

Sense_Puzzle

Mattia d'Argenio, Davide Di Sarno

Gennaio 2024

Sommario

Nell'ambito della bioinformatica, l'allineamento di sequenze genomiche è una procedura critica per l'analisi comparativa di DNA, RNA e proteine, essenziale per la comprensione delle funzioni biologiche, delle relazioni evolutive e della genetica delle malattie. Tuttavia, data la vastità e la complessità dei dataset genomici, l'allineamento tradizionale presenta significative sfide computazionali. Questo studio introduce un approccio innovativo che sfrutta le reti neurali siamesi per affrontare le limitazioni dei metodi convenzionali, offrendo un mezzo per accelerare e ottimizzare l'allineamento di sequenze genomiche. L'obiettivo principale del nostro lavoro è stato quello di sviluppare un'euristica che potesse servire come strumento di valutazione preliminare, o "oracolo", per guidare metodi computazionalmente più efficienti nell'identificazione degli allineamenti ottimali. Questo approccio si basa sull'ipotesi che, attraverso l'apprendimento profondo, possiamo catturare le relazioni complesse tra le sequenze in uno spazio di embedding ridotto, consentendoci di approssimare la somiglianza tra le sequenze in maniera più veloce e con risorse computazionali ridotte. Abbiamo implementato e addestrato il nostro modello su un dataset di sequenze genomiche, utilizzando la perdita contrastiva per ottimizzare gli embedding in modo che riflettano accuratamente le distanze genetiche. Attraverso l'applicazione di tecniche di Brute Force e di algoritmi genetici, abbiamo esplorato diversi scenari di allineamento, valutando la performance del nostro modello sia in termini di accuratezza che di efficienza computazionale. I risultati ottenuti dimostrano che l'uso delle reti neurali siamesi, combinate con strategie di ottimizzazione intelligente, può significativamente accelerare il processo di allineamento di sequenze genomiche mantenendo un'elevata precisione. In particolare, abbiamo osservato che il nostro approccio fornisce un'eccellente euristica per la selezione dei candidati all'allineamento, riducendo notevolmente il campo di ricerca e il tempo computazionale richiesto dai metodi di allineamento tradizionali.

Indice

1 Introduzione	4
1.1 Contestualizzazione del Progetto	4
1.2 Obiettivi del Progetto	4
2 Metodologia	6
2.1 Dataset	6
2.1.1 Preprocessing dei dati	6
2.1.2 Dataset artificiale	6
2.2 Rete Neurale Siamese	7
2.3 Rete Neurale Convoluzionale	7
2.4 Configurazione e training	9
2.5 Metodi di allineamento sperimentati	10
2.5.1 Approccio 'Brute Force'	10
2.5.2 Approccio 'Algoritmo genetico'	12
3 Sperimentazione	17
3.1 Stato dell'arte	17
3.1.1 Allineamento Pairwise con Bio.pairwise2	17
3.2 Sviluppo metodo personalizzato	18
3.3 Metodo di confronto con embeddings	19
3.4 Sottosequenze genetiche	20
4 Risultati	22
4.1 Risultati del modello	22
4.2 Confronto tra Metodi	22

1 Introduzione

1.1 Contestualizzazione del Progetto

L'analisi delle sequenze è una delle principali aree di ricerca della bioinformatica e ha un'ampia gamma di applicazioni nella ricerca nei database, nell'annotazione delle sequenze, nella metagenomica, nella genomica comparativa e nella predizione dei geni. Tradizionalmente, l'allineamento di sequenze, sia esso globale, locale, a coppie o multiplo, si affida a metodi consolidati come l'algoritmo di Needleman-Wunsch (NW), riconosciuto per la sua precisione ottimale. Tuttavia, la principale sfida di questi approcci classici risiede nella loro intensiva richiesta di risorse computazionali, specialmente nel trattamento di ampi dataset. Negli ultimi due decenni, sono stati sviluppati vari approcci legati al campo del Machine Learning per fornire alternative computazionalmente efficienti.

1.2 Obiettivi del Progetto

Il nostro progetto si è concentrato sull'elaborazione di un metodo avanzato per l'analisi di sequenze genomiche, attraverso l'implementazione di un modello innovativo denominato Sense. Questo modello, è stato progettato per ottimizzare l'allineamento di sequenze genomiche, assicurando la precisione e l'accuratezza del posizionamento delle sequenze fornite come input. Utilizzando un approccio innovativo che combina l'uso di reti siamesi con tecniche di Brute Force e algoritmi genetici, abbiamo sviluppato uno strumento in grado di identificare l'allineamento ottimale tra coppie di sequenze genomiche. Il cuore di questa ricerca risiede nella capacità di trasformare le sequenze in embedding vettoriali tramite reti siamesi, fornendo una rappresentazione densa e informativa che può essere successivamente utilizzata per valutare la somiglianza tra le sequenze. Il metodo Brute Force è stato adottato come punto di riferimento per valutare tutte le configurazioni possibili nell'allineamento delle sequenze, assicurando una copertura completa delle potenziali soluzioni. Questa scelta, pur essendo computazionalmente esigente, ha garantito l'accuratezza dell'analisi. Per mitigare l'onere computazionale derivante dall'utilizzo di questo metodo, il progetto ha sperimentato ed integrato un algoritmo genetico come strumento di ottimizzazione euristica. Attraverso i processi evolutivi quali selezione, crossover e mutazione, l'algoritmo ha facilitato un'esplorazione efficiente dello spazio delle soluzioni, identificando allineamenti ottimali o sub-ottimali. Questo approccio ha permesso di ridurre i tempi di computazione senza compromettere significativamente la qualità dell'allineamento.

L'obiettivo principale di questo progetto era dunque duplice: da un lato, sfruttare la capacità di Sense di generare rappresentazioni vettoriali profonde delle sequenze genomiche; dall'altro, incorporare questa ca-

pacità con algoritmi computazionalmente efficienti per l'identificazione rapida e accurata degli allineamenti. Attraverso questo approccio ibrido, abbiamo creato uno strumento di euristica che non solo migliora l'accuracy dell'allineamento di sequenze ma anche ne ottimizza il processo di identificazione, ponendosi come un potenziale candidato per applicazioni reali in bioinformatica e oltre.

2 Metodologia

2.1 Dataset

Per l'esperimento condotto, abbiamo focalizzato la nostra attenzione sul dataset di sequenze reali denominato Qiita, ottenuto da Qiita (studio n. 10052), che include 66 campioni di pelle, saliva e feci, con un totale di 6 734 572 sequenze di 151 bp, coprenti la regione ipervariabile V4 del gene 16S rRNA. Parallelamente, è emerso un secondo dataset denominato RT988, che copre le regioni V3-V4. Al fine di ottimizzare i tempi computazionali durante la fase di implementazione e progettazione, abbiamo optato per l'utilizzo di una demo, contenente 10 mila sequenze.

2.1.1 Preprocessing dei dati

Nella fase di preprocessing, abbiamo inizialmente elaborato i file di sequenza provenienti da Qiita nel formato FASTQ, i quali sono stati successivamente convertiti nel formato FASTA al fine di uniformare il dataset. Abbiamo proceduto con l'unione dei file di sequenza, creando così un unico insieme di dati coeso esclusivamente di Qiita. Successivamente, abbiamo eseguito una pulizia approfondita, eliminando i caratteri malformati non appartenenti alla sequenza nucleotidica e le sequenze troppo corte che potrebbero compromettere un allineamento significativo. L'obiettivo di questo processo era garantire la coerenza e l'uniformità del dataset proveniente da Qiita, preparandolo per le successive fasi di analisi e interpretazione.

2.1.2 Dataset artificiale

Nel quadro di una valutazione esaustiva del modello, è stato implementato un metodo per la creazione di un dataset fittizio. Questo dataset è composto da sequenze generate casualmente per testare la capacità del modello di gestire dati con un grado di ambiguità potenzialmente elevato, fornendo così un controllo aggiuntivo rispetto alle sequenze reali già utilizzate. Il processo di generazione del dataset fittizio ha incluso la selezione casuale di sequenze e la loro combinazione in modelli di coppie, simulando la diversità e la complessità che si potrebbero riscontrare in campioni reali. In questo modo, si è potuto osservare il comportamento del modello di rete neurale siamese di fronte a un'ampia varietà di scenari fittizi, valutando così la sua capacità di distinguere e allineare correttamente le sequenze in presenza di disturbo e variazione casuale. Questa strategia è stata adottata con lo scopo di testare la precisione e la robustezza di SENSE. In particolare, si è cercato di determinare se il modello addestrato potesse mantenere un elevato grado di precisione nell'allineamento

```

>MISEQ02:42:000000000-AAN9D:1:2114:11465:27347#3:N:0
CCTCTCGATCCCCATGTTCGCCCCCAGCGTCAGTAGTCGCCGGATAGCTGCCCTCGCTTGGCGTCCGTGTTAGATACGTCTTCACCCCTACC
CCACGGCACTTCGCCTCTCCTCTCCCTCACTATCCTGTCTTC
>MISEQ02:42:000000000-AAN9D:1:2114:14871:27348#3:N:0
CCTGTTGCTCCCAGCCTTCGAGCCTCAGCGTCAGTTACAAGGCCAGAGAGCCGCTTCGCCACCGGTGTTCCCATATCTCTACGCATTCACCGCTACA
CATGGAAATTCCACTCTCCCTTCTGCACTCAAGGTAACCGTCCCAA
>MISEQ02:42:000000000-AAN9D:1:2114:11485:27348#3:N:0
CCTCTTGCTCCCTATGCTCCGCCCGAGGCCAGGCCACCCAGGTAGTTGCTTTGTTCCCGTCTCCCTCACAGCCCTACGATCCACTGCTAAC
TGCCGAATTCCAATTCTCCCGTACCTCTAGTCTCCAGTTCAGAC
>MISEQ02:42:000000000-AAN9D:1:2114:14474:27348#3:N:0
TCTAATGTCGTCACTGCTGCTCTGGTGTGGTGTGATATTCTAGGTACCGCTGTTGCCGATACTTGGAAACAATTTCGGGAAAGACGGTAAAG
CTGATGGTATTGGCTCTTTGCTACTCAAGAACGGTCGGTATTAA
>MISEQ02:42:000000000-AAN9D:1:2114:15120:27348#3:N:0
TGACTCCGAAGTTAGTGCTGAGGTGACTTAGTCTCATCAGAACGCCAGAATCAGGGTATGGCTTCTCATATTGGCGCTACTGCAAAGGATATTCTAA
TGTCGCCACTGATGCTGGTGGCTGATATTCCCTCATGGT|

```

Figura 1: Porzione del dataset contenente le coppie Indice-Sequenza.

delle sequenze nonostante l'introduzione di dati non autentici e generati in modo arbitrario, verificando la sua effettiva applicabilità all'interno di contesti bioinformatici.

2.2 Rete Neurale Siamese

La Figura 1 offre uno sguardo approfondito sulla strategia proposta, introducendo la rete neurale Siamese. Questa architettura, caratterizzata da due bracci identici, opera simultaneamente sulla coppia di sequenze di DNA in input, generando vettori di embedding corrispondenti. La distanza di allineamento, calcolata con l'algoritmo Needleman-Wunsch, e la distanza di incorporamento, misurata attraverso la distanza di Jaccard generalizzata, sono fondamentali nel processo di addestramento. L'obiettivo centrale della rete Siamese è minimizzare la discrepanza ideale tra queste distanze, sfruttando la backpropagation per ridurre l'errore quadratico medio durante il training. La Figura 1 illustra chiaramente il flusso di questo processo di apprendimento per la rete neurale Siamese.

2.3 Rete Neurale Convoluzionale

Nel contesto del deep learning, le architetture principali utilizzate sono le reti neurali convoluzionali (CNN) e le reti neurali ricorrenti (RNN). Nell'ambito di questo studio, la preferenza è stata data alle CNN, poiché l'addestramento delle RNN è generalmente considerato più complesso, nonostante la capacità intrinseca delle RNN di rilevare modelli spaziali dinamici e processare sequenze di lunghezza variabile. Per adattarsi alle esigenze specifiche del caso di studio, è stato necessario uniformare la lunghezza delle sequenze di input, come precedentemente menzionato. Questa decisione è stata motivata dalla focalizzazione dell'analisi sulle

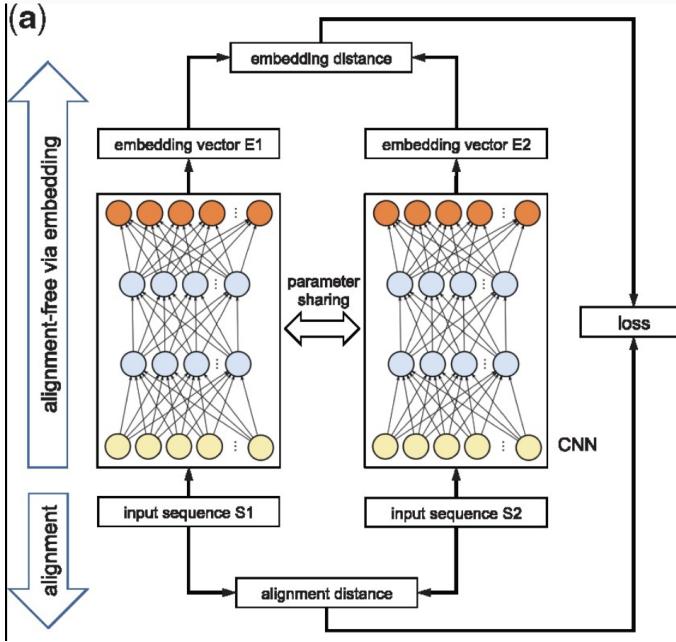


Figura 2: Processo di training della rete neurale siamese.

sequenze di ampliconi, le quali presentano tipicamente lunghezze molto simili. La Figura 2 offre un’illustrazione dettagliata della struttura progettata per le CNN in questione. Questa architettura è composta da tre moduli di convoluzione seguiti da uno strato di appiattimento e uno strato completamente connesso. Quando una sequenza di input viene introdotta nella rete, viene prima convertita in una matrice di dimensioni $L \times 4$, utilizzando la codifica one-hot. Successivamente, questa matrice alimenta uno strato di convoluzione (Figura 3). Durante la scansione della sequenza, entrambe le operazioni si concentrano esclusivamente sulla direzione della lunghezza. L’output di uno strato di convoluzione è denominato ”mappa delle caratteristiche”. Tipicamente, ogni strato di convoluzione presenta più filtri, ognuno dei quali genera un canale nella mappa delle caratteristiche. Ciascun valore della mappa viene poi elaborato attraverso la funzione ReLU. L’uscita dello strato ReLU passa quindi attraverso uno strato di max-pooling. Lo strato di max-pooling esegue un downsampling non lineare, suddividendo ciascun canale della mappa delle caratteristiche in rettangoli non sovrapposti e producendo il massimo per ogni sottoregione. Questo processo mira a ridurre il numero di parametri e il carico computazionale nella rete, controllando simultaneamente l’overfitting. Dopo l’elaborazione attraverso i tre moduli di convoluzione, la sequenza risultante viene inoltrata allo strato di appiattimento, dove le mappe delle caratteristiche precedenti vengono concatenate in un unico vettore. Infine, questo strato

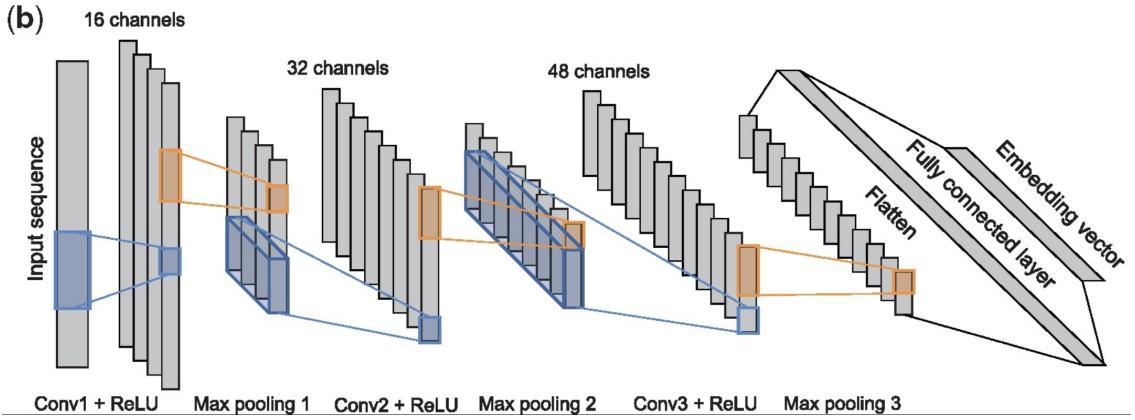


Figura 3: Struttura dettagliata della CNN utilizzata a tre livelli.

di appiattimento è completamente connesso allo strato di uscita, generando un vettore di embedding che rappresenta in modo significativo la sequenza elaborata dalla CNN.

2.4 Configurazione e training

Durante il processo di training del modello proposto, la rete neurale siamese è stata sottoposta a una serie di passaggi finalizzati all'apprendimento efficace delle rappresentazioni delle sequenze di ampliconi. Inizialmente, il numero di nodi nello strato di input è stato adeguato alla lunghezza massima delle sequenze di input, stabilendo una connessione diretta tra la struttura della rete e le caratteristiche delle sequenze trattate. La lunghezza del filtro, fissata a 5, e il numero di filtri nei tre strati di convoluzione, impostato rispettivamente a 16, 32 e 48, hanno definito le dimensioni dei filtri come 5×4 , 5×16 e 5×32 . Il processo di training è stato orchestrato attraverso la backpropagation, una tecnica di ottimizzazione che mira a minimizzare l'errore quadratico medio tra le previsioni del modello e i valori effettivi del target. Questo permette al modello di adattarsi progressivamente alle caratteristiche specifiche delle sequenze di ampliconi presenti nel dataset di addestramento. Nel dettaglio, per gli strati di convoluzione, sono state utilizzate impostazioni di padding pari a 2 e stride pari a 1, garantendo che la mappa delle caratteristiche in uscita conservasse la stessa lunghezza dell'input. Nei livelli di max-pooling, una finestra di dimensioni 2 e uno stride di 2 sono stati determinanti per l'esecuzione di un downsampling non lineare, riducendo il numero di parametri e controllando l'overfitting. Il numero di nodi nello strato di output è stato fissato a 2, mentre il numero di nodi nel layer di output, corrispondente alla dimensione dei vettori di embedding, è stato equiparato alla lunghezza delle sequenze

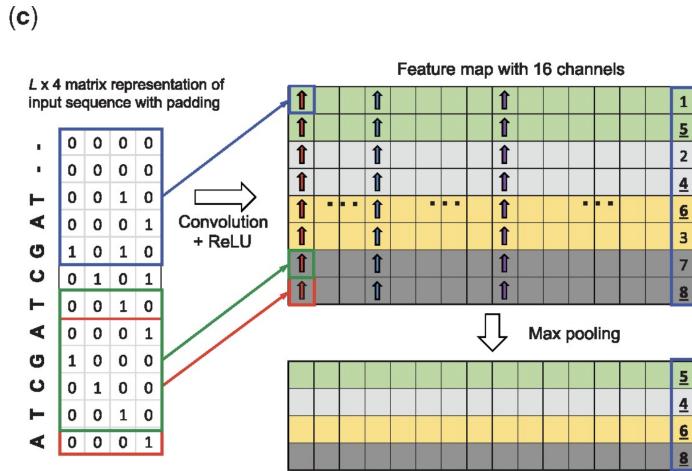


Figura 4: Esempio che illustra come vengono eseguite un'operazione di convoluzione e un max pooling su una sequenza.

di ingresso. Durante l’iterazione attraverso i dati di addestramento, la backpropagation ha regolato i pesi della rete in modo da minimizzare l’errore tra le previsioni del modello e i target noti, guidando la rete verso una rappresentazione più efficace delle relazioni tra le sequenze. Questo processo iterativo ha portato al continuo miglioramento delle prestazioni del modello, rendendolo in grado di catturare in modo ottimale le caratteristiche discriminanti delle sequenze di ampliconi.

2.5 Metodi di allineamento sperimentati

In questa sezione, esploreremo due approcci distinti per l’allineamento di sequenze: un algoritmo basato sulla forza bruta e un algoritmo genetico. Entrambi i metodi cercano di ottimizzare la somiglianza tra sequenze, ma seguono approcci diversi per raggiungere questo obiettivo.

2.5.1 Approccio ‘Brute Force’

L’approccio Brute Force nell’allineamento di sequenze genomiche, con l’ausilio della rete siamese, rappresenta una strategia dettagliata e meticolosa per analizzare e identificare l’allineamento ottimale tra coppie di sequenze. Questo metodo si basa sull’analisi esauriente di tutte le combinazioni possibili, utilizzando la potenza computazionale della rete siamese per valutare la somiglianza tra le sequenze in termini di embedding vettoriali.

```

Per la sequenza 7:
Miglior punteggio di allineamento: 0.4636295437812805
Sequenza corrispondente:
TAAGTGGCTAAAGATTGTAGATATCTAGCCAAGTGTACACGGCTTGTCAAGGCCACTCTAGTAGTCGTGGTAGAGTCGGAGAGCACCGGTAGAACCTTAAC
ATATGCTGGACGGCGGTGGCTGGAAGCAGTGTGATGGTGGCTAGC
Numero di sequenza corrispondente: 8 su 10 (percentuale: 80.00%)

```

Figura 5: Corrispondenza di una sequenza nell'approccio 'Brute Force'.

Il primo passo dell'approccio è **la generazione di embedding** per le sequenze, dove le sequenze genomiche vengono trasformate in rappresentazioni vettoriali dense attraverso la rete siamese. Questi embedding catturano l'essenza delle sequenze in uno spazio ridotto, preservando le informazioni cruciali per l'allineamento.

Nel passo successivo avviene il **calcolo della distanza tra embedding**. Una volta ottenuti gli embedding, si procede con il calcolo della distanza tra di essi utilizzando il modulo MaxMinout. Questo calcolo non si affida a metriche tradizionali come la distanza euclidea o coseno, ma piuttosto determina la distanza attraverso il rapporto tra maxout e minout, derivato dagli embedding. Questa specifica metodologia di misurazione della distanza quantifica la somiglianza tra le sequenze basandosi sulle loro rappresentazioni vettoriali profonde, con una distanza minore che indica una maggiore somiglianza.

Attraverso la **valutazione dell'allineamento** tramite Brute Force, l'algoritmo esamina ogni coppia possibile di sequenze, calcolando gli embedding per ciascuna e determinando la distanza tra questi. Ogni coppia riceve un valore di somiglianza basato su questa distanza, consentendo di valutare l'allineamento in modo obiettivo e quantificabile.

L'**identificazione dell'allineamento ottimale** avviene selezionando la coppia di sequenze che presenta il valore di somiglianza più alto, ovvero la minore distanza tra gli embedding. Questo processo garantisce la scelta dell'allineamento che massimizza la corrispondenza tra le sequenze, basandosi su una valutazione profonda e dettagliata fornita dall'analisi degli embedding.

L'approccio Brute Force con reti siamesi si distingue per la sua capacità di fornire una soluzione esaustiva e accurata all'allineamento di sequenze genomiche, pur richiedendo significative risorse computazionali. La trasformazione delle sequenze in embedding e il successivo calcolo della distanza tra questi rappresentano un avanzamento significativo rispetto ai metodi tradizionali, offrendo una nuova prospettiva sull'allineamento di sequenze e sulla valutazione della loro somiglianza. Questo metodo si rivela, quindi, prezioso in quanto risulta adatto al ruolo di 'oracolo' per il confronto con metodologie computazionalmente meno onerose.

2.5.2 Approccio 'Algoritmo genetico'

L'approccio genetico adottato trae ispirazione dai meccanismi dell'evoluzione naturale e della selezione delle specie per affrontare e risolvere problemi complessi mediante una strategia di ottimizzazione basata su concetti molto semplici che approfondiremo successivamente. Si parte da una popolazione di individui e si applicano gli operatori dell'algoritmo genetico che nel nostro caso sono il crossover e la mutazione. Questi due operatori vanno a generare coppie di sequenze che si vanno ad aggiungere alla popolazione iniziale e, per ogni individuo della popolazione, si va ad effettuare il calcolo della fitness. La funzione di fitness che noi abbiamo utilizzato è Sense, nello stesso modo in cui vengono valutate le sequenze nell' approccio Brute Force. L'idea vincente è stata quella di utilizzare Sense come fitness del nostro algoritmo genetico in modo che potessimo rendere il processo di valutazione degli allineamento il più accurato possibile. In questo caso il calcolo della fitness ci aiuta a capire quando bene una coppia di sequenze è allineata. Una volta applicati gli operatori genetici ci ritroveremo con una popolazione che è nettamente superiore a quella iniziale in termine di individui quindi andiamo a selezionare i migliori k individui in termini di fitness (con k uguale al numero iniziale della popolazione) per far sì che ogni generazione viene iterata sullo stesso numero di individui all'interno della popolazione. Inizialmente, l'algoritmo inizia con la creazione di una popolazione iniziale di individui. Ogni individuo è formato da coppie di sequenze genomiche estratte casualmente da un insieme di sequenze disponibili. Questa popolazione rappresenta la prima generazione. Attraverso iterazioni successive, l'algoritmo seleziona gli individui più adatti per formare le prossime generazioni in base alla funzione di fitness, sottoponendoli a processi di crossover e mutazione e selezionando i migliori individui ottenuti da quella generazione. L'obiettivo è generare discendenti con caratteristiche migliorate, superando in fitness i loro antenati. Un elemento cruciale di questo processo è la gestione accurata degli array di origine. Questo permette di mantenere una chiara tracciabilità delle origini di ogni individuo attraverso le generazioni, facilitando un'analisi puntuale dell'evoluzione delle sequenze nel tempo e permettendo di poter ottenere una statistica relativa alle sottoporzioni genetiche utilizzate negli array creati, discussa nel capitolo "Sperimentazione". Lo scopo degli array di origini è, quindi, sapere ogni individuo della popolazione risultato da quante e quali sottoporzioni di altre sequenze è formato.

Il processo di **crossover** rappresenta un pilastro fondamentale dell'algoritmo genetico implementato, emulando la ricombinazione genetica osservata in natura per promuovere la diversità genetica all'interno di una popolazione. Questa tecnica seleziona in modo casuale coppie di individui e identifica punti specifici all'interno delle loro sequenze per lo scambio di segmenti genetici. Il risultato di questo scambio è la creazione di nuovi individui, le cui sequenze sono una combinazione di segmenti prelevati dai loro "genitori", portando

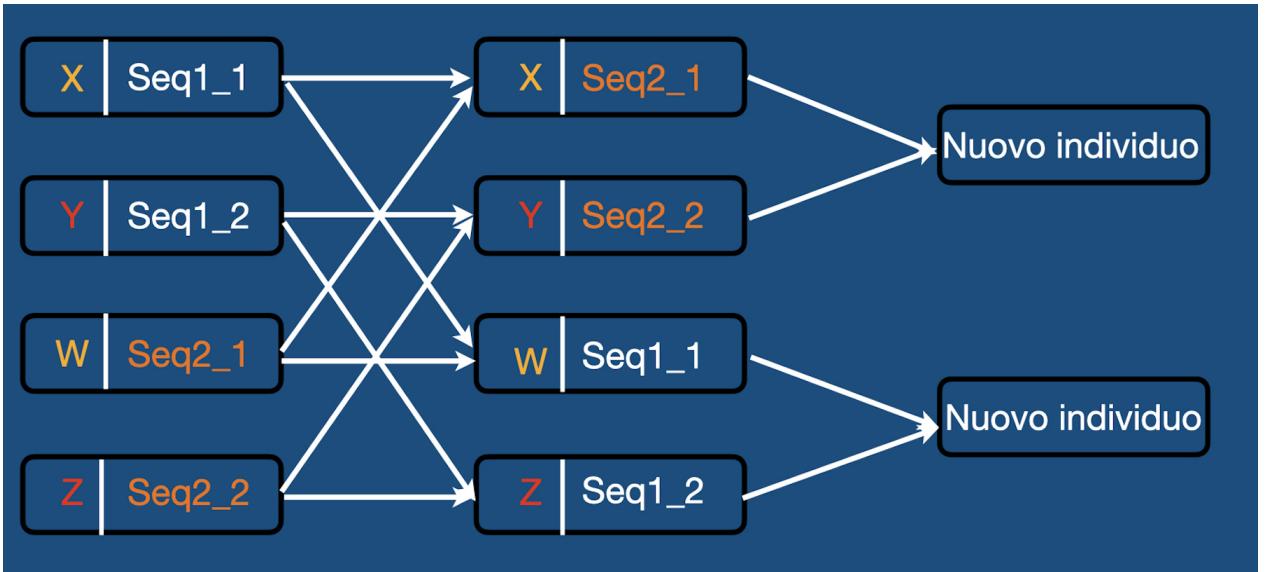


Figura 6: Rappresentazione grafica del crossover.

a una popolazione con una maggiore varietà genetica. Questo metodo si basa sulla premessa che combinando caratteristiche genetiche di diversi individui si possono ottenere discendenti con una fitness superiore, in termini di adattabilità e sopravvivenza, rispetto a quelli delle generazioni precedenti. Il successo di questa strategia risiede nella sua capacità di esplorare lo spazio delle soluzioni in modo più efficace, evitando il rischio di convergenza prematura verso ottimi locali che potrebbero non rappresentare la soluzione ottimale a livello globale.

Gli array di origine, che tracciano la provenienza di ogni segmento di sequenza all'interno degli individui, vengono aggiornati per riflettere questi cambiamenti. Questo non solo garantisce una comprensione dettagliata della genealogia e dell'eredità genetica di ogni individuo ma fornisce anche un meccanismo per analizzare l'evoluzione della popolazione nel tempo. Attraverso questa metodologia, l'algoritmo genetico proposto sfrutta i principi dell'evoluzione biologica per guidare la ricerca di soluzioni ottimali in un modo che è sia dinamico che adattivo.

Il meccanismo di **mutazione** costituisce un'altra componente fondamentale dell'algoritmo genetico, essenziale per introdurre variazioni genetiche all'interno della popolazione al fine di esplorare nuove aree dello spazio delle soluzioni e aumentare la diversità genetica. Selezionando individui in modo casuale e applicando modifiche specifiche alle loro sequenze genetiche si emulano le mutazioni spontanee che avvengono in natura,

Figura 7: Esempio di origin array, i numeri diversi sono gli id relativi alle sequenze che hanno subito mutazione e crossover.

creando l'opportunità per scoprire combinazioni genetiche che potrebbero presentare una fitness migliorata. La decisione su quale delle due sequenze disponibili per ciascun individuo andrà mutata, così come la scelta del segmento da mutare, è determinata in maniera casuale. Questa strategia porta alla formazione di individui caratterizzati da combinazioni genetiche precedentemente non esistenti nella popolazione. L'introduzione di nuove combinazioni genetiche all'interno della popolazione possono essere sfruttate dall'algoritmo genetico per esplorare spazi delle soluzioni altrimenti inaccessibili, riducendo il rischio di fermarsi in ottimi locali subottimali. Gli array di origine vengono aggiornati per mantenere la tracciabilità dell'origine di ogni segmento genetico. In questo modo, l'algoritmo genetico si arricchisce di un ulteriore livello di flessibilità e capacità di adattamento, sfruttando i meccanismi dell'evoluzione biologica per affinare la ricerca di soluzioni ottimali in maniera efficiente.

La differenza sostanziale tra i due operatori è il modo in cui vengono effettuati i cambiamenti delle sequenze e il numero di individui che vanno a generare. Nel nostro caso specifico, come si può vedere dalle rappresentazioni grafiche dei due operatori, il crossover effettua un taglio in posizione casuale dell'individuo andando a tagliare entrambe le sequenze che formano l'individuo, quindi taglia nello stesso punto i due individui selezionati casualmente e poi scambia le porzioni cambiate. La mutazione invece, provvede ad

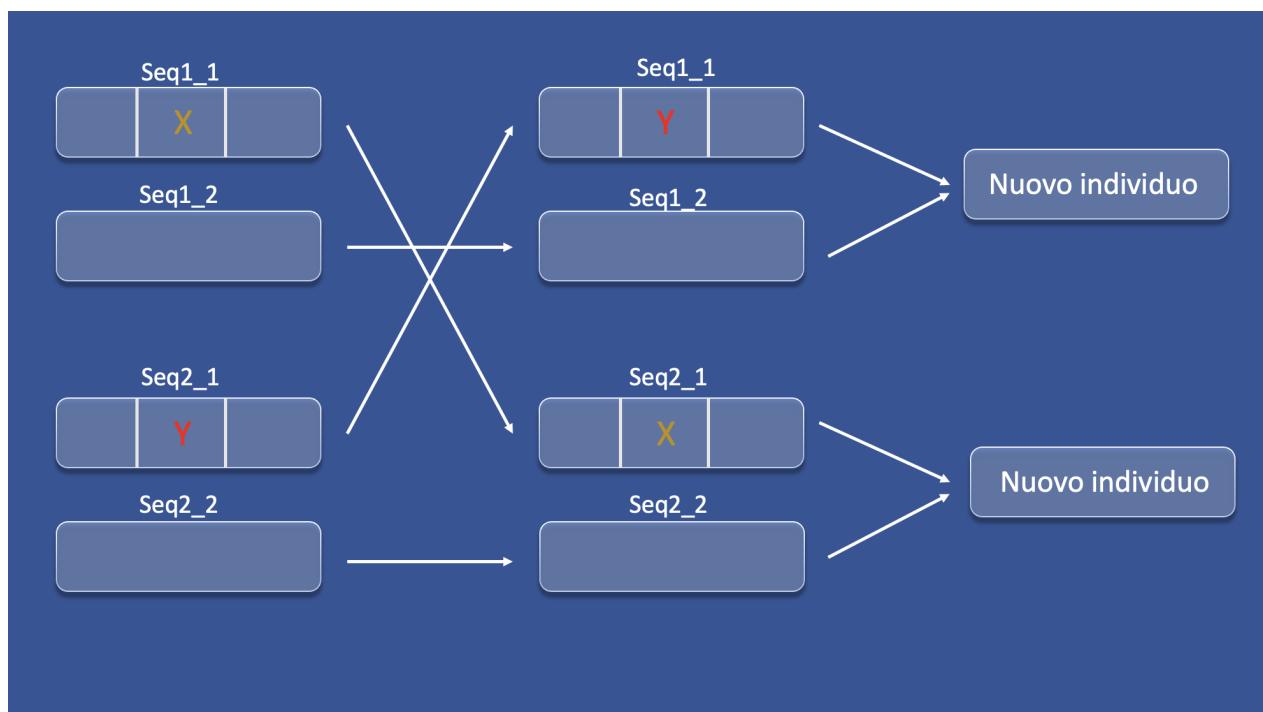


Figura 8: Rappresentazione grafica della mutazione.

effettuare due tagli nell’individuo, il primo riguarda l’inizio della sotto-sequenza che viene scambiata, il secondo taglio riguarda la dimensione inteso come numero di caratteri che vengono scambiati. Diversamente dal crossover, la mutazione effettua i tagli solo su una sequenza che compone l’individuo e procede allo scambio delle sotto-sequenze con la sequenze dell’altro individuo di pari posizione (ad esempio: scambia una porzione della prima sequenza del primo individuo con la prima sequenza del secondo individuo). La scelta di quale sequenza scegliere per la mutazione è presa dall’algoritmo in modo casuale. Un’altra differenza sostanziale tra gli operate è il numero di individui che vanno a generare, infatti per l’operazione di crossover andiamo a generare un numero di nuovi individui pari al doppio della popolazione, mentre invece per la mutazione generiamo nuovi individui pari alla popolazione. Le scelte implementative sono state prese in seguito ad una attenta sperimentazione che abbiamo condotto, in particolare abbiamo notato che un’eccessivo numero di individui creati durante la fase di mutazione può distruggere buone soluzioni esistenti e rendere il processo di evoluzione meno diretto. Limitare la mutazione a una sola volta per run aiuta a bilanciare l’esplorazione con la preservazione di individui già adatti. Allo stesso tempo abbiamo notato che il crossover in questo modo aumenta le possibilità di scoprire combinazioni di geni (parti delle sequenze) che portano a una fitness elevata.

```
Allineamento:  
CCTACGGG-  
GGCAGCAGTGGGAATCTCCGAATGGACGAAAGCTGACGGAGCAACGCCGTGAGTGTGACGCCCTCGGGTTGAAAGCTGTAAATCGGG-  
CGAAAT-GG-TTCTTGTC-AATAGT--  
GGCAGGATTGACGGTACCGGAATAGAAAGCCACGGCTAATCAGTGCAGCAGCCGCGTAATACGTAGGTGCAAGCGTTGTCGGATTATTGGCGTA  
AGCGCCGCAGGGATTG-GTCAGTCTGTTAAAAGTTGGGGCTTAACCCGTATGGG-ATGAAACTGCCA--  
ATCTAGAGTATCGGAGAGGAAAGTGAATTCTAGTGTAGCGGTAAATCGTAGATATTAGGAAGAACACCGTGGCGAAGGC-  
GACTTCTGGACGAAAATGACGCTGAGGCCAGGGAGCGAACCGGATTAGATACCC-TAGTGT  
|||||||  
|||||||  
|||||||  
|||||||  
|||||||  
|||||||  
|||||||  
|||||||  
CCTACGGG-  
GGCAGCAGTGGGAATCTCCGAATGGACGAAAGCTGACGGAGCAACGCCGTGAGTGTGACGCCCTCGGGTTGAAAGCTGTAAATC-  
GGAACGAA-AGGCCTTC-T-TGC-GAATAGTTAG-GAGGA-  
TTGACGGTACCGGAATAGAACGCAACGGCTAATCAGTGCAGCAGCCGCGTAATACGTAGGTGCAAGCGTTGTCGGATTATGGCGTAAGCGCGC  
CAGCGGA-T-CAGTCAGTCTGTTAAAAGTTGGGGCTTAACCCGTGA-GGGGATGAAACTG-C-  
TGATCTAGAGTATCGGAGAGGAAAGTGAATTCTAGTGTAGCGGTAAATCGTAGATATTAGGAAGAACACCGTGGCGAAGG-  
TGACTTCTGGACGAAAATGACGCTGAGGCCAGGGAGCGAACCGGATTAGATACCCGT-GTAGT
```

Figura 9: Corrispondenza di una sequenza nell'approccio 'Brute Force' con l'uso di 'align'.

3 Sperimentazione

3.1 Stato dell'arte

3.1.1 Allineamento Pairwise con Bio.pairwise2

Bio.pairwise2 è un modulo fornito dalla libreria Biopython, che è una collezione di strumenti e risorse per la bioinformatica. Biopython offre un insieme di moduli e pacchetti che semplificano lo sviluppo di script e programmi per l'analisi di dati biologici. In particolare, Bio.pairwise2 è progettato per affrontare problemi di allineamento di sequenze. Può essere utilizzato per identificare regioni simili o omologhe tra le sequenze. La funzione chiave fornita da Bio.pairwise2 è denominata 'align', la quale permette di eseguire l'allineamento di sequenze. Nel caso specifico utilizzato, l'opzione 'globalxx' permette di effettuare un allineamento globale, utilizzando un punteggio di corrispondenza uguale a 1 per ogni corrispondenza e 0 per ogni incongruenza o gap. Questo tipo di allineamento è particolarmente utile quando si desidera valutare l'intera lunghezza delle sequenze, rilevando accuratamente le corrispondenze e le differenze tra di esse. La figura 11 rappresenta un passo di esecuzione del vecchio script in cui viene evidenziato l'allineamento fatto dalla libreria Bio. Il metodo format_alignment mostra l'allineamento effettuato tra le due sequenze con il relativo punteggio.

Per la sequenza 1:
 Miglior punteggio di allineamento: 0.9340767860412598
 Sequenza corrispondente:
 CCTACGGGGGCCAGCAGTG666AATCTTCCGCAATGGCGAACAGCCTGACGGAGCACAGCCGCGTGAAGAAGGTCTCGGATCGTAAAGCTCTGTTGGGAGCAAAGGGAGGGAGGAATGCCCTTTAAGACGCTACCGGAC
 GAGCGAACGCCAGCGTAACCTAGTGGCCAGCAGCCCGGTAAATCCTAGTGGCGAGCGTTGCGGAATGATTGGCGTAAAGGGAGCGCAGGTGGCACGTTAAAGCTCTTAAAGCGTGGGCTCAGCCCCATGAAGGGAGGAACATTC
 GATCTTAGTGGCGGAGAGGAAAGCGGAATTCCAGTGTAGCGTGAATTCGGAGAACACCAAGTGGCGAAGGCCTTCTGGACGCCACTGACACTTGAGGCTCGAAAGCCAGGGAGCGAACAGGGATTAGATAACCTGG
 TAGTC
 Numero di sequenza corrispondente: 235 su 381 (percentuale: 61.68%)
 Allineamento:
 CCTACGGGGGCCAGCAGTG666AATCTTCCGCAATGGCGAAC-CCCTGACGGAGCACGCCGCGTGAAGAAGGTCTCGGATCGTAAAGCTCTGTTGAT-GGGGACGAACTGCGCAATGCG--A-ATA-G--TTTAAAGGC
 -AATGACGGTACCC-ATCGAGG-AAGGCCACGCCAACCTAGTGGCCAGCAGCCCGGTAAATCCTAGTGGCGAGCGTTGCGGAATC-ATTGGCGTAAAGGGAGGCCAGGC-GGGCAT-G-TAAGTCT-TTCTAAAGTGGC-GGG
 -CTCAA-CCCCG-TGAT-GGGAAAG-AAACTATGT-G-TCTTGGAGTA-CA-GGAGGAGAACCGGAATTCCAGTGTAGCGTGAATTCGGAGAACACCAAGTGGCGAAGGCCTTCTGGACTG-CAACTGACG
 -CTGAGGCTCGAAAGCAGGGAGCGAACGGATTAGATAACCC-GAGTAGTC
 ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| ||||| |||||
 CCTACGGGGGCCAGCAGTG666AATCTTCCGCAATGGCG-AAGCCTGACGGAGCACGCCGCGTGAAGAAGGTCTCGGATCGTAAAGCTCTGTTGAT-GGGGAGCAAAG-G-GGGAGGA-AATGCCCTTT---TAA
 -GACGGTACCCGA-CGA-GCAAGGCCACGCCAACCTAGTGGCCAGCAGCCCGGTAAATCCTAGTGGCGAGCGTTGCGGAAT-GATTGGCGTAAAGGGAGGCCAGG-TGGG-A-CGGTAAGTC-CTTCTAAAAA-GCGTGGGCTC
 -AGCCC-ATGA-AGGG-AAGGAAACTA-TCGATCTTGGAT-GC-CGGAGGAGAACCGGAATTCCAGTGTAGCGTGAATTCGGAGAACACCAAGTGGCGAAGGCCTTCTGGAC-GCCAACCTGAC
 -ACTGGGGCTCGAAAGCAGGGAGCGAACGGATTAGATAACCTGGT-GTAGTC
 Score=430

Figura 10: Esempio di stampa del metodo 'global_alignment' con score.

3.2 Sviluppo metodo personalizzato

Di seguito è riportata un'implementazione della funzione **global_alignment** personalizzata che calcola l'allineamento globale tra due sequenze di DNA. Un aspetto chiave è l'uso di una matrice di allineamento per memorizzare i punteggi intermedi durante il processo di allineamento. La matrice è inizializzata con dimensioni adeguate alle lunghezze delle sequenze in ingresso e viene riempita iterativamente calcolando i punteggi in base alle corrispondenze, non corrispondenze e penalità per i gap. Il processo di ricostruzione dell'allineamento a partire dalla matrice è gestito attraverso un loop che risale dalla fine della matrice. Durante questo processo, vengono identificati i passaggi che hanno portato al punteggio corrente, permettendo la ricostruzione delle sequenze allineate. La flessibilità dell'algoritmo è evidenziata dalla possibilità di personalizzare i punteggi per corrispondenze (match_score), non corrispondenze (mismatch_score), e la penalità per i gap (gap_penalty). Questi parametri consentono all'utente di adattare l'allineamento in base alle specifiche esigenze dell'analisi. La scelta del massimo tra i punteggi di match, delete e insert in ogni passaggio assicura che l'allineamento ottenuto massimizzi il punteggio complessivo.

In aggiunta, la funzione **format_alignment** è progettata per presentare in modo chiaro e leggibile l'allineamento tra le due sequenze. Vengono creati segmenti di sequenze di lunghezza massima specificata (`line_length`), e le corrispondenze tra le basi sono evidenziate con il simbolo "—" mentre le differenze sono rappresentate da spazi. Questa formattazione agevola l'analisi visiva dell'allineamento, facilitando la

```

Allineamento:
CCTACGGGAGGCAGCGAGTAAG-AATATTCCG-CAATGGGGGAACCTGA-CGGAGCGACGCCCGTGAAC-GAAGAAG
||||||| ||| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
CCTACGGGGGGCAGCAG-TGGGAATTG-GACAATGGCGCAAGCCTGATCC-AGCCATGCCCGTGT-CTGAAGAAG

GCCGGACGG-TTGAAAGTTCTTCTGTCGAGGAATAA--GTG-TAGGAGGAATGCC--TG-CATGGTACGGTAGG
||| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
GCCTT-CGGTTGTAAGGACTTT--GTCAG-GGAAGAAAAG-GATAGG-GTTAATACCCCTGTC-TGATGACGGTACC

-GCAGGAATAAGCACCGGCTAATTACGTGCCAGCAGCGCGTAACACGTAAGGTGCGAGCGTTTCGAATTATTGGG
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
TGAAG-AATAAGCACCGGCTAACTACGTGCCAGCAGCGCGTAATACGTAGGGTGCAGCGTTAACCGAATTACTGGG

CGTAAAG-G-GCATGCAGGGCGGT-CGCCAAGCTTG-TAAGAAATACCGGGCTCAACTCCGG-AGCTATATTGAGAAC
||||||| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
CGTAAAGCGAGC--GCAGACGGTTACTT-AAGCA-GGATGTGAATCCCGGGCTCAAC-CTGGAACTGCGTTCTGAAC

TGGC-GAGCTAGAGT-TGCC-GAAGG-G-TATCCGAATTCCGCGTGAAGGGGTGAAAT-CTGTAGATATGCCAAGAAC
||| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
TGGGTGA-CTAGAGTGTGTCAGAGGGAGGTA---G-AATTCCACGTGAGCAGTGAAATGC-GTAGAGATGTGGAGGAAT

ACCGATGGCGAAGGCAGGATACCGCGGA---CGACTGACGCTGAGG-TGCAGA-G-GTGCAGGAGCAAACAGGATTAG
||||||| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
ACCGATGGCGAAGGCAGCCT-CCTG-GGATAAC-ACTGACGTTCATGCT-CGAAAGCGT-GGT-AGCAAACAGGATTAG

ATACCCCGGTAGTC
||||||| | | | |
ATACCCAGTAGTC

```

Figura 11: Corrispondenza di una sequenza nell’approccio ‘Brute Force’ con l’uso dei metodi personalizzati.

comprendere delle corrispondenze e delle variazioni tra le sequenze.

Nonostante i metodi personalizzati ci siamo resi conto che lo ’score’ utilizzato dai metodi non era rappresentativo al massimo dato che era un calcolo che non veniva fatto da Sense, piuttosto dalla libreria pairwise2. Per questo motivo non rappresenta una soluzione discriminante per la fase di selezione che interessa a noi

3.3 Metodo di confronto con embeddings

Nel processo di confronto tramite gli embeddings generati dalla rete neurale Siamese, ogni sequenza genomica viene rappresentata da un vettore numerico compatto che cattura le sue caratteristiche salienti. Questi embeddings fungono da rappresentazioni semantiche delle sequenze, acquisendo informazioni dettagliate sulla struttura e sulla composizione delle basi genetiche. Il cuore di questo approccio risiede nella capacità della rete Siamese di apprendere automaticamente queste rappresentazioni. Per ogni sequenza nel dataset, la funzione `net.get_embedding` estrae l’embedding corrispondente, fornendo un vettore numerico che codifica

le informazioni genetiche specifiche della sequenza stessa. Una volta ottenuti gli embeddings per tutte le sequenze coinvolte, il confronto tra di esse diventa una questione di valutare la distanza e la similarità tra i rispettivi vettori di embedding. Questo processo è svolto attraverso il calcolo della distanza di Jaccard che riflette la somiglianza semantica tra le sequenze.

3.4 Sottosequenze genetiche

Nell'ambito dell'algoritmo genetico implementato, come sottolineato negli obiettivi, lo scopo è quello di andare a formare sequenze genomiche che vengono rappresentate da sottosequenze provenienti dai risultati del crossover implementato. L'analisi delle sottosequenze si basa sull'osservazione degli indici di provenienza delle sottosequenze utilizzate durante il crossover. Nello specifico, abbiamo implementato una struttura dati ad hoc, chiamata `origin_array`, che ci permettesse di tener traccia delle modifiche e delle loro provenienze anche in caso di numerose generazioni, come nel nostro caso. Attraverso l'identificazione degli indici con la massima frequenza di occorrenza all'interno di tali strutture, è possibile individuare specifiche regioni genomiche che sono state soggette a mutazione più frequenti. Ad esempio, considerando il processo di crossover delle sequenze, le sottosequenze emergenti da tali operazioni sono rappresentate dagli intervalli di inizio e fine casuali all'interno delle sequenze originali.

L'analisi dei risultati mostra come tali sottosequenze siano coinvolte in processi di ricombinazione genetica (crossover). La frequenza di occorrenza di queste sottosequenze fornisce indicazioni sulle dinamiche evolutive delle sequenze genomiche nel corso delle generazioni. Questa precisa tecnica permette di tracciare le traiettorie evolutive delle sequenze selezionate, contribuendo a una comprensione più approfondita delle influenze della struttura genomica nell'ambito dell'algoritmo genetico utilizzato.

```

array originale 1_1
['0', '9', '9', '3', '3', '7', '1', '7', '7', '5', '3', '9', '8', '5', '1', '3', '9', '3', '9', '0',
'7', '9', '3', '0', '9', '7', '5', '8', '3', '9', '0', '8', '0', '0', '5', '0', '1', '3', '1', '9',
'7', '0', '1', '7', '1', '0', '0', '3', '0', '0', '8', '7', '9', '5', '5', '7', '3', '3', '0', '3',
'3', '8', '7', '5', '8', '9', '9', '5', '1', '9', '1', '3', '8', '3', '9', '9', '9', '5', '9', '8',
'3', '7', '7', '0', '1', '8', '7', '5', '9', '9', '7', '7', '5', '9', '9', '5', '0', '9', '5', '0',
'5', '7', '1', '9', '1', '3', '7', '3', '5', '7', '9', '5', '3', '9', '5', '7', '5', '7', '9', '9',
'0', '7', '9', '3', '7', '7', '3', '3', '1', '1', '0', '0', '1', '8', '7', '3', '9', '9', '1',
'8', '0', '0', '9', '8', '7', '9', '9', '0', '1']

nuovo array 1_1
['0', '9', '9', '3', '3', '7', '1', '7', '7', '5', '3', '9', '1', '5', '9', '7', '7', '7', '0', '5',
'9', '5', '0', '8', '5', '7', '8', '1', '9', '0', '7', '0', '8', '0', '0', '7', '1', '7', '0', '5',
'3', '0', '0', '1', '8', '0', '0', '0', '3', '3', '8', '3', '9', '8', '7', '7', '7', '9', '0', '8',
'9', '9', '3', '7', '3', '9', '9', '3', '3', '3', '9', '9', '3', '7', '5', '8', '8', '9', '9', '9',
'1', '9', '5', '0', '8', '9', '5', '1', '3', '3', '9', '9', '5', '7', '1', '0', '1', '8', '0', '8',
'0', '5', '8', '1', '7', '0', '5', '8', '9', '7', '0', '1', '1', '1', '3', '0', '3', '9', '0', '5',
'1', '9', '7', '1', '1', '0', '9', '1', '8', '7', '5', '3', '0', '5', '7', '5', '5', '7', '9',
'3', '5', '5', '3', '1', '0', '0', '1', '7', '7']

```

Figura 12: Esempio di mutazione avvenuta.

```

● ● ●

Array numero 1:
Max Occurrences (Array 1):
{'0': 50, '96': 120, '26': 1, '16': 30, '66': 49, '94': 1, '40': 23, '50': 19, '60': 17, '82': 43,
'34': 6, '4': 24, '52': 17, '74': 33, '36': 1, '6': 31}
Indice con il valore massimo in Array 1: 96 (120/465 corrisponde al 25.806%)

```

Figura 13: Esempio di sottoporzione evidenziata data una sequenza.

4 Risultati

4.1 Risultati del modello

Nell’ambito del confronto tra approcci all’allineamento di sequenze nucleotidiche, i risultati ottenuti dal modello di rete neurale siamese sono stati valutati utilizzando come riferimento un algoritmo oracolare basato su tecniche convenzionali. Il codice dell’oracolo, un semplice script Python, calcola la distanza tra due sequenze come il rapporto tra il numero di non corrispondenze (mismatch) e il numero totale di caratteri confrontati. Questo fornisce una misura diretta dell’allineamento delle sequenze nucleotidiche: più basso è il valore restituito, maggiore è la similitudine tra le sequenze. Il modello di apprendimento automatico, la rete neurale siamese, è stato addestrato e ottimizzato per minimizzare la distanza tra le sequenze che allinea, cercando di avvicinarsi il più possibile al valore fornito dall’oracolo. La rete siamese è stata soggetta a diverse configurazioni di parametri, come lunghezza delle sequenze, dimensioni degli embedding, numero di epoch, learning rate e dimensione dei batch, per identificare la combinazione più efficace per replicare la metrica dell’oracolo.

Attraverso questo approccio sistematico, si è scoperto che la rete neurale siamese è in grado di emulare con notevole precisione il comportamento dell’oracolo tradizionale. Questo è evidenziato dalla distribuzione dei risultati, i quali mostrano una forte aderenza alla metrica dell’oracolo, segnalando che il modello non solo ha appreso con successo le regole di allineamento del dataset ma è anche in grado di generalizzare questa conoscenza a nuovi dati. Questo indica che le reti neurali siamese potrebbero rappresentare una metodologia alternativa e potenzialmente più efficiente per l’allineamento di sequenze nucleotidiche, con implicazioni significative per la bioinformatica e la biologia computazionale.

4.2 Confronto tra Metodi

Di seguito sono riportati i risultati delle sperimentazioni. Questi indicano le gli accoppiamenti ottimali individuati dall’algoritmo brute force. Per ogni sequenza utilizzata viene identificata la sequenza corrispondente che l’algoritmo assegna con il punteggio di allineamento che rappresenta l’accoppiamento ottimale per la sequenza di partenza. Viene inoltre mostrata la sequenza assegnata dall’algoritmo. Gli allineamenti identificati dall’algoritmo forza bruta vengono ricercati all’interno della popolazione generata dall’algoritmo genetico. Come anticipato, i risultati forza bruta vengono utilizzati come ”oracolo” per valutare l’efficacia dell’algoritmo genetico nell’individuare gli accoppiamenti ottimali nonostante l’approccio eristico. L’immagine mostra le sequenze individuate alla fine delle generazione, in particolare per ogni accoppiamento individuato dal-

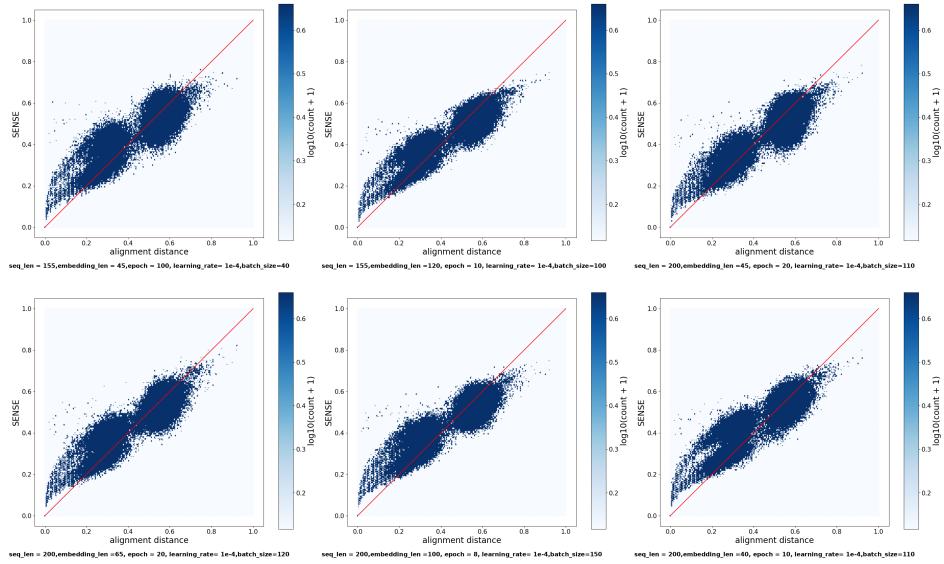


Figura 14: : Alcuni dei risultati del modello con diverse combinazioni di iperparametri.

l’approccio brute force, riporta la percentuale di corrispondenze e la percentuale di presenti all’interno degli individui della popolazione risultante. Il numero di corrispondenze corrispondono alle dimensioni delle sottosequenze individuate mentre il numero di array rappresentano le sequenze create dopo i processi di mutazione e crossover e selezione. In questa specifica sperimentazione possiamo notare come il numero di accoppiamenti individuati corrisponde all’70% delle coppie trovate e come il numero di sequenze che posseggono un determinato accoppiamento sia piuttosto alto. Questo significa che una volta che l’algoritmo genetico trova un accoppiamento ottimale, corrispondente del brute force, esso riesce a vincere sempre nel processo di selezione in base alla fitness e riesce a propagarsi tra i vari individui mediante il processo di selezione. Di seguito abbiamo reso disponibile il file di log che abbiamo utilizzato durante questa sperimentazione per rendere a tutti disponibile la consultazione dei risultati ottenuti e una attenta osservazione di come funzionano gli operatori utilizzando, prevendo numero stampe durante il processo atte a rendere bene l’idea del funzionamento. Per accedere ai file di log clicca qui. In conclusione, la sperimentazione proposta è stata effettuata su un campione piccolo di sequenze (10) e su un numero elevato di generazioni (1000). I risultati ottenuti sono soddisfacenti, dato che raggiungiamo l’70% degli accoppiamenti individuati, nonostante il campione utilizzato. Nella nostra sperimentazione non abbiamo potuto utilizzare la potenza di calcolo di GPU, per tanto immaginiamo che all’aumentare della popolazione impiegata nell’algoritmo genetico e di conseguenza della

complessità dell'algoritmo, le prestazioni rimangono invariate, lasciamo un numero di generazioni piuttosto alto.

```
Per l'allineamento 1-3 sono state trovate 18 corrispondenze in 14 array diversi.  
Percentuale di corrispondenza: 11.61%. Percentuale di array: 70.00%.  
Per l'allineamento 2-4 sono state trovate 69 corrispondenze in 18 array diversi.  
Percentuale di corrispondenza: 44.52%. Percentuale di array: 90.00%.  
Per l'allineamento 3-1 sono state trovate 0 corrispondenze in 0 array diversi.  
Percentuale di corrispondenza: 0.00%. Percentuale di array: 0.00%.  
Per l'allineamento 4-2 sono state trovate 21 corrispondenze in 16 array diversi.  
Percentuale di corrispondenza: 13.55%. Percentuale di array: 80.00%.  
Per l'allineamento 5-4 sono state trovate 0 corrispondenze in 0 array diversi.  
Percentuale di corrispondenza: 0.00%. Percentuale di array: 0.00%.  
Per l'allineamento 6-4 sono state trovate 70 corrispondenze in 20 array diversi.  
Percentuale di corrispondenza: 45.16%. Percentuale di array: 100.00%.  
Per l'allineamento 7-8 sono state trovate 0 corrispondenze in 0 array diversi.  
Percentuale di corrispondenza: 0.00%. Percentuale di array: 0.00%.  
Per l'allineamento 8-2 sono state trovate 28 corrispondenze in 14 array diversi.  
Percentuale di corrispondenza: 18.06%. Percentuale di array: 70.00%.  
Per l'allineamento 9-2 sono state trovate 15 corrispondenze in 12 array diversi.  
Percentuale di corrispondenza: 9.68%. Percentuale di array: 60.00%.  
Per l'allineamento 10-8 sono state trovate 41 corrispondenze in 17 array diversi.  
Percentuale di corrispondenza: 26.45%. Percentuale di array: 85.00%.
```

Figura 15: : Risultati del confronto, riportati nel file di log.