UNIVERSITÀ DEGLI STUDI DI PADOVA

UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea in

INGEGNERIA INFORMATICA

Stima della dimensione del genoma tramite k-mers: confronto tra metodi computazionali.

Relatore:
Prof. Matteo Comin

Laureando:
Mattia Tamiazzo

Data di laurea 23/09/2022

Anno Accademico 2021-2022

Sommario

La dimensione del genoma è la quantità totale di DNA nucleare aploide presente nelle cellule di un organismo. La determinazione della dimensione del genoma costituisce un argomento di interesse, perché non esistono valori di riferimento assoluti che permettono di stabilire quale approccio sia più efficace, e perché i metodi sperimentali per la sua misurazione sono attualmente costosi dal punto di vista temporale ed economico. Una soluzione alla stima della dimensione del genoma con metodi computazionali è l'utilizzo di k-mers, sottostringhe di DNA di lunghezza k. Questa trattazione si pone l'obiettivo di analizzare e comparare vari approcci algoritmici pubblicati in letteratura per la stima della dimensione del genoma.

Indice

Capitolo 1

Introduzione

Il sequenziamento del DNA costituisce una tecnica fondamentale per lo studio del genoma di una specie, perché permette di determinare l'ordine dei nucleotidi che costituiscono il DNA. Tale processo trova applicazione in molti studi biologici che riguardano vari ambiti, come ad esempio la medicina riproduttiva, l'oncologia o l'infettivologia, attraverso indagini tra cellule diverse dello stesso individuo o lo studio delle mutazioni genetiche tra individui di una stessa specie [1].

1.1 Metodi di sequenziamento

1.1.1 Metodi di prima generazione

Le prime tecniche di sequenziamento del genoma furono sviluppate nella seconda metà del Novecento. Nel 1977 infatti vennero pubblicati due metodi di sequenziamento: il metodo Sanger, nel quale la sequenza di nucleotidi viene frammentata grazie a un terminatore di catena [2, 3], e il metodo di Maxam e Gilbert, in cui vengono utilizzati reagenti chimici per tagliare il DNA in frammenti in corrispondenza di basi specifiche [4]. Entrambi i metodi prevedono la misurazione dei frammenti creati tramite elettroforesi su gel di poliacrilammide con una corsia per base, in modo che siano separati in ordine di lunghezza e sia possibile dedurre l'ordine delle basi della sequenza in esame [5]. Mentre il metodo Maxam-Gilbert è stato progressivamente accantonato per la difficoltà tecnica e l'uso di sostanze tossiche che lo caratterizzano, il metodo Sanger è stato affinato con l'utilizzo di marcatori fluorescenti diversi per ogni base, che un lettore laser può distinguere restituendo la sequenza di basi di ogni frammento.

1.1.2 Shotgun assembly

Il sequenziamento del genoma può essere fatto su sequenze più lunghe, come un intero cromosoma, tramite *shotgun assembly*, metodo suggerito già nel 1979 [6]. Il genoma iniziale viene duplicato in modo da produrne più copie identiche, le quali vengono tagliate

in frammenti casuali (*shotgun*) che possono essere letti singolarmente. Si procede quindi con l'*assembly*, cioè la ricostruzione del genoma iniziale. L'assemblamento dei frammenti può avvenire con due metodi diversi, tramite *reference assembly* o con *de novo assembly*.

Reference assembly Per la ricostruzione del genoma viene utilizzata una sequenza di riferimento il più possibile simile alle letture disponibili, che permette di assemblare i frammenti più facilmente tramite allineamento [7].

De novo assembly Se non è disponibile una sequenza di riferimento appropriata, i frammenti vengono legati insieme identificando pattern sovrapponibili e formando più *contig*, che vengono poi combinati con altre tecniche algoritmiche [8].

Ogni frammento viene quindi allineato con gli altri shotgun disponibili, formando una sequenza comune, il *consensus*; ciascuna base della sequenza assemblata ha una certa copertura (*coverage*), che è pari al numero di letture che contribuiscono al posizionamento della base nel consensus. La figura 1.1 mostra un semplice esempio di questo metodo.

Figura 1.1: Esempio di allineamento di shotgun per la formazione del consensus.

1.1.3 Metodi di seconda e terza generazione

I metodi di seconda generazione, comunemente chiamati *NGS - Next Generation Sequencing*, introducono un'elevata parallelizzazione del sequenziamento. Sostanzialmente, i vari metodi disponibili (come ad esempio *454*, *Illumina* o *Ion Torrent*) prevedono inizialmente la frammentazione del genoma da sequenziare, e l'unione di particolari sequenze ai frammenti creati. Essi vengono quindi amplificati tramite *emPCR* o *bridge amplification* [9], e possono essere sequenziati con letture parallele, spesso facendo uso di molecole fluorescenti. Segue quindi l'assemblamento delle letture ottenute per la ricostruzione della sequenza originale.

Il sequenziamento con metodi di terza generazione, quali *Nanopore* o *MinION*, si basano sul sequenziamento a singola molecola [5]. Pur mostrando un'efficienza maggiore rispetto ai metodi precedenti, sono attualmente ancora in fase di sviluppo.

1.2 Stima della dimensione del genoma

Il problema della misurazione della dimensione del genoma costituisce un argomento di interesse, perché oltre a fornire informazioni sulla sua evoluzione [10], permette di approssimare la quantità di dati che verranno prodotti nel sequenziamento e di valutare la complessità delle sequenze assemblate [11].

Di seguito sono elencate le due principali tipologie di metodi per la stima della dimensione del genoma, basate rispettivamente su approcci sperimentali o computazionali.

1.2.1 Metodi sperimentali

Inizialmente la ricerca scientifica ha cercato di stimare la dimensione del genoma con approcci biochimici, come ad esempio i metodi *Feulgen photometry* o *flow cytometry*. Tali metodologie però, oltre ad essere costose dal punto di vista economico e poco efficienti, devono basarsi su genomi specifici di riferimento [11, 12].

1.2.2 Metodi computazionali

Grazie all'aumento delle capacità computazionali, sono stati sviluppati metodi che possono calcolare la lunghezza del genoma utilizzando i dati ricavati dall'assemblaggio di shotgun. Dato che i dati assemblati sono di solito incompleti, è più conveniente cercarne di stimare la dimensione, utilizzando ad esempio i *k-mer*.

In questa trattazione verranno analizzati e confrontati vari approcci algoritmici che utilizzando i k-mer compiono la stima della dimensione del gemoma.

1.3 Sequenze di lunghezza k: i k-mer

I *k-mer* sono tutte le sottostringhe di lunghezza k presenti nella sequenza del genoma [13]. Si prenda ad esempio la sequenza descritta in 1.1.

$$AGATTCGC \tag{1.1}$$

I k-mer di lunghezza $k = 1$ saranno le quattro basi che formano la sequenza: G, T, A, C. Ponendo invece $k = 2$, i k-mer trovati saranno tutte le coppie formate da due basi: $AG, GA, AT, TT, TC, CG, GC$.
Allo stesso modo, per $k = 3$ le sequenze sono: $AGA, GAT, ATT, TTC, TCG, CGC$.

Il listing 1.1 nella pagina seguente mostra una semplice implementazione in pseudocodice per determinare i k-mer di lunghezza `k`, iterando la sequenza di input `seq` e dando in output tutte le sottostringhe di lunghezza `k` presenti in essa.

```
procedure k-mers(seq, k)
  lunghezza = length(seq)
  arr = array di L - k + 1 stringhe vuote

  // La sequenza iniziale viene iterata,
  // salvando il k-mer n-esimo nell'array di output
  for n = 0 to L - k + 1 escluso do
    arr[n] = sottostringa di seq da seq[n] a seq[n+k] escluso

  return arr
```

Listing 1.1: Algoritmo in pseudocodice per la costruzione dei k-mer.

Dato che il numero di k-mer aumenta esponenzialmente all'aumentare del parametro `k`, sono necessari algoritmi più complessi per il calcolo dei k-mer, come ad esempio *Jellyfish* [13] o *KMC2* [14].

1.3.1 K-mer profile

Il *k-mer profile*, detto anche *k-mer spectrum*, rappresenta un indicatore della complessità del genoma preso in esame. Date in input le letture shotgun del genoma, esso mostra il numero di volte che ogni k-mer viene trovato rispetto la quantità di k-mer distinti presenti, ovvero la molteplicità di ciascun k-mer nella sequenza rispetto il numero di k-mer con quella molteplicità [15]. Un esempio di k-mer profile è mostrato dalla figura 1.2 tratta da [16], in cui si può notare come la natura del genoma influenzi direttamente il grafico in ogni sua componente.

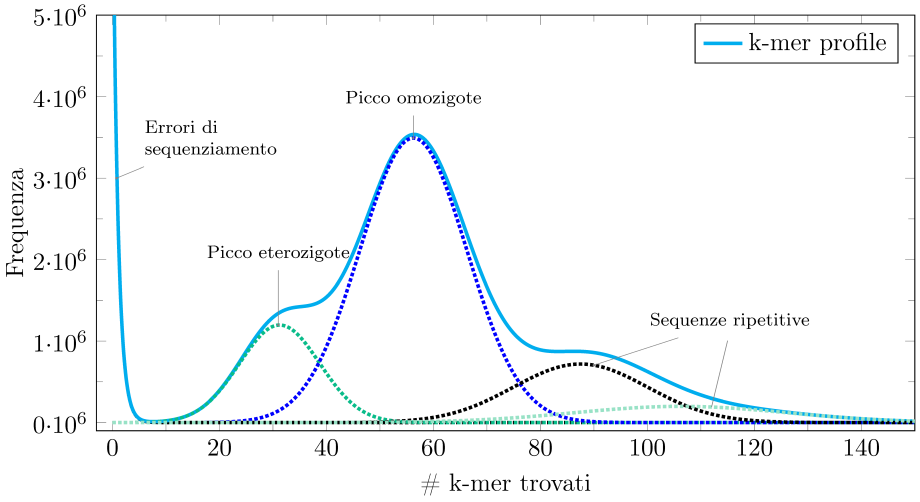


Figura 1.2: Composizione di un k-mer profile.

Ipotizzando che il genoma sia ideale, omozigote e senza ripetizioni, e che le letture siano state fatte senza errori con una certa copertura, il grafico del k-mer profile sarà una distribuzione di Poisson centrata sulla copertura media disponibile [17].

In casi reali invece, il genoma sarà eterozigote con una certa percentuale di eterozigosi e saranno presenti errori di sequenziamento; il k-mer profile presenterà tre picchi

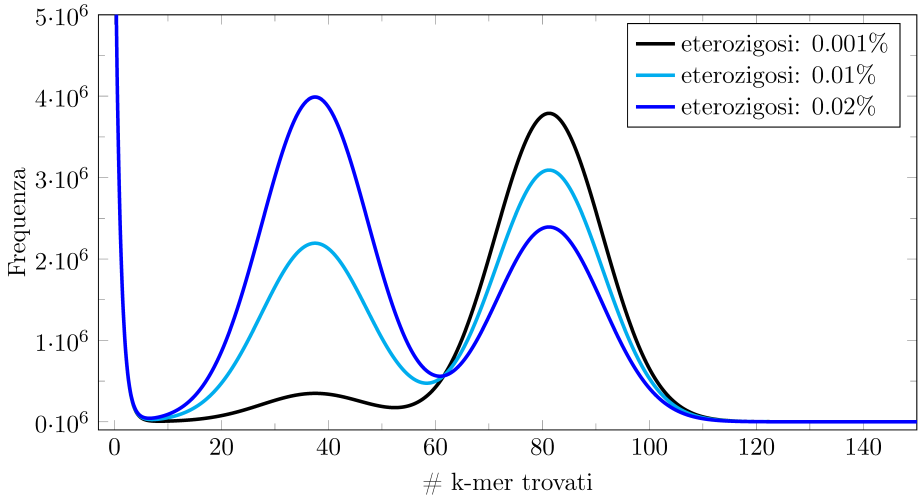


Figura 1.3: Variazione del grafico del k-mer profile al variare dell'eterozigosi.

principali [11]. Il primo picco del grafico corrisponde ai k-mer derivati da errori di sequenziamento, che accadono spesso ma che hanno bassa frequenza perché presentano poche occorrenze nelle letture di input; il secondo invece, rappresenta i k-mer eterozigoti e il terzo quelli omozigoti, presenti quindi su uno o entrambi gli alleli del set di cromosomi. I k-mer eterozigoti devono essere trattati più attentamente, perché possono risultare simili a quelli del primo picco, derivanti da errori di sequenziamento [16].

La lunga coda della distribuzione rappresenta invece le sequenze ripetitive, che occorrono con alta frequenza e sono presenti in un elevato numero di locus. Eventuali ripetizioni aggiungono al grafico ulteriori picchi, mentre errori nelle letture aumentano la varianza e producono distorsioni nel grafico.

La figura 1.3 tratta da [18] mostra come all'aumentare del rapporto di eterozigosi la quantità di k-mer eterozigoti del secondo picco diventi dominante rispetto ai k-mer omozigoti del terzo picco, che invece diminuiscono.

Capitolo 2

Metodi analizzati

In questo capitolo vengono presentati, in ordine cronologico di pubblicazione, i metodi presi in esame. Ciascun programma viene descritto svolgendo un'analisi dell'algoritmo che lo caratterizza, ed elencando le funzionalità previste per la gestione di eventi particolari, come ad esempio la presenza di errori di sequenziamento.

2.1 ALLPATHS-LG

ALLPATHS-LG è un programma che permette di eseguire il sequenziamento di un genoma tramite de novo assembly di letture shotgun, e che calcola implicitamente la dimensione totale del genoma. Esso si basa sul programma *ALLPATHS* [19, 20] sviluppato precedentemente e, rispetto al suo predecessore, permette l'assembly di genomi di dimensioni maggiori e con copertura minore, di gestire sequenze ripetitive, di correggere errori di lettura e di utilizzare in modo più efficiente le risorse disponibili durante il sequenziamento [21].

2.1.1 Algoritmo

L'algoritmo del programma si basa sul precedente software ALLPATHS [19]. Dato un numero minimo k di basi che si sovrappongono nelle letture shotgun, si definisce *branch* una sequenza di k basi (k-mer) che compare in due o più letture diverse, e la cui base successiva o precedente è diversa in ogni lettura. Spezzando il genoma in corrispondenza di ciascun branch, esso viene scomposto in un insieme di sequenze, dette *unipath*. Tali sequenze vengono create a partire dalle letture shotgun di input non allineate.

Formazione dei k-mer path Inizialmente negli shotgun viene corretto il maggior numero di errori di lettura, e vengono poi riconosciuti tutti i k-mer di lunghezza k. In ogni sequenza, ciascun k-mer viene numerato con un numero intero unico; a k-mer già trovati viene assegnato lo stesso valore. In questo modo ciascuna lettura potrà essere espressa come una sequenza di numeri interi, ognuno dei quali rappresenta un k-mer.

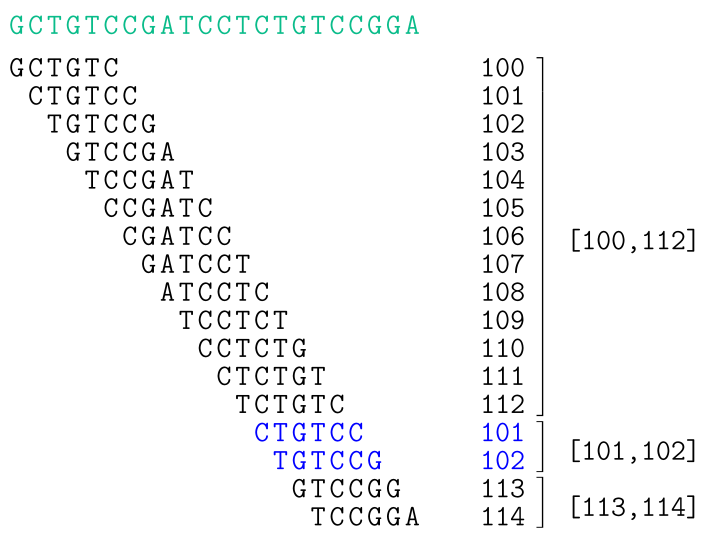


Figura 2.1: Esempio di numerazione dei k-mer e traduzione della sequenza in intervalli. Posto $k = 6$, vengono individuati tutti i k-mer presenti, ognuno dei quali viene numerato con un numero unico, riutilizzandolo nel caso di k-mer ripetuti. Usando gli intervalli, la sequenza iniziale viene quindi tradotta in $([100, 112], [101, 102], [113, 114])$.

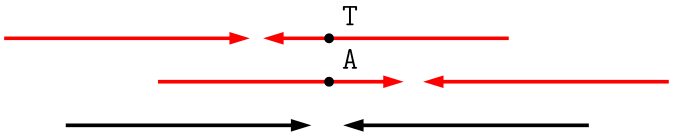
I numeri della sequenza vengono poi raggruppati in intervalli, in modo da formare i *k-mer path*; essi associano ciascun intervallo al k-mer path a cui esso appartiene, permettendo una facile ricerca di tutte le sequenze che condividono un certo k-mer e semplificando quindi l'assembly delle letture shotgun. Si veda la figura 2.1 per un esempio di come avviene la numerazione dei k-mer e la creazione dei k-mer path. Le letture così tradotte costituiscono il database in cui è possibile effettuare ricerche per la costruzione degli unipath.

Formazione degli unipath Tutti i numeri degli intervalli trovati nei k-mer path vengono scanditi. L'obiettivo per ogni numero è trovare il più lungo intervallo senza interruzioni che lo contenga. Si cercano nel database tutti gli intervalli che contengono quel numero, e si sceglie l'intervallo continuo più lungo, che diventa un *unipath interval*. Il processo è ripetuto per tutti i k-mer che non sono stati ancora inclusi in un unipath interval.

Per ogni unipath interval viene preso il primo numero di k-mer; esso viene cercato nel database e, a partire da quest'ultimo, si determina, se presente, un suo possibile predecessore. Se ne esiste esattamente uno, l'unipath interval del primo numero viene collegato all'intervallo alla sua sinistra. Procedendo iterativamente, viene formato un unipath, che consiste quindi in un gruppo di k-mer contigui.

Isolamento dei seed unipath Tra gli unipath creati, è possibile isolare dei *seed unipath* attorno ai quali poi eseguire l'assembly. Preso l'insieme di tutti gli unipath, iterativamente si rimuovono alcuni di essi; quindi, preso un certo unipath u, si isolano quelli adiacenti a destra e a sinistra rispetto a u. Tra i due vicini di u viene dedotta la distanza; se essa è minore di una certa soglia, u può essere rimosso dall'insieme. Si procede in questo modo

(a) *L'allineamento della coppia di letture in nero è possibile usando un'altra coppia di letture, raffigurate in rosso.*



(b) *Le due coppie di letture in rosso potrebbero allineare la coppia di letture in nero, ma presentano un SNP. Vengono quindi mantenute entrambe, fornendo due diverse soluzioni possibili.*

Figura 2.2: Esempi grafici del processo di gap filling.

per tutti gli unipath presenti finché continua a essere possibile la rimozione. Gli unipath rimasti costituiranno un seed unipath.

Assembly locale e globale Attorno ai seed unipath viene quindi assemblato il vicinato (*neighborhood*), cioè le regioni di 10 kb che precedono e seguono il seme. Per farlo, vengono creati due gruppi di sequenze, il primo (*primary read cloud*) contenente letture la cui posizione reale è vicina al seme, mentre il secondo (*secondary read cloud*) contiene brevi frammenti la cui sequenza può essere assemblata con le letture del primo gruppo. Il vicinato dei semi viene quindi assemblato utilizzando i due gruppi di letture, formando un *sequence graph* dell'assembly locale, cioè attorno al seed unipath corrispondente.

I vari assembly locali vengono fatti in parallelo e sono poi uniti per formare un sequence graph unico, relativo cioè all'assembly globale.

2.1.2 Gestione di genomi di grandi dimensioni

I predecessori di ALLPATHS-LG ottengono risultati promettenti solo per genomi di piccole dimensioni. Per gestire genomi di dimensioni più grandi, come quelli dei mammiferi, sono state fatte delle modifiche notevoli [21].

Il programma cerca di comprimere le ripetizioni in modo da favorirne l'allineamento. Se una sequenza ripetitiva è presente in due letture separate, il programma utilizza un'altra coppia di shotgun per poter allineare le prime due (*gap filling*). Le letture vengono unite se un'altra coppia fornisce una sovrapposizione, e viceversa. Il metodo può essere utilizzato anche se sono presenti mutazioni SNP, che forniscono più soluzioni di assemblamento. La figura 2.2 tratta da [21], mostra come avviene la gestione delle sequenze ripetitive nei due diversi casi.

La nuova versione del programma, inoltre, migliora la correzione degli errori di lettura: dato un k-mer, vengono analizzate tutte le letture che lo contengono; nel caso in cui una

singola lettura differisca dalla maggior parte delle altre, essa viene corretta se non ci sono voti in conflitto, altrimenti non viene modificata.

Ulteriori miglioramenti sono stati applicati anche per la gestione di sequenze a bassa copertura: dato che in questi casi la sovrapposizione tra letture può essere breve, in tali zone è preferibile utilizzare k-mer di dimensioni minori cioè scegliere un valore di k più basso. Il programma permette di utilizzare un valore di $k > 15$, ma solo nelle regioni che verrebbero assegnate a uno spazio vuoto tra altre due sequenze.

2.1.3 Stima della dimensione del genoma

Il programma, pur avendo come scopo primario la costruzione dell'assembly a partire dalle letture del genoma, produce in output un valore stimato della sua dimensione. Per farlo, dato il grafico del k-mer profile, esso identifica l'ascissa del punto di flesso f_v che si trova tra il primo e il secondo picco, cioè tra quello dei k-mer derivanti da errori di sequenziamento e quello dei k-mer omozigoti, e l'ascissa f_p corrispondente al picco eterozigote [11].

La dimensione del genoma G viene quindi stimata considerando gli N k-mer con copertura C compresi tra f_v e $3f_p/2$, scartando quindi dalle letture i k-mer a frequenze basse o molto alte, tramite la formula $G = N/C$.

2.2 GCE

Il programma *GCE* permette di modellare la distribuzione dei k-mer presenti nei dati sequenziati e di stimare la dimensione del genoma, le sequenze ripetitive e il rapporto di eterozigosi [22]. Il programma è open-source, scritto in C/C++ e si basa sulla statistica bayesiana.

2.2.1 Algoritmo

Conteggio dei k-mer Il programma fa uso del k-mer profile delle letture disponibili per stimare le caratteristiche del genoma. Inizialmente deve essere determinato il valore ottimale di k, che deve risultare grande abbastanza da far apparire ogni k-mer quasi unico all'interno del genoma, ma allo stesso tempo il più piccolo possibile per evitare un uso eccessivo di memoria durante l'estrazione dei k-mer. Per questo di solito si preferisce un valore k tale che $4^k > 5 \times G$, con G dimensione del genoma. In base al valore k scelto, può essere calcolato il k-mer profile. Il programma utilizza, oltre alle letture del genoma da sequenziare, un genoma di riferimento (*reference genome*).

Modello ideale Inizialmente, ipotizzando che entrambi i genomi siano ideali, cioè che il reference genome sia una sequenza casuale, senza eterozigosi o ripetizioni, e che le letture del genoma da sequenziare siano di uguale lunghezza e senza errori di sequenziamento,

la distribuzione delle frequenze dei k-mer seguirà una distribuzione di Poisson [17]. Considerando i valori n_{base} e n_{k-mer} il numero totale di basi e k-mer presenti nelle letture, e rispettivamente $c_{base} = n_{base}/G$ e $c_{k-mer} = n_{k-mer}/G$ le coperture per le basi e i k-mer nelle letture da sequenziare, una lettura di L basi genera $L - k + 1$ k-mer, quindi $n_{k-mer}/n_{base} = (L - k + 1)/L$. Si ricavano di conseguenza l'equazione 2.1 per il calcolo della copertura delle basi, e l'equazione 2.2 per la lunghezza del genoma.

$$c_{base} = c_{k-mer} \times L/(L - k + 1); \tag{2.1}$$

$$G = n_{k-mer}/c_{k-mer} = n_{base}/c_{base}. \tag{2.2}$$

Utilizzando i simboli $n = n_{k-mer}$ e $c = c_{k-mer}$, l'algoritmo deve determinare entrambi i parametri, per poi calcolare la lunghezza del genoma tramite l'equazione 2.2.

Il valore n può essere dedotto facilmente dal k-mer profile, ricavato il precedenza, come numero totale dei k-mer trovati. Per il valore della copertura c invece, vengono utilizzate le formule delle distribuzioni di Poisson 2.3 e 2.4, che descrivono rispettivamente il k-mer profile, il quale rappresenta il numero di specie di k-mer ($P_{Kspecies}$), e il prodotto tra i valori del numero di specie di k-mer e la copertura, che rappresenta il numero di individui di k-mer ($P_{Kindividuals}$); x si riferisce alla copertura del k-mer profile.

$$P_{Kspecies}(x) = \frac{c^x}{x!}e^{-c}; \tag{2.3}$$

$$P_{Kindividuals}(x) = xP_{Kspecies}(x)/c = P_{Kspecies}(x - 1). \tag{2.4}$$

Dalle curve delle distribuzioni, è possibile stimare il valore di c tramite una delle due equazioni 2.5 o 2.6.

$$c = \frac{P_{Kspecies}(x + 1)}{P_{Kspecies}(x)}(x + 1); \tag{2.5}$$

$$c = \frac{P_{Kindividuals}(x + 1)}{P_{Kindividuals}(x)}x. \tag{2.6}$$

Gestione di sequenze ripetitive Nella maggior parte dei casi un genoma reale contiene sequenze ripetitive, le quali aumentano la difficoltà del sequenziamento. Viene quindi utilizzato come riferimento un genoma già sequenziato, nel quale i k-mer vengono classificati in base alla frequenza genomica i (*genomic frequency*), che rappresenta la frequenza dei k-mer nel genoma di riferimento. Per ciascuna classe i vengono calcolati il rapporto delle specie di k-mer ($a_i = n_{i,genomic,Kspecies}/n_{genomic,Kspecies}$) e il rapporto di k-mer individuali ($b_i = n_{i,genomic,Kindividuals}/n_{genomic,Kindividuals}$). Le ripetizioni causano più picchi nel k-mer profile, che possono essere rappresentati come composizione di distribuzioni di Poisson. Le formule delle curve diventano quindi somma di distribuzioni di Poisson,

descritte dalle formule 2.7 e 2.8.

$$P_{Kspecies}(x) = \sum_{i=1}^{m} a_i \times P_{Kspecies,i}(x); \tag{2.7}$$

$$P_{Kindividuals}(x) = \sum_{i=1}^{m} b_i \times P_{Kindividuals,i}(x). \tag{2.8}$$

La stima del parametro a_i viene fatta iterativamente basandosi su un modello bayesiano, utilizzando la formula 2.9 in cui il termine v_j rappresenta la probabilità della specie di k-mer con copertura j, mentre l'espressione $P(j|i)$ rappresenta la probabilità che una specie di k-mer con frequenza i sia trovata j volte nelle letture da sequenziare. Per il parametro c invece è stata ricavata la formula 2.10, che permette di stimare la dimensione del genoma tramite la formula $G = n_{k-mer}/c$, in cui il termine n_{k-mer} rappresenta il numero totale di k-mer.

$$a_{i,t+1} = \sum_{j=0}^{w} \frac{a_{i,t}P(j|i)}{\sum_{i=1}^{m} a_{i,t}P(j|i)} \times v_j; \tag{2.9}$$

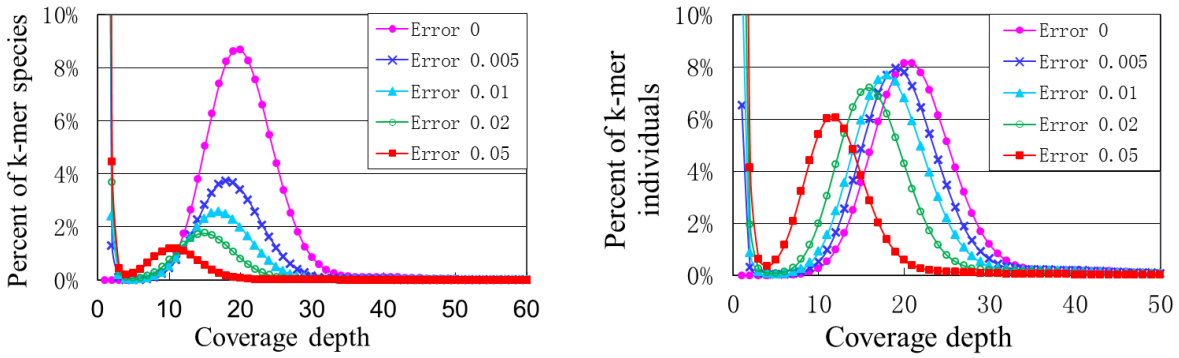
$$c_{t+1} = \frac{P_{Kspecies,1,t+1}(x+1)}{P_{Kspecies,1,t+1}(x)} \times (x+1). \tag{2.10}$$

Sequenziamento di genomi eterozigoti Sequenziando genomi diploidi reali, risulta utile stimare il rapporto di eterozigosi, dovuto a mutazioni SNP. Posto che i k-mer eterozigoti si distribuiscano in modo sparso nel genoma, saranno presenti $2 \times k$ k-mer eterozigoti attorno a ciascun sito SNP, che daranno origine a un nuovo picco nella curva di distribuzione dei k-mer.

2.2.2 Errori di sequenziamento e coverage bias

La gestione degli errori di sequenziamento viene fatta considerando i valori stimati di c e a_i. Il grafico della distribuzione dei k-mer in questi casi, infatti, presenta un grande picco iniziale, che riduce la dimensione degli altri picchi e li sposta verso sinistra, come mostrato dalla figura 2.3 a fronte tratta da [22]. Tramite i suddetti valori che rappresentano rispettivamente la copertura e il rapporto delle specie di k-mer, è possibile ricostruire la curva della distribuzione e confrontarla con la curva reale, in modo da determinare il numero di k-mer corretti e rimuovere gli errori di sequenziamento.

I *coverage bias*, che corrispondono a deviazioni rispetto alla distribuzione uniforme delle letture [23], sono invece più difficili da gestire, poiché appiattiscono le curve rendendo più difficile l'individuazione dei picchi e la stima di c. Anche se in presenza di tali errori i k-mer non vengono campionati in maniera equiprobabile, lo spettro di campionamento risulta continuo. Il nuovo modo di rappresentare le distribuzioni di specie e di individui di k-mer diventa quindi un modello continuo, che viene approssimato con un modello discreto



(a) *Grafici delle specie di k-mer, al variare della percentuale di errore.*

(b) *Grafici degli individui di k-mer, al variare della percentuale di errore.*

Figura 2.3: Effetto degli errori di sequenziamento sui grafici delle specie e degli individui di k-mer. Il picco dei k-mer omozigoti viene appiattito e spostato a sinistra all'aumentare della percentuale di errore.

a livelli non uniformi a_k, nel quale la copertura c scelta corrisponde al maggior valore a_k, mentre il parametro a_i viene stimato come somma dei parametri a_k corrispondenti a ciascun picco. Una volta trovati i valori c e a_i, viene stimata la dimensione del genoma tramite la formula 2.2 a pagina 11.

2.3 GenomeScope

Il progetto open source *GenomeScope* cerca sia di stimare le caratteristiche del genoma completo, come ad esempio la sua lunghezza o il rapporto di eterozigosi, sia di determinare le proprietà delle letture di DNA che prende in input, come la copertura (*read coverage*) o l'error rate [18]. Il programma per determinare tali caratteristiche utilizza il k-mer profile del genoma preso in esame, descritto nella sezione 1.3.1 a pagina 4.

2.3.1 Algoritmo

Il programma effettua una regressione non lineare dei dati iniziali, generando un profilo che cerca di approssimare il k-mer profile reale. Prendendo in input le letture del genoma che si vuole studiare, esso crea un modello che approssima il più possibile il k-mer profile. La funzione $f(X)$ scelta per l'interpolazione delle frequenze dei k-mer trovati è la somma di quattro distribuzioni binomiali negative $Y \sim \mathcal{NB}(X; p, n)$, per rappresentare rispettivamente i k-mer eterozigoti trovati nel genoma diploide una volta (unici) o tre volte (duplicati), e i k-mer omozigoti di cui si trovano due occorrenze (unici) o trovati quattro volte (duplicati). La funzione $f(X)$ è descritta dall'equazione 2.11 nella pagina successiva, in cui G rappresenta un coefficiente di scala legato alla dimensione del genoma,

λ e ρ sono rispettivamente la media e la varianza della distribuzione.

$$f(X) = G \times (\alpha\mathcal{NB}(X;\lambda,\lambda/\rho) + \beta\mathcal{NB}(X;2\lambda,2\lambda/\rho)+ \\ \gamma\mathcal{NB}(X;3\lambda,3\lambda/\rho) + \delta\mathcal{NB}(X;4\lambda,4\lambda/\rho)). \quad (2.11)$$

I coefficienti α, β, γ e δ dipendono dai parametri r e d, che rappresentano rispettivamente il rapporto di eterozigosi, cioè la percentuale di basi che sono specifiche a uno o due cromosomi omologhi, e la percentuale del genoma che è presente in due copie.

Lo scopo del programma è quindi determinare i coefficienti r, d, λ e ρ, oltre alla dimensione totale del genoma G. La funzione scelta $f(X)$, tramite cui poi può essere calcolata la dimensione del genoma, è quella che restituisce la minore somma dei quadrati degli errori residui (*Residual Sum of Square Error - RSSE*), che cioè minimizzi la somma tra i quadrati degli errori tra i valori osservati e quelli stimati, come descritto dall'equazione 2.12. Per dedurre i valori dei coefficienti, viene utilizzata la funzione `nls` del linguaggio di programmazione *R*, che compie il fitting dei dati alla funzione obiettivo.

$$RSSE = \sum_{x=E}^{+\infty} \left(kmer_{obs}[x] - kmer_{pred}[x]\right)^2. \quad (2.12)$$

Al termine, il programma mostra all'utente i dati relativi al genoma trovati, come il rapporto di eterozigosi, la media e la varianza della distribuzione, l'indice RSSE, che rappresenta la percentuale di k-mer non considerati dal modello, e la dimensione stimata del genoma.

La figura 2.4 nella pagina successiva mostra un confronto tra il k-mer profile reale e il modello costruito dal programma.

2.3.2 Gestione degli errori di sequenziamento

Eventuali errori di sequenziamento, ad esempio dovuti a duplicazioni con PCR o a sequenze contaminate, sono determinati solo empiricamente: dopo varie iterazioni del software in cui viene abbassata la soglia di copertura richiesta, i k-mer che non riescono ad essere rappresentati dal modello vengono identificati come errori di sequenziamento.

La stima dei k-mer sequenziati in modo errato è importante perché viene utilizzata per determinare la percentuale di basi errate nelle letture: una singola base inesatta infatti può dar luogo fino a k k-mer errati, aumentando notevolmente il numero di errori. GenomeScope permette una percentuale e di basi errate in ciascun k-mer. Tale valore è calcolato con un fitting dei k-mer errati a una distribuzione binomiale tramite la funzione `uniroot` del linguaggio di programmazione *R*. Questo metodo permette al programma di non dover assumere che la distribuzione degli errori di sequenziamento abbia una particolare forma, né di utilizzare un valore di soglia.

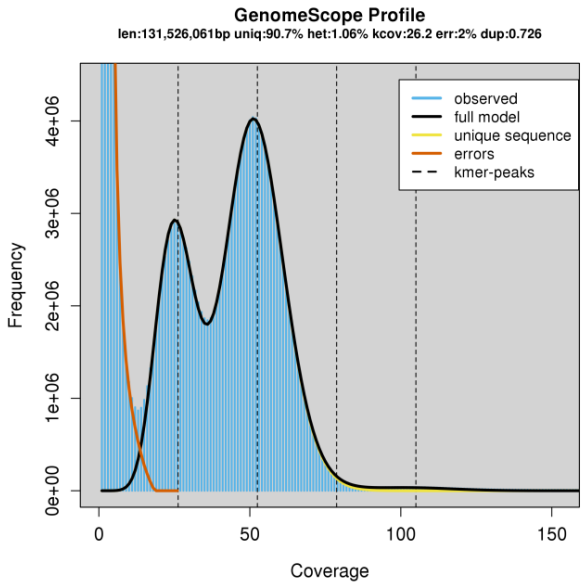


Figura 2.4: Modello del k-mer profile generato dal programma. L'istogramma in azzurro rappresenta i dati del k-mer profile reale, la curva nera il modello stimato e quella color arancione gli errori di sequenziamento stimati.

2.4 findGSE

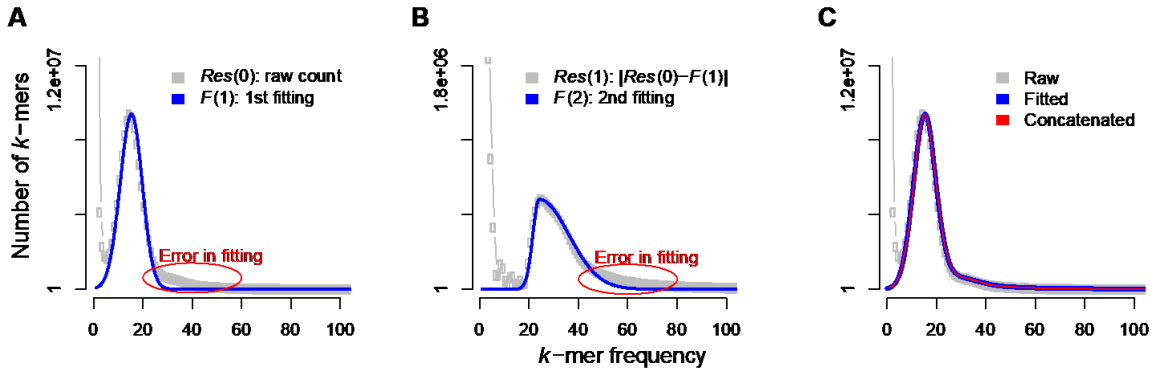
Il programma *findGSE* [11] ha come obiettivo principale la stima della lunghezza del genoma. Utilizzando le frequenze dei k-mer trovati nelle letture a disposizione, il programma compie una regressione non lineare dei dati utilizzando come funzione una distribuzione normale asimmetrica (*skew normal distribution* [24, 25]).

2.4.1 Algoritmo

Nel programma viene assunto che le frequenze dei k-mer possano essere approssimate da una distribuzione normale asimmetrica $Y \sim \mathcal{SN}(\xi, \omega^2, \alpha)$. Presa in input la distribuzione delle frequenze dei k-mer (k-mer profile), l'algoritmo effettua la regressione determinando i quattro parametri che descrivono una distribuzione normale asimmetrica, la media ξ, la deviazione standard ω, l'asimmetria α e un fattore di scala s. Ad ogni iterazione, il programma cerca di minimizzare l'errore tra i dati di input e la funzione stimata, in modo da approssimare il più possibile il k-mer profile reale. La figura 2.5 nella pagina seguente mostra come viene effettuato iterativamente il fitting del k-mer profile utilizzando come funzione la suddetta distribuzione.

Dato un genoma aploide con G basi, il numero di k-mer possibili sarà $G - k + 1$. Ponendo C la copertura media dei k-mer, in modo che in media ogni k-mer sia trovato in C letture diverse, e N il numero di k-mer trovati nelle letture, la quantità di k-mer presenti nel genoma è descritta dall'equazione 2.13.

$$N = C \times (G - K + 1). \tag{2.13}$$



(a) *Primo fitting dei dati dalle letture di input.*

(b) *Secondo fitting, utilizzando anche i k-mer residui dalla precedente operazione.*

(c) *Fitting finale.*

Figura 2.5: Esempio di fitting iterativo del k-mer profile.

Posta la dimensione del genoma molto maggiore del numero di basi utilizzate $G \gg k$, l'equazione 2.14 approssima la dimensione totale del genoma in analisi.

$$G \approx N/C. \tag{2.14}$$

A partire sia dal profilo reale che dal modello stimato, il programma calcola quindi il numero totale di k-mer trovati N e la copertura media dei k-mer C, per poi calcolare la dimensione del genoma attraverso l'equazione 2.14.

2.4.2 Analisi di genomi reali

Utilizzando letture reali, i dati devono essere precedentemente processati per minimizzare gli errori di stima del genoma. Per questo è consigliabile ridurre prima le letture con il programma *Skewer* [26], ed eliminare quelle con lunghezza minore di 33 bp. Per ottenere risultati migliori, inoltre, è consigliabile rimuovere le letture doppie dovute alla duplicazione PCR tramite il programma *FastUniq* [27], e le sequenze di simili a DNA dei mitocondri o dei cloroplasti con i programmi *BWA* [28] e *SAMtools* [29].

2.5 MGSE

Il programma *Mapping-based Genome Size Estimation* (*MGSE*) stima la dimensione del genoma attraverso il calcolo della copertura media a partire dall'assembly ad alta contiguità delle letture [12]. Lo script è open-source e scritto in Python, e processa le informazioni sulla copertura delle letture di input restituendo la dimensione stimata del genoma.

2.5.1 Algoritmo

Posto che le letture di input siano distribuite equamente sull'intera sequenza del genoma, il programma ne stima la dimensione calcolandone la copertura media. Se infatti sono noti il numero L di basi sequenziate e la copertura C in una certa posizione, la lunghezza totale N del genoma sarà pari a $N = L/C$.

Dato che le letture non forniscono la stessa copertura in tutte le regioni, ad esempio a causa di sequenze ripetitive, per una stima reale della dimensione del genoma è importante un calcolo corretto della copertura media. Per questo, il programma permette all'utente di inserire una lista di regioni di riferimento usate per il calcolo della media e della mediana della copertura. A questo scopo può essere utile il programma *Benchmarking Universal Single Copy Orthologs* (*BUSCO*) [30], che identifica un gruppo di sequenze *bona fidae*, cioè che sono a singola copia e che hanno solo una sola corrispondenza all'interno del genoma [31], le quali possono risultare adatte al calcolo della copertura media.

Il programma per poter effettuare la stima della dimensione del genoma riceve in input un file contenente la versione binaria compressa di sequenze allineate (file *bam* [29]), oppure un file contenente informazioni sulla copertura per ogni porzione della sequenza da sequenziare (file *cov* [32]). Dopo aver calcolato la media e la mediana della copertura, il programma produce in output le dimensioni del genoma stimate per entrambi i valori.

Capitolo 3

Confronto tra i metodi

In questo capitolo TODO

3.1 Complessità e performance

Vengono analizzati per ciascun metodo i ..., criteri utilizzati per numero di file/LOC (solo codice, esclusi commenti/bianchi) TODO

3.1.1 ALLPATHS-LG

Dato che il programma ha come obiettivo principale l'assembly delle letture di input, esso presenta una complessità elevata (1026 file, 215400 righe di codice). Inoltre sono necessarie prestazioni elevate (48 processori, 512 GB di RAM) e tempi lunghi di processamento TODO

Non è il metodo migliore perché richiede grande sforzo computazionale, ma può servire da reference TODO.

Glossario

Distribuzione normale asimmetrica TODO Una distribuzione normale asimmetrica.. 15

Distribuzione binomiale negativa TODO Una distribuzione.. 13

Distribuzione di Poisson TODO Una distribuzione.. 4

Fitting TODO Una distribuzione.. 14

Locus TODO Una distribuzione.. 5

Rapporto di eterozigosi TODO Una distribuzione.. 5

SNP TODO Una distribuzione.. 9, 12

Bibliografia

[1] J. Shendure e E. L. Aiden. «The expanding scope of DNA sequencing». In: *Nature Biotechnology* 30.11 (nov. 2012), pp. 1084–1094. ISSN: 1546-1696. DOI: 10.1038/nbt.2421.

[2] F. Sanger, S. Nicklen e A. R. Coulson. «DNA sequencing with chain-terminating inhibitors». In: *Proceedings of the National Academy of Sciences* 74.12 (gen. 1977), pp. 5463–5467. DOI: 10.1073/pnas.74.12.5463.

[3] F. Sanger et al. «Nucleotide sequence of bacteriophage ϕX174 DNA». In: *Nature* 265.5596 (feb. 1977), pp. 687–695. ISSN: 1476-4687. DOI: 10.1038/265687a0.

[4] A. M. Maxam e W. Gilbert. «A new method for sequencing DNA.» In: *Proceedings of the National Academy of Sciences* 74.2 (feb. 1977), pp. 560–564. DOI: 10.1073/pnas.74.2.560.

[5] J. Shendure et al. «DNA sequencing at 40: past, present and future». In: *Nature* 550.7676 (ott. 2017), pp. 345–353. DOI: 10.1038/nature24286.

[6] R. Staden. «A strategy of DNA sequencing employing computer programs». In: *Nucleic Acids Res* 6.7 (giu. 1979), pp. 2601–2610. DOI: 10.1093/nar/6.7.2601.

[7] B. Wajid e E. Serpedin. «Do it yourself guide to genome assembly». In: *Briefings in Functional Genomics* 15.1 (nov. 2014), pp. 1–9. ISSN: 2041-2649. DOI: 10.1093/bfgp/elu042.

[8] C. Noune. «Dynamics, diversity and evolution of Baculoviruses». Tesi di dott. Queensland University of Technology, 2017. DOI: 10.5204/thesis.eprints.113154.

[9] J. M. Heather e B. Chain. «The sequence of sequencers: The history of sequencing DNA». In: *Genomics* 107.1 (gen. 2016), pp. 1–8. DOI: 10.1016/j.ygeno.2015.11.003.

[10] T. R. Gregory. «Synergy between sequence and size in Large-scale genomics». In: *Nature Reviews Genetics* 6.9 (set. 2005), pp. 699–708. ISSN: 1471-0064. DOI: 10.1038/nrg1674.

[11] H. Sun et al. «findGSE: estimating genome size variation within human and Arabidopsis using k-mer frequencies». In: *Bioinformatics* 34.4 (ott. 2017), pp. 550–557. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx637.

[12] B. Pucker. «Mapping-based genome size estimation». In: *bioRxiv* (apr. 2019). DOI: 10.1101/607390.

[13] G. Marçais e C. Kingsford. «A fast, lock-free approach for efficient parallel counting of occurrences of k-mers». In: *Bioinformatics* 27.6 (gen. 2011), pp. 764–770. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btr011.

[14] S. Deorowicz et al. «KMC 2: fast and resource-frugal k-mer counting». In: *Bioinformatics* 31.10 (gen. 2015), pp. 1569–1576. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btv022.

[15] D. Mapleson et al. «KAT: a K-mer analysis toolkit to quality control NGS datasets and genome assemblies». In: *Bioinformatics* 33.4 (feb. 2017), pp. 574–576. DOI: 10.1093/bioinformatics/btw663.

[16] J. Sohn e J. Nam. «The present and future of de novo whole-genome assembly». In: *Briefings in Bioinformatics* 19.1 (ott. 2016), pp. 23–40. ISSN: 1477-4054. DOI: 10.1093/bib/bbw096.

[17] X. Li e M. S. Waterman. «Estimating the repeat structure and length of DNA sequences using L-tuples». In: *Genome Res* 13.8 (ago. 2003), pp. 1916–1922. DOI: 10.1101/gr.1251803.

[18] G. W. Vurture et al. «GenomeScope: fast reference-free genome profiling from short reads». In: *Bioinformatics* 33.14 (lug. 2017), pp. 2202–2204. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx153.

[19] J. Butler et al. «ALLPATHS: de novo assembly of whole-genome shotgun microreads». In: *Genome Res* 18.5 (mag. 2008), pp. 810–820. DOI: 10.1101/gr.7337908.

[20] I. Maccallum et al. «ALLPATHS 2: small genomes assembled accurately and with high continuity from short paired reads». In: *Genome Biol* 10.10 (ott. 2009), R103. DOI: 10.1186/gb-2009-10-10-r103.

[21] S. Gnerre et al. «High-quality draft assemblies of mammalian genomes from massively parallel sequence data». In: *Proceedings of the National Academy of Sciences* 108.4 (gen. 2011), pp. 1513–1518. DOI: 10.1073/pnas.1017351108.

[22] B. Liu et al. «Estimation of genomic characteristics by analyzing k-mer frequency in de novo genome projects». In: *arXiv e-prints* (ago. 2013), arXiv:1308.2012. DOI: 10.48550/ARXIV.1308.2012.

[23] M. G. Ross et al. «Characterizing and measuring bias in sequence data». In: *Genome Biology* 14.5 (mag. 2013), R51. ISSN: 1474-760X. DOI: 10.1186/gb-2013-14-5-r51.

[24] A. Azzalini. «A Class of Distributions Which Includes the Normal Ones». In: *Scandinavian Journal of Statistics* 12.2 (1985), pp. 171–178. ISSN: 03036898, 14679469. URL: http://www.jstor.org/stable/4615982 (visitato il 05/08/2022).

[25] A. Azzalini. «The Skew-normal Distribution and Related Multivariate Families». In: *Scandinavian Journal of Statistics* 32.2 (mag. 2005), pp. 159–188. DOI: 10.1111/j.1467-9469.2005.00426.x.

[26] H. Jiang et al. «Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads». In: *BMC Bioinformatics* 15.1 (giu. 2014), p. 182. ISSN: 1471-2105. DOI: 10.1186/1471-2105-15-182.

[27] H. Xu et al. «FastUniq: A Fast De Novo Duplicates Removal Tool for Paired Short Reads». In: *PLOS ONE* 7.12 (dic. 2012), pp. 1–6. DOI: 10.1371/journal.pone.0052249.

[28] H. Li e R. Durbin. «Fast and accurate short read alignment with Burrows–Wheeler transform». In: *Bioinformatics* 25.14 (mag. 2009), pp. 1754–1760. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btp324.

[29] H. Li et al. «The Sequence Alignment/Map format and SAMtools». In: *Bioinformatics* 25.16 (giu. 2009), pp. 2078–2079. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btp352.

[30] F. A. Simão et al. «BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs». In: *Bioinformatics* 31.19 (giu. 2015), pp. 3210–3212. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btv351.

[31] W. Li et al. «TruMatch—a BLAST post-processor that identifies bona fide sequence matches to genome assemblies». In: *Bioinformatics* 21.9 (gen. 2005), pp. 2097–2098. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bti257.

[32] B. Pucker e S. F. Brockington. «Genome-wide analyses supported by RNA-Seq reveal non-canonical splice sites in plant genomes». In: *BMC Genomics* 19.1 (dic. 2018), p. 980. ISSN: 1471-2164. DOI: 10.1186/s12864-018-5360-z.