



Studio di Genoogle e dei tool utilizzati nel 2024

La bioinformatica ha il compito fondamentale di cercare sequenze genetiche simili nelle banche dati, le quali stanno aumentando esponenzialmente grazie alle nuove tecnologie di sequenziamento, causando un allungamento dei tempi di ricerca. Questo lavoro propone di combinare tecniche di indicizzazione dei dati per ridurre il costo computazionale con metodi di ricerca parallelizzati, in modo da ottimizzare l'uso delle risorse multicore.

Leonardo Monaco

Algoritmi di Allineamento

Needleman Wunsch e Smith-Waterman

Gli algoritmi di allineamento Needleman-Wunsch e Smith-Waterman, basati sulla programmazione dinamica, presentano costi computazionali e di memoria quadratici ($O(mn)$), rendendoli poco pratici per grandi banche dati.

FASTA e BLAST

FASTA e BLAST i due tool che utilizzano algoritmi di allineamento hanno ottimizzato il costo dell'allineamento limitandolo solo a parole simili precedentemente identificate. Tuttavia, la complessità rimane $O(nmq)$ perché è necessario esaminare tutte le sequenze per individuare il P.A.S. (Pattern di Allineamento Sottostante).

Indici Invertiti

L'ottimizzazione del processo di ricerca può avvenire attraverso l'uso di indici invertiti, che permettono di localizzare rapidamente i dati indicizzati in tempo costante ($O(1)$).



Strutture Dati per l'Indicizzazione

1

Alberi dei Suffissi

Gli alberi dei suffissi, utilizzati da Gusfield, permettono di accedere rapidamente alle posizioni delle sottosequenze e di identificare sequenze ripetute o l'antecittale comune più lungo, ma richiedono un elevato consumo di memoria.

2

Vettori

Il vettore è una struttura dati che funge da array di elementi, con ogni posizione che contiene un altro array per localizzare le informazioni.

3

Tabelle Hash

Alcune tecniche di ricerca, come quella di Kalafus, impiegano tabelle hash per l'allineamento di genomi.



Genoogleg: Un Motore di Ricerca Innovativo

1

Introduzione

Genoogleg è stato sviluppato per affrontare la sfida della crescita esponenziale delle banche dati genetiche, combinando tecniche di indicizzazione e parallelizzazione per ottimizzare la ricerca di sequenze simili.

2

Metodologie

Genoogleg utilizza un indice invertito e tecniche di parallelizzazione per sfruttare i processori multi-core, migliorando significativamente i tempi di ricerca rispetto agli strumenti tradizionali come BLAST.

3

Implementazione e Risultati

Il software è stato sviluppato in Java e offre interfacce web, servizi web e modalità testo. I test hanno dimostrato che Genoogleg è 20 volte più veloce di BLAST in ricerca sequenziale e 26,60 volte più veloce in ricerca parallela.



Il Sistema Genoole

Indice Invertito

Genoole utilizza un indice invertito per gestire le sottosequenze di DNA, con una struttura dati principale costituita da un vettore la cui dimensione è determinata dal numero possibile di sottosequenze.

Maschere

Genoole utilizza delle maschere, ispirate a PatternHunter, per migliorare la sensibilità di ricerca e ridurre la dimensione dell'indice.

Parallelizzazione

Genoole utilizza tre tecniche di parallelizzazione per migliorare i tempi di ricerca e sfruttare le capacità di multi-elaborazione.

Interfacce

Genoole offre un'interfaccia testuale, una semplice interfaccia web e un'interfaccia per servizi web, permettendo agli utenti di eseguire query e automatizzare l'accesso ai servizi.



Il Processo di Ricerca

1

Elaborazione della Sequenza

Nella fase di elaborazione, si applica una maschera a ciascuna sottosequenza della sequenza di input, codificandola in binario per facilitarne l'accesso e l'individuazione nell'indice invertito.

2

Ricerca nell'Indice

Il processo di ricerca recupera le informazioni dall'indice invertito a partire dalla sequenza di input codificata, memorizzando le posizioni delle sottosequenze simili in array.

3

Estensione e Allineamento

Dopo aver identificato i potenziali allineamenti di sottosequenze (P.A.S.), si procede a un'operazione di estensione e allineamento locale utilizzando una versione modificata dell'algoritmo di Smith-Waterman.

Parallelizzazione in Genoogle



Parallelizzazione dell'Indice

Genoogle parallelizza l'accesso all'indice invertito, suddividendo la banca dati in frammenti e effettuando ricerche indipendenti tramite thread per ciascun frammento.



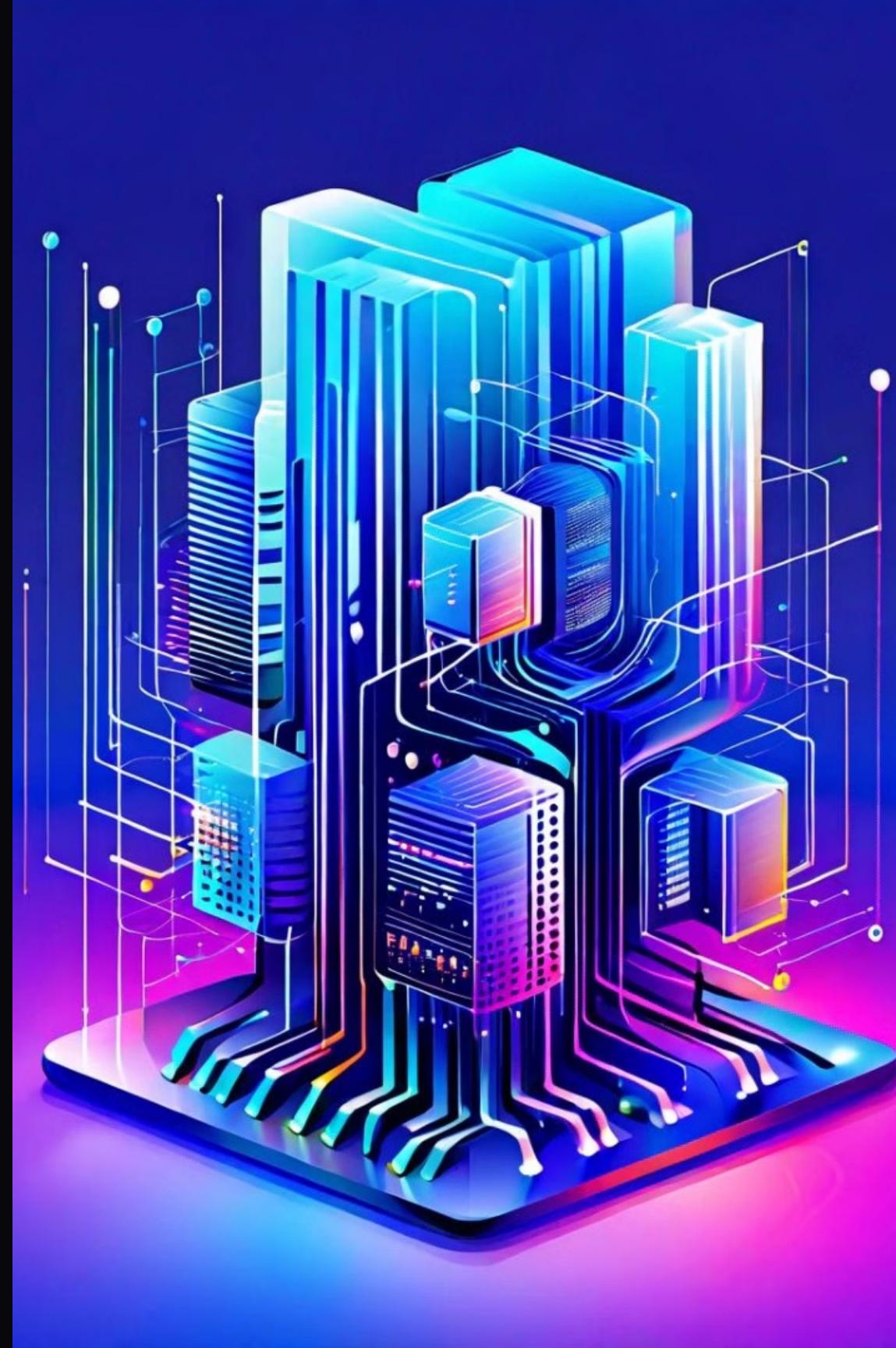
Parallelizzazione di Estensione e Allineamento

Genoogle parallelizza anche le fasi di estensione e allineamento, utilizzando tutti i core del computer per elaborare i P.A.S. (Potenziali Allineamenti di Sequenze) in modo efficiente.



Parallelizzazione della Divisione della Banca Dati

Genoogle divide sia la banca dati che le sequenze di input, consentendo di effettuare ricerche parallele senza congestionare la memoria con strutture dati eccessive.



Risultati Sperimentali



Qualità dei Risultati

La qualità dei risultati di Genoole è stata valutata confrontando l'output con quello di BLAST, mostrando un'efficacia elevata nel trovare allineamenti con valori elettronici inferiori a 10^{-25} .



Miglioramento delle Prestazioni

Gli esperimenti hanno mostrato che Genoole è significativamente più veloce di BLAST, con un'accelerazione di oltre 26,60 volte in modalità parallela e quasi 20 volte in modalità sequenziale.



Conclusioni

1

Indicizzazione e Parallelizzazione

Genoogleg combina tecniche di indicizzazione e parallelizzazione per migliorare l'efficacia nella ricerca di somiglianze genetiche, affrontando in modo efficace le sfide legate all'ottimizzazione della ricerca.

2

Strumento Innovativo

Il lavoro presentato è significativo poiché introduce un metodo innovativo per la ricerca di sequenze genetiche, utilizzando l'indicizzazione ed il calcolo parallelo.

3

Ottimizzazione della Ricerca

Con l'aumento dei core nei processori e la crescita esponenziale delle banche dati, l'approccio di Genoogleg affronta in modo efficace le sfide legate all'ottimizzazione della ricerca.

Analisi del Codice

In questa sezione verrà analizzato il codice del tool Genoogle



Analisi del Codice di Genoole

Algoritmi Chiave

Il codice di Genoole include algoritmi innovativi come il `DividedStringGenooleSmithWaterman` per l'allineamento locale di sequenze di DNA e l'encoder di sequenze DNA/RNA con maschera.

Suite di Test

Genoole è dotato di una suite completa di test unitari che verificano la correttezza di componenti critiche come l'indice invertito, l'encoder di sequenze e il lettore di file FASTA.

Parallelizzazione

Il codice sfrutta il parallelismo per accelerare le operazioni di ricerca nell'indice, l'estensione e l'allineamento delle sequenze, ottenendo significativi miglioramenti delle prestazioni.

Analisi Dell'Algoritmo

DividedStringGenoogleSmithWaterman

1

Descrizione Dell'Algoritmo

L'algoritmo di Smith-Waterman originale è noto per il suo utilizzo nell'allineamento locale di sequenze, dove l'obiettivo è trovare i segmenti di sequenze con la più alta somiglianza possibile. L'algoritmo **DividedStringGenoogleSmithWaterman** introduce una variante che suddivide le sequenze in segmenti per ottimizzare l'allineamento, migliorando l'efficienza e l'accuratezza nei confronti di sequenze di DNA.

2

Implementazione Del Test Unitario

Il codice di test per l'algoritmo è implementato in Java, all'interno di una classe di test unitario. Questa classe contiene diversi metodi di test progettati per verificare la correttezza e l'efficacia dell'algoritmo attraverso vari scenari di input.

3

Verifica Dell'Output

In ciascun metodo di test, l'algoritmo viene eseguito con sequenze di input e parametri specifici, e l'output viene verificato rispetto ai risultati attesi. La funzione `Assert.assertEquals()` è utilizzata per confrontare l'output effettivo con quello atteso, garantendo che l'algoritmo funzioni come previsto.



Analisi Dell'Encoder Di Sequenze DNA/RNA Con Maschera

Descrizione Dell'Encoder

L'encoder di sequenze DNA/RNA con maschera è uno strumento che prende in ingresso una sequenza di nucleotidi e una maschera binaria. La maschera specifica, tramite bit 1 e 0, quali parti della sequenza devono essere mantenute o scartate. Questo processo è utile in vari contesti di bioinformatica, come il filtraggio di sequenze o l'identificazione di regioni conservate.

Implementazione Del Test Unitario

La classe di test unitario è stata sviluppata per verificare la correttezza dell'encoder. Questa classe include diversi metodi di test, ognuno dei quali applica l'encoder a sequenze diverse, con specifiche maschere, e confronta i risultati ottenuti con quelli attesi.

Verifica Dell'Output

In ciascun metodo di test, l'encoder viene configurato con un alfabeto specifico (DNA o RNA) e una lunghezza di sottosequenza determinata. La maschera viene applicata alla sequenza di input e il risultato ottenuto viene verificato rispetto al risultato atteso tramite l'uso di `Assert.assertEquals()`.

Suite Di Test Per L'Encoder Di Sequenze Di DNA

1 Descrizione Della Classe Di Suite Di Test

La classe di suite di test include un metodo statico chiamato `suite()` che costruisce e restituisce una suite di test completa. Questo metodo è responsabile di aggregare i test pertinenti in una singola unità eseguibile, facilitando così la verifica del comportamento del sistema in modo completo e organizzato.

2 Classi Di Test Incluse

La suite di test include le seguenti tre classi di test: **SequenceEncoderTest**, **SequenceEncoderToIntegerTest** e **MaskEncoderTest**.

3 Implementazione Del Metodo Suite()

Il metodo `suite()` crea una nuova istanza di **TestSuite** e aggiunge ciascuna delle classi di test sopra menzionate alla suite utilizzando il metodo **addTestSuite()**. Questo processo assicura che tutti i test pertinenti siano inclusi e che possano essere eseguiti insieme in un'unica operazione.



Test Unitario Per L'Encoder Di Sequenze Di DNA

Descrizione Del Test

La classe di test contiene un unico metodo chiamato **testGetBitsBySize()**, che verifica il comportamento del metodo **bitsByAlphabetSize()** della classe SequenceEncoder. Questo metodo è essenziale per determinare il numero di bit necessari per rappresentare un alfabeto di una determinata dimensione.

Implementazione Del Test

Il test utilizza il metodo **assertEquals()** per confrontare l'output del metodo **bitsByAlphabetSize()** con i valori attesi. Questo metodo è un'asserzione standard in JUnit, utilizzata per verificare che due valori siano uguali. Se i valori non corrispondono, il test fallisce, indicando un possibile problema nel metodo testato.

Conclusione

Il test unitario descritto garantisce che il metodo **bitsByAlphabetSize()** funzioni correttamente per varie dimensioni di alfabeti. Verificare accuratamente il comportamento di questo metodo essenziale per il corretto funzionamento del sistema di encoding, poiché una rappresentazione inefficiente potrebbe compromettere la qualità e la precisione dei dati genetici codificati.



Test Unitario Per Il Compressore Di Sequenze Di DNA e RNA



Encoding e Decoding di Sottosequenze di DNA e RNA

Questi test verificano la capacità del compressore di codificare sottosequenze specifiche e di decodificarle correttamente, assicurando che la rappresentazione finale corrisponda alla sequenza originale.



Encoding e Decoding di Sequenze Complete di DNA e RNA

In questo caso, i test esaminano la capacità del sistema di gestire sequenze complete, verificando che l'intero processo di compressione e decompressione sia accurato e privo di errori.



Verifica per Diverse Lunghezze e Contenuti di Sequenza

Per garantire la robustezza del compressore, i test includono sequenze di diverse lunghezze e composizioni. Questo assicura che l'encoder e il decoder possano gestire correttamente qualsiasi variazione nel contenuto delle sequenze.





Suite Di Test Per L'Indice Di Genoole

- 1 InvertedIndexBuilderTest
- 2 SubSequencesArrayIndexTest 8
- 3 SubSequencesArrayIndexTest 11
- 4 SubSequencesArrayIndexTest 11Masked

Questo test verifica la costruzione dell'indice invertito, un componente cruciale per l'efficienza del motore di ricerca.

Testa la correttezza dell'indice delle sottosequenze di DNA di lunghezza 8.

Verifica la gestione delle sottosequenze di lunghezza 11.

Esamina la corretta applicazione delle maschere sulle sottosequenze di lunghezza 11.



Test Unitario Per L'Inverted Index Builder

Metodo **testBeginEnd()**

Verifica la costruzione dell'indice invertito dall'inizio alla fine.

Metodo **testBeginInsertOneSmallSequenceEnd()**

Si concentra sull'inserimento corretto di sequenze nell'indice invertito.



Studio dei tool utilizzati nel 2024

Nei capitoli precedenti è stato analizzato il tool Genoole, il nostro studio però vuole introdurre una panoramica generale sui tool che vengono utilizzati attualmente concentrandosi sulle caratteristiche di quest'ultimi.

CLUSTAL Omega

1 MSA

CLUSTAL Omega è progettato principalmente per allineare più sequenze contemporaneamente. L'MSA è un compito bioinformatico fondamentale che prevede la disposizione di diverse sequenze in modo da massimizzare la somiglianza tra posizioni omologhe.

2 Identificazione Delle Regioni Conservate

Allineando più sequenze, CLUSTAL Omega aiuta i ricercatori ad identificare le regioni conservate nell'insieme delle sequenze. Le regioni conservate spesso corrispondono ad elementi funzionali o strutturali, fornendo informazioni sul significato biologico delle sequenze.

3 Analisi Delle Relazioni Evolutive

Le sequenze allineate possono essere utilizzate per dedurre relazioni evolutive tra gli organismi o le proteine da cui derivano le sequenze. I cambiamenti nelle posizioni allineate nel tempo possono indicare divergenza o conservazione evolutiva.



MAFFT

Allineamento Di Sequenze Multiple (MSA)

MAFFT è progettato principalmente per allineare simultaneamente più sequenze di DNA, RNA o proteine. L'MSA è un passo cruciale nella bioinformatica, poichè consente il confronto di sequenze omologhe per identificare regioni conservate e dedurre relazioni evolutive.

Velocità Ed Efficienza

MAFFT è noto per la sua alta velocità, che lo rende particolarmente adatto per allineare grandi set di dati con un numero considerevole di sequenze. L'algoritmo utilizza tecniche euristiche e di ottimizzazione per ottenere un allineamento efficiente senza compromettere la precisione.

Precisione E Robustezza

Nonostante la sua velocità, MAFFT mantiene un elevato livello di precisione di allineamento. L'algoritmo impiega strategie di allineamento progressive e impiega vari metodi per migliorare la robustezza del processo di allineamento.

Galaxy



1

Natura Open Source

Galaxy è un progetto open source, il che significa che il suo codice sorgente è liberamente disponibile per essere visualizzato, modificato e distribuito dagli utenti. La natura aperta incoraggia la collaborazione, i contributi della comunità e lo sviluppo di estensioni e plugin.

2

Interfaccia Intuitiva

Una delle caratteristiche distintive di Galaxy è la sua interfaccia user-friendly, progettata per rendere accessibili analisi bioinformatiche complesse a utenti con diversi livelli di competenza. L'interfaccia grafica elimina la necessità per gli utenti di avere competenze di programmazione, rendendolo adatto ai principianti in bioinformatica.

3

Gestione Del Flusso Di Lavoro

Galaxy consente agli utenti di creare e gestire flussi di lavoro unendo diversi strumenti in una sequenza logica. I flussi di lavoro possono essere salvati, riutilizzati e condivisi, promuovendo la riproducibilità e la collaborazione.

4

Riproducibilità

Galaxy enfatizza la riproducibilità consentendo agli utenti di condividere flussi di lavoro completi insieme a set di dati e versioni degli strumenti. Questa funzionalità garantisce che le analisi possano essere rieseguite con gli stessi parametri, ottenendo risultati coerenti.



DESeq2

Analisi Dell'Espressione Genica Differenziale

DESeq2 è specificamente progettato per l'identificazione dei geni espressi in modo differenziale (DEG) confrontando i dati RNA-seq tra diverse condizioni sperimentali o gruppi di campioni.

Modellizzazione Statistica

DESeq2 utilizza un approccio di modellizzazione statistica basato su una distribuzione binomiale negativa per tenere conto della variabilità intrinseca nei dati di conteggio generati dagli esperimenti RNA-seq.

Normalizzazione

DESeq2 incorpora metodi di normalizzazione per tenere conto delle variazioni nella profondità di sequenziamento tra i campioni. La normalizzazione garantisce che l'analisi rifletta accuratamente le differenze biologiche piuttosto che artefatti tecnici.

Strumenti Di Visualizzazione

DESeq2 include strumenti di visualizzazione come heatmap, grafici MA (log-fold change vs. mean average) e altri grafici diagnostici per aiutare nell'interpretazione dei risultati.



PyMOL



Visualizzazione 3D

PyMOL eccelle nella visualizzazione tridimensionale delle strutture molecolari, permettendo agli utenti di esplorare e analizzare interattivamente proteine, acidi nucleici, piccole molecole e altri complessi biomolecolari.



Allineamento Delle Strutture

PyMOL permette agli utenti di allineare multiple strutture proteiche, facilitando il confronto di molecole simili o correlate. Gli allineamenti strutturali aiutano ad identificare regioni conservate e comprendere somiglianze funzionali.



Scripting E Automazione

PyMOL è dotato di un potente linguaggio di scripting (basato su Python) che consente agli utenti di automatizzare compiti, creare visualizzazioni personalizzate e integrare PyMOL in flussi di lavoro computazionali più ampi.

HADDOCK

Docking Proteina-Proteina

HADDOCK è specializzato nella previsione delle strutture tridimensionali dei complessi proteina-proteina attraverso il docking molecolare.

Approccio High Ambiguity Driven

HADDOCK utilizza un approccio che considera la flessibilità e l'ambiguità nelle informazioni sperimentali sulle proteine interagenti.

Protocollo di Docking Flessibile

Il protocollo di docking flessibile di HADDOCK permette la modellazione dei cambiamenti conformazionali nelle proteine durante il processo di docking.

Integrazione dei Dati Sperimentali

HADDOCK può incorporare dati sperimentali come quelli provenienti dalla Risonanza Magnetica Nucleare per guidare i calcoli di docking.

