Università degli Studi di Milano - Bicocca
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di laurea in Informatica

Relazione di progetto
"Elementi di Bioinformatica"

Relazione di:
Lorenzo Salute - 894461

Anno Accademico 2024-2025

1 Introduzione

In questa relazione verrà discussa la soluzione relativa al *Tema 1* del progetto d'esame inerente al corso di "Elementi di Bioinformatica" durante l'anno accademico 2024-2025.

La soluzione è stata implementata tramite il linguaggio di programmazione *Python* sfruttando la piattaforma *Jupyter* che permette di sfruttare l'editor a blocchi *Jupyter Notebook*. In particolare, all'interno della relazione, verranno descritti:

- Parametri in input del Notebook
- Metodi/algoritmi rilevanti implementati
- Scelte effettuate e criteri utilizzati rilevanti
- Librerie utilizzate
- Formato dell'output prodotto

È possibile consultare il codice scritto sul *Notebook* al file *"main.ipynb"*.

Per semplificare l'interazione con il sistema, è stato sviluppato lo script *"runner.cpp"* che consente di eseguire il codice *Python* ed aprire il report automaticamente; di conseguenza dopo la compilazione basterà eseguire il file *.exe* ottenuto.

2 Considerazioni preliminari

Vengono proposte alcune considerazioni emerse durante lo sviluppo della soluzione ed il testing di essa:

- è stato verificato che tutte le reads nel file BAM provenissero dal range coperto dai trascritti del file GTF;

- è possibile vedere che ad ogni *query_name* corrispondono due reads, una forward ed una reverse; questo significa che ognuno dei read è di tipo *paired-end*;

- è stato verificato che lo strand del trascritto sequenziato è positivo.

3 Parametri in Input

Sono stati forniti, congiuntamente alla traccia, due file utili per lo svolgimento:

- File GTF (*annotation_one_tr_chr8.gtf*) che fornisce i trascritti di geni umani su un certo cromosoma, in questo caso il *cromosoma 8* del genoma umano; al suo interno è presente un solo trascritto per gene.

- File BAM (*sample-chr8.bam*) che fornisce gli allineamenti di reads trascrittomici sequenziati da uno dei trascritti presenti nel file GTF.

In seguito, è stato necessario recuperare, dal dataset pubblico *Ensembl Genome Browser*, il file contenente la sequenza nucleotidica del cromosoma in analisi:

- File FASTA (*Homo_sapiens.GRCh38.dna.chromosome.8.fa*) che fornisce la sequenza primaria relativa al *cromosoma 8* del genoma umano.

4 Metodi/algoritmi rilevanti

In questa sezione verranno discussi i principali metodi/algoritmi implementati nella soluzione proposta, i quali risultano essere particolarmente rilevanti per il raggiungimento del risultato.

get_spliced_reads_file (all_alignments, output_file)

Questa funzione prende in input la lista di tutti gli allineamenti derivati dal *file BAM* ed il path del file su cui scrivere l'output; l'obiettivo di questo metodo, è quello di generare il file contenente tutti i reads spliced e restituire in output la lista degli stessi.

Nel codice viene recuperata la lista di tutti i reads spliced tramite una *list comprehension*, la quale effettua un controllo sulla presenza del carattere N all'interno delle cigar string. I reads spliced, infatti, sono dei read "splittati" su esoni non contigui tra loro; fra essi esistono dei gap detti *introni* che sono riconoscibili proprio dalla presenza del carattere N nella cigar string dei reads.

Inoltre viene generata la stringa di qualità per il read, parametro necessario da salvare nel file FASTQ. Più precisamente, viene imposta una stringa di qualità di default nel caso in cui questo dato non fosse presente nel file BAM.

Viene poi effettuata la scrittura di ciascun read sul file ed infine viene restituita la lista degli allineamenti spliced.

get_transcripts (df)

Questa funzione prende in input il dataframe costruito a partire dal file GTF, con l'obiettivo di produrre in output il dizionario che ha come chiavi i trascritti e come valori le tuple che contengono la posizione di inizio del trascritto, la posizione di fine del trascritto ed il gene associato ad esso.

Nel codice viene creata una lista contenente tutte le tuple *(start, end)* degli esoni per il trascritto considerato; da quest'ultima è possibile inferire sulle posizioni di start-end del trascritto, recuperando lo start del primo esone e l'end dell'ultimo esone.

get_associated_transcripts (transcript_dict, all_alignments)

Questo metodo prende in input il dizionario dei trascritti (generato dalla funzione precedente) e la lista di tutti gli allineamenti trovati nel file BAM. L'obiettivo è quello di produrre in output un dizionario che associ a ciascun read il trascritto da cui è stato sequenziato.

Vengono analizzate le posizioni di inizio e fine dei reads sulla reference; se entrambe sono comprese all'interno del locus del trascritto, quest'ultimo sarà quello da cui è stato sequenziato il read.

get_sequentied_transcript_subprocess (bam, gtf, out)

Questa funzione sfrutta le librerie *HTSeq* e *Subprocess*. In particolare, partendo dai file BAM e GTF forniti in input, è possibile produrre automaticamente un file di output (nel

path fornito in input) in cui, ad ogni trascritto, è associato il numero di read sequenziati da esso. È facile verificare, tramite il risultato di questa funzione, la correttezza dell'output fornito dal metodo *get_associated_transcripts* descritto precedentemente.

Nel codice viene costruito il comando bash che sfrutta *HTSeq* per l'analisi dei file; quest'ultimo viene eseguito grazie alla libreria *Subprocess* che consente di generare un nuovo processo per l'esecuzione del comando fornito come argomento. I risultati del processo vengono salvati successivamente in un file di testo.

find_introns ()

Grazie a questo comando è possibile recuperare un dizionario in cui, ad ogni tupla *(start, end)*, corrispondente ad inizio e fine di ogni introne, è associato il proprio supporto, cioè il numero di reads allineati ad esso.

get_sequentied_transcript_file (transcript, gene, out_file)

Questa funzione prende in input il trascritto da cui sono stati sequenziati i reads ed il gene a cui esso è associato, generando un file testuale al path fornito contenente queste informazioni.

get_introns_support_file (all_introns, best_intron, out_file)

La funzione prende in input il dizionario degli introni e l'introne con supporto maggiore; l'output fornito è un file contenente la lista degli introni, con il supporto ad essi associato, e la descrizione dell'introne con maggior supporto. Il file viene salvato nel path dato in input.

get_alignment_file (intron_aligned_reads, reference, out_file)

Questa funzione prende in input la lista dei reads allineati all'introne considerato e la sequenza del cromosoma. L'obiettivo è quello di generare un file testuale al path dato in input, tramite il quale è possibile visualizzare graficamente gli allineamenti dei reads al cromosoma di riferimento.

All'interno del codice vengono ricostruite le porzioni di read da allineare agli esoni tenendo conto del formato della cigar string.

Successivamente viene recuperata la sottosequenza della reference a cui si va ad allineare la porzione del read in considerazione; viene poi scelto l'allineamento migliore che viene in seguito stampato sul file di testo.

write_report (transcript_f, introns_f, alignments_f, report_f)

Questa funzione ha il compito di concatenare il file contenente trascritto e gene sequenziati, il file contenente la descrizione degli introni ed il file contenente gli allineamenti testuali. L'output è un report testuale completo sui risultati dello studio.

5 Scelte effettuate e criteri utilizzati

Per la riuscita del progetto sono state sfruttate le conoscenze acquisite durante lo svolgimento delle lezioni di laboratorio.

- Nella fase iniziale si è scelto di costruire un dataframe, contenente i trascritti recuperati dal file GTF, in formato utile allo svolgimento del progetto;

- durante lo sviluppo si è prestata attenzione allo strand delle sequenze, in modo da lavorare in ogni momento su dati corretti. Il file BAM salva i reads reverse applicando su di essi la funzione di R&C mostrando subito il loro formato corretto.

- i read spliced sono stati recuperati cercando all'interno delle sequenze la presenza di caratteri *N* nelle cigar string;

- è stato verificato che le reads fetchate dal file BAM non avessero stringhe di qualità associate;

- il file FASTQ è stato formattato secondo gli standard di questo formato;

- il dizionario dei trascritti è stato creato tenendo conto dello strand degli esoni; nel caso di strand negativo le coordinate devono essere gestite; è stato scelto di ordinarle in modo crescente per facilitare la loro lettura;

- durante la generazione degli allineamenti sono state sfruttate le cigar string; in particolare sono state allineate quelle porzioni che coincidono a match (M) e saltate quelle che corrispondono ad introni (N), soft e hard clip (S, H) non sono state allineate al cromosoma;

- sono stati sfruttati dizionari per una maggiore semplicità nel recuperare i dati utili;

- sapendo che tutti i read sono stati sequenziati da un unico trascritto, la ricerca di esso è stata semplificata, analizzando soltanto che i locus dei read fossero contenuti nel locus del trascritto;

- è stato scelto di utilizzare *HTSeq* per confrontare il risultato ottenuto in modo da garantirne la correttezza;

- viene sfruttata la funzione *find_introns()* applicata al file BAM per la ricerca degli introni e di conseguenza il supporto ad essi associato. In seguito sono stati confrontati gli introni ottenuti con gli introni derivati dal file GTF;

- è stato scelto di analizzare il primo tra gli introni di supporto massimo;

- è stato sfruttato *Ensembl Genome Browser* per recuperare la sequenza del cromosoma analizzato;

- per la scrittura degli allineamenti viene sfruttato *PairwiseAligner()*;

- si è scelto di produrre in output un report in formato testuale per maggiore semplicità di scrittura e lettura;

- è stato scelto di scrivere uno script *c++* per l'esecuzione automatica del codice e l'apertura del report testuale.

6 Librerie utilizzate

```
import pysam
from pysam import AlignmentFile
import pandas as pd
import numpy as np
import re
import Bio
from Bio import Align
from collections import Counter
import itertools
import subprocess
```

La libreria *pysam* viene sfruttata per la lettura del file BAM ed il recupero degli allineamenti contenuti al suo interno.

pandas è utilizzato per la generazione del dataframe contenente i trascritti derivati dal file GTF.

La libreria *Bio*, grazie ad *Align*, è usata per la scrittura degli allineamenti testuali.

Il resto delle librerie vengono sfruttate all'interno del codice per rendere agevole lo sviluppo.

7 Formato dell'output

All'interno del codice *Python*, nella sezione *"File di output"*, sono istanziate le variabili contenenti i path sui quali vengono salvati i file generati. È possibile modificare i parametri per personalizzare la propria esperienza d'uso. Viene fornita di default una cartella *output_file* vuota nella quale vengono salvati i file testuali.

Formato report

Vengono forniti i codici del trascritto e del gene sequenziato.

```
SEQUENTIED TRANSCRIPT: ENST00000517875
ASSOCIATED GENE: ENSG00000133739
```

Gli introni vengono visualizzati nel formato *(start introne, end introne) : supporto*. Inoltre viene descritto l'introne con maggior supporto preso in analisi.

```
...
- (85107399, 85110114) : 28
- (85110180, 85112931) : 56
- (85113099, 85115099) : 52

...

Max support intron -> (85110180, 85112931), support = 56
...
```

Gli allineamenti vengono visualizzati mostrando le informazioni sul read e descrivendo le zone allineate alternandole agli introni.

```
...

read330 ------------------------

READ INFO:

Read sequence -> GCTTGCAGAGGACTTGAAGAACTAATTAATCA...
Cigar string -> 10M2715N66M2751N24M


ALIGNMENT:

target              0 GCTTGCAGAG 10
                    0 |||||||||| 10
query               0 GCTTGCAGAG 10

******* ... INTRON ... *******

target              0 GACTTGAAGAACTAATTAATCTGACTAGACTAAA...
                    0 ||||||||||||||||||||||||||||||||||...
query               0 GACTTGAAGAACTAATTAATCTGACTAGACTAAA...

...
```