Multiple Sequence Alignment (MSA) di sequenze SARS-CoV-2

Silva Edoardo 816560
Marchetti Davide 815990

A.A.: 2019/2020

Abstract

Usando i sequenziamente genomici del virus denominato Covid-19, reperibili tramite NCBI[1] e GISAID[2], abbiamo allineato un insieme di sequenze, relative a paesi del medioriente, sfruttando gli strumenti messi a disposizione dall'European Bioinformatics Institute[3] al fine di identificare differenze con la sequenza di riferimento ottenuta su un campione di Wuhan. In particolare, per l'allineamento si è scelto di utilizzare `Clustal Omega` e `MUSCLE`.

[1] `https://www.ncbi.nlm.nih.gov/genbank/sars-cov-2-seqs/`
[2] `https://www.gisaid.org/`
[3] `https://www.ebi.ac.uk/Tools/msa/`

1 Sequenze Analizzate

In aggiunta alla sequenza di riferimento di Wuhan, sono state selezionate alcune sequenze relative a paesi dell'area mediorientale.

Reference di Wuhan

- `NC_045512.2` pubblicata il 17/01/2020 (aggiornamento)

Iran

- `MT320891.2` pubblicata il 10/04/2020
- `MT281530.2` pubblicata il 04/04/2020
- `EPI_ISL_442523` sequenziata il 09/03/2020
- `EPI_ISL_437512` sequenziata il 26/03/2020

Israele

- `MT276598.1` sequenziata il 02/04/2020
- `MT276597.1` sequenziata il 02/04/2020
- `EPI_ISL_447469` sequenziata il 14/04/2020

Pakistan

- `MT262993.1` pubblicata il 25/03/2020
- `MT240479.1` pubblicata il 25/03/2020
- `EPI_ISL_417444` sequenziata il 04/03/2020

Turchia

- `MT327745.1` pubblicata il 13/04/2020,
- `EPI_ISL_437334` sequenziata il 24/03/2020
- `EPI_ISL_437317` sequenziata il 27/03/2020

Il lavoro consiste in identificare le differenze tra le sequenze e tra gli allineamenti dei 2 tool sulle medesime sequenze.

2 Descrizione

L'analisi filogenetica serve a ricostruire la storia delle mutazioni, ossia come specie antiche si sono evolute in quelle moderne.
I geni sono composti da sequenze di Acido Desossiribonucleico (DNA) (o RNA in caso di alcuni virus), composto da uno zucchero(dessossiribosio) che unisce un gruppo fosfato ad una base azotata per comporre nucleotidi (i nucleotidi sono unità ripetitive costitutive degli acidi nucleici) che compongono il genoma.
Un GENE non è nient' altro che una particolare sequenza di DNA, che codifica l' informazione in un linguaggio a quattro lettere, nel quale ogni lettera è rappresentata da una base.
Il genotipo di un individuo è dato dal suo corredo genetico. Il fenotipo, invece, è l'insieme dei caratteri che l'individuo manifesta:dipende dal suo genotipo, dalle interazioni fra geni e anche da fattori esterni.

Il DNA viene utilizzato:

1. trascrivendolo in RNA
2. rendere l' RNA stabile aggiungendo ai limiti 7-metilguanosina (7mGTP)
3. trascritto il risultato precedente in amminoacidi che creeranno proteine.

3 Codice

Il codice è in python ed esegue:

1. pulisce la cartella di output per far spazio ai nuovi file, usando la libreria os.
2. per ogni coppia di file in input contenendo l'analisi di allineamento eseguita usando i due tool:
   (a) salva il nome del file json in ooutput seguendo l'analisi.
   (b) ne compara le differenze.

3.1 librerie

- **re:** usato in **parsers.py** per dividere ogni linea tra: [id_sequenza, sequenza, posizioni] nelle funzioni di parsing.
- **hashlib:** usato in **utils.py** per creare l'hash da inserire come nome ai file JSON di output.
- **json:** usato in **utils.py** per elaborare ulteriormente l'uotput al fine di comparare i 2 tools di allineamento.
- **datetime:** usato in **utils.py** per prendere il tempo da inserire nell'hash, al fine di rendere unico l'output ed evitare sovrascritture.
- **os:** usato in **utils.py, main.py** per gestire input e output. Anche per pulire la cartella input nel main.

3.2 descrizione metodi

- **runClustal**(inputFile, reference_id, nseq = 3): funzione che esegue il parsing del file di allineamento clustal (inputFile), esegue il parsing degli allineamenti e li salva nel file di output.
  nseq serve al parser in quanto la classe **ClustalParser** richiede il numero di sequenze da elaborare nelle sue funzioni.
- **runMuscle**(inputFile, reference_id, nseq = 3): funzione che esegue il parsing del file di allineamento muscle (inputFile), esegue il parsing degli allineamenti e li salva nel file di output.

- **parse**(self, filename, reference=None, list=[]): legge il file in ingresso e restituisce il file in input suddiviso in: reference, sequenze, lunghezza sequenze. svolge la sua funzione sfruttando il metodo: **parseLines(self, lines)**
- **save**(alignment, analyzer, reference_id=None, tool=None, path=None): crea file json di output chiamato reference_id_hash-sha1 del tempo di inzio esecuzione concatenato con sequences_ids delle sequenze.
- **jsonComp**(file1, file2): funzione cre prende i 2 file json generati dall'elaborazione clustal e muscle e ritorna le differenze tra i 2 oggetti al fine di compararli.
- **saveCompareFile**(filename = "differences.txt", country = "", diff = [], path='output'): file che divide per paese prende la lista di differenze tra gli allineamenti clustal e muscle e li aggiunge al file di output "differences.txt" per mostrarli.

4 Output

Come output vengono prodotti dei file json corrispondenti al nome reference_seq_hash.hexdigest(): l'id della reference accostata all'hash sha1 con encoding utf8 dell'id delle sequenze usate concatenate al tempo di inizio lavorazione (aggiunto per evitare di sovrascrivere l'output dello stessa serie di sequenze sia con muscle che con clustal).
l'oggetto JSON è così composto: { "reference": reference_Id,
"analyzed_sequences": insieme di sequenze analizzate,
"unmatches": altri oggetti JSON =
{ id (hash inizio mismatch + fine mismatch): "from": inizio mismatch,
"to": fine mismatch,
"sequences": lista sequenze in diverse tra loro in quell'intervallo }
}

5 Conclusioni

Clustal tronca id troppo lunghi, i 2 tool di allineamento non sono identici, ...

6 references