.org/10.1038/265687a0

[15] 2017 Illumina, inc. An introduction to Next-Generation Sequencing Technology.

https://www.illumina.com/content/dam/illumina-marketing/documents/products/illumina_sequencing_introduction.pdf

[16] 2011, Life Technologies Corporation.

https://tools.thermofisher.com/content/sfs/brochures/cms_094273.pdf

[17] KBase. https://kbase.us/applist/apps/kb_SPAdes/run_SPAdes/release?gclid=CjwKCAjwiOCgBhAgEiwAjv5whOQQCYKCC1MFbVZz60sSBIhUfaG4ijmC3fuG-nyNQ-6Sz-YChIvQQhoCScEQAvD_BwE

[18] Bankevich A, Nurk S, Antipov D, Gurevich AA, Dvorkin M, Kulikov AS, Lesin VM, Nikolenko SI, Pham S, Prjibelski AD, Pyshkin AV, Sirotkin AV, Vyahhi N, Tesler G, Alekseyev MA, Pevzner PA. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol. 2012 May;19(5):455-77. doi: 10.1089/cmb.2012.0021. Epub 2012 Apr 16. PMID: 22506599; PMCID: PMC3342519.