Metodi AF e Mash

Alessandro Ricchetti Marco Cappiello

***Abstract*—Le analisi di sequenza senza allineamento (AF) sono state applicate a problemi che vanno dalla filogenesi dell'intero genoma alla classificazione delle famiglie di proteine, all'identificazione di geni trasferiti orizzontalmente e al rilevamento di sequenze ricombinate. In questo paper è presente anche un benchmarking avvenuto su 24 tool per un totale di 74 varianti di tecniche AF su cinque possibili applicazioni nella ricerca. Inoltre, è stato studiato il funzionamento del tool Mash ed è stato applicato su geni di riferimento.**

I. Introduzione

L'analisi comparativa delle sequenze di DNA e aminoacidi è di fondamentale importanza nella ricerca biologica, in particolare nella biologia molecolare e nella genomica. È il primo passo fondamentale per l'analisi evolutiva molecolare, la previsione della funzione del gene e della regione regolatoria, l'assemblaggio di sequenza, la ricerca dell'omologia, la previsione della struttura molecolare, la scoperta del gene e l'analisi della relazione struttura-funzione delle proteine.

Tradizionalmente, il confronto tra sequenze si basava sull'allineamento a coppie o sull'allineamento di sequenze multiple (MSA). Gli strumenti software per l'allineamento di sequenze, come BLAST e CLUSTAL, sono alcuni degli approcci più utilizzati nella bioinformatica. Sebbene le tecniche basate sull'allineamento rimangano generalmente i riferimenti per il confronto delle sequenze, i metodi basati su MSA non sono scalabili con i dataset molto grandi oggi disponibili.

II. Problemi con metodi basati su allineamento

Sebbene gli approcci basati sull'allineamento siano più accurati e potenti per il confronto delle sequenze quando sono fattibili, le loro applicazioni sono limitate in alcune situazioni. Le tecniche basate sull'allineamento hanno dimostrato di essere imprecise in scenari di bassa identità di sequenza (ad esempio, sequenze regolatrici di geni e omologhi di proteine lontanamente correlati). Inoltre, gli algoritmi di allineamento presuppongono che l'ordine lineare dell'omologia sia conservato all'interno delle sequenze confrontate, quindi questi algoritmi non possono essere applicati direttamente in presenza di riordinamenti di sequenza (ad esempio, ricombinazione e scambio di domini proteici) o di trasferimento orizzontale nei casi in cui vengono elaborati set di dati di sequenze su larga scala, ad esempio per la filogenetica dell'intero genoma. Inoltre, l'allineamento di due sequenze di DNA lunghe milioni di nucleotidi non è fattibile nella pratica. Bisogna anche tenere in considerazione che alcune parti dei genomi potrebbero non essere sequenziate a causa della distribuzione stocastica delle letture lungo i genomi e della difficoltà di sequenziare alcune parti dei genomi, soprattutto quando il coverage è relativamente basso. Le regioni non codificanti, come le regioni regolatorie dei geni, non sono altamente conservate, tranne alcune regioni funzionali come i siti di legame della trascrizione, e non possono essere allineate in modo affidabile. Pertanto, gli approcci basati sull'allineamento non sono adatti per studiare l'evoluzione delle regioni regolatrici dei geni. Infine, il calcolo di un allineamento accurato di sequenze multiple è un problema NP-hard, il che significa che l'allineamento non può essere calcolato in tempi realistici. Quindi, i confronti di genomi e metagenomi basati su grandi quantità di dati di sequenziamento di nuova generazione (NGS) pongono sfide significative agli approcci basati sull'allineamento, a causa delle enormi dimensioni dei dati e della lunghezza relativamente breve delle letture. Pertanto, in alternativa all'allineamento delle sequenze, sono stati sviluppati molti approcci definiti alignment-free (AF) riguardo l'analisi delle sequenze.

III. cos'è il confronto AF

Gli approcci alignment-free al confronto tra sequenze possono essere definiti come qualsiasi metodo di quantificazione della somiglianza/dissimilarità tra sequenze che non utilizza o produce allineamento in nessuna fase dell'applicazione dell'algoritmo. Fin dall'inizio, tale restrizione pone gli approcci AF in una posizione favorevole: i metodi AF non si basano sulla programmazione dinamica, sono meno costosi dal punto di vista computazionale (poiché la loro complessità è generalmente lineare e dipende solo dalla lunghezza della sequenza della query) e quindi sono adatti per i confronti dell'intero genoma. I metodi senza allineamento sono anche resistenti agli eventi di rimescolamento e ricombinazione e sono applicabili quando la bassa conservazione della sequenza non può essere gestita in modo affidabile dall'allineamento. Infine, a differenza dei metodi basati sull'allineamento, non dipendono da ipotesi sulle traiettorie evolutive dei cambiamenti di sequenza.

IV. Confronto tra metodi basati su allineamento e metodi AF

Metodi basati su allineamento:

- Questi metodi presuppongono che le regioni omologhe siano contigue (con spazi vuoti)
- Calcola tutti i possibili confronti a coppie di sequenze; quindi è computazionalmente costoso.
- Approccio consolidato nella filogenomica
- Richiede modelli di sostituzione/evoluzione
- Sensibile alla variazione stocastica delle sequenze, alla ricombinazione, al trasferimento genetico orizzontale (o

laterale), all'eterogeneità del tasso e a sequenze di lunghezza diversa, soprattutto quando la somiglianza si trova nella "zona crepuscolare".
- La migliore pratica utilizza algoritmi di inferenza con complessità di almeno $O(n^2)$; meno efficienti in termini di tempo
- La significatività statistica della relazione tra i punteggi di allineamento e l'omologia è difficile da valutare.
- Si affida alla programmazione dinamica (costosa dal punto di vista computazionale) per trovare un allineamento che abbia un punteggio ottimale.

Metodi AF:

- Non presuppone la contiguità delle regioni omologhe.
- Basato sulle occorrenze di sotto-sequenze; composizione; computazionalmente poco costoso, può essere intensivo in termini di memoria.
- L'applicazione alla filogenomica è relativamente recente e limitata; necessita di ulteriori test di robustezza e scalabilità.
- Meno dipendente da modelli di sostituzione/evoluzione
- Meno sensibile alla variazione stocastica delle sequenze, alla ricombinazione, al trasferimento genetico orizzontale (o laterale), all'eterogeneità del tasso e alle sequenze di lunghezza diversa.
- Algoritmi di inferenza tipicamente $O(n^2)$ o meno; più efficienti in termini di tempo
- Soluzioni esatte; la significatività statistica delle distanze di sequenza (e del grado di somiglianza) può essere facilmente valutata
- evita la programmazione dinamica, costosa dal punto di vista computazionale, indicizzando il numero di parole o le posizioni nello spazio frattale.

V. Classificazione metodi

La maggior parte di questi metodi si basa sulle caratteristiche statistiche delle parole o sul loro confronto e la loro scalabilità consente di applicarli a set di dati molto più ampi rispetto ai metodi convenzionali basati sulla MSA. È stata sviluppata un'ampia gamma di approcci AF risguardo il confronto tra sequenze. Questi approcci includono metodi basati sul conteggio delle parole o dei k-mer, sulla lunghezza delle sottostringhe comuni, sui micro-allineamenti, sulle rappresentazioni delle sequenze basate sulla teoria del caos, sui momenti delle posizioni dei nucleotidi, sulle trasformazioni di Fourier, sulla teoria dell'informazione e sui sistemi a funzione iterata.

I metodi più diffusi basati sulle frequenze di k-mer/parole includono il profilo di frequenza delle caratteristiche (FFP), il vettore di composizione (CV), la distribuzione del tempo di ritorno (RTD), la rappresentazione del gioco del caos delle frequenze (FCGR) e AFKS.

I metodi basati sulla lunghezza delle sottostringhe comuni utilizzano la somiglianza e le differenze delle sottostringhe in una coppia di sequenze. Questi algoritmi sono stati utilizzati soprattutto per l'elaborazione delle stringhe. Esempi di tecniche basate sulle sottostringhe comuni sono: sottostringa comune media (ACS), k-mismatch media delle sottostringhe comuni (kmacs) e la Distanze di mutazione (Kr).

Alcuni metodi basati sul numero di corrispondenze di parole (spaziate) sono Mash, Slope-Tree e Skmer.

I metodi basati su microallineamenti non sono esattamente privi di allineamenti. Utilizzano semplici microallineamenti privi di gap in cui le sequenze devono corrispondere in determinate posizioni predefinite. Le posizioni allineate nelle restanti posizioni dei microallineamenti, in cui sono consentite le mancate corrispondenze, vengono poi utilizzate per l'inferenza della filogenesi. In questa categoria rientrano Cophylog, andi e Prot-SpaM.

La teoria dell'informazione ha fornito metodi di successo per l'analisi e il confronto delle sequenze senza allineamento. Le applicazioni attuali della teoria dell'informazione comprendono la caratterizzazione globale e locale di DNA, RNA e proteine, la stima dell'entropia del genoma e la classificazione di motivi e regioni. Tecniche basate sulla teoria dell'informazione sono la correlazione base-base (BBC), metodi basati su compressione e sulla correlazione informativa e correlazione informativa parziale (IC-PIC).

VI. Benchmarking

Per il benchmarking sono stati impiegati 12 dataset suddivisi in 5 categorie:

- Protein Sequence Classification
- Gene Tree Inference
- Regulatory Sequences
- Genome-based Phylogeny
- Horizonal Gene Transfer

In totale sono stati testati 24 tool e le rispettive varianti se presenti, che utilizzano metodi differenti:

A. Exact k-mer count

- AAF: AAF ricostruisce una filogenesi direttamente dalle letture di sequenziamento di nuova generazione non assemblate. In particolare, AAF calcola la distanza di Jaccard tra gli insiemi di k-meri di due campioni di brevi letture di sequenza.
- AFKS: è un pacchetto per il calcolo di 33 misure di dissimilarità/distanza tra sequenze nucleotidiche o proteiche basate su k-mer. Lo strumento determina la dimensione ottimale dei k-mer per le sequenze in ingresso e calcola misure di dissimilarità/distanza tra i conteggi dei k-mer che includono pseudoconti (aggiungendo 1 a ciascun conteggio dei k-mer).
- alfpy: fornisce 38 misure di dissimilarità AF con cui calcolare le distanze tra sequenze nucleotidiche o proteiche. Lo strumento include 25 misure basate su k-mer, otto misure teoriche dell'informazione, tre misure basate su grafi e due misure ibride.
- CAFÈ: è un pacchetto per il calcolo efficiente di 28 misure di dissimilarità AF, tra cui 10 misure convenzionali basate sul numero di k-mer. Offre inoltre 15 misure basate sulla presenza/assenza di k-mer. CAFE consente di utilizzare come input sia sequenze di genoma assemblate sia

letture shotgun di sequenziamento di nuova generazione non assemblate.
- FFP: stima le distanze tra sequenze di nucleotidi o amminoacidi. Lo strumento calcola il conteggio di ciascun k-mer e poi divide il conteggio per il totale di tutti i k-mer per normalizzare le frequenze di una determinata sequenza.
- jD2Stat: utilizza una serie di statistiche D2 per estrarre i k-meri da un insieme di sequenze biologiche e generare le distanze a coppie per ogni possibile coppia come matrice.

B. Teoria dell'informazione

- LZW-Kernel: classifica le sequenze proteiche e identifica l'omologia proteica remota tramite una funzione kernel convoluzionale. LZW-Kernel sfrutta i blocchi di codice rilevati dai compressori universali di testo Lempel-Ziv-Welch (LZW) e costruisce una funzione kernel a partire da essi. Il tool può anche stimare la distanza tra le sequenze proteiche utilizzando le distanze di compressione normalizzate (LZW-NCD).

C. Inexact k-mer count

- Spaced: è simile ai metodi precedenti che confrontano la composizione k-mer di sequenze di DNA o proteine. Tuttavia, il programma utilizza le cosiddette "parole spaziate" al posto dei k-meri. Il vantaggio di utilizzare parole spaziate invece di k-mers esatte è che i risultati ottenuti sono statisticamente più stabili.

D. k-mer count

- kWIP: stima la dissimilarità genetica tra i campioni direttamente dai dati di sequenziamento di nuova generazione senza la necessità di un genoma di riferimento. Lo strumento utilizza la metrica del prodotto interno ponderato (WIP), che mira a ridurre l'effetto del rumore tecnico e biologico e ad elevare il segnale genetico rilevante ponderando i conteggi dei k-mer in base alla loro entropia informativa nel set di analisi.

E. Maximal length of exact common substring

- ALFRED-G: utilizza un algoritmo efficiente per calcolare la lunghezza delle sottostringhe comuni massime con k errori di corrispondenza tra due sequenze. In particolare il programma calcola la lunghezza delle coppie di parole massime, una parola per ciascuna sequenza, con un massimo di k corrispondenze.
- kmacs: confronta due sequenze di DNA o proteine cercando le più lunghe sottostringhe comuni con un massimo di k corrispondenze. Per ogni posizione i in una sequenza, il programma identifica la più lunga coppia di sottostringhe con un massimo di k mismatch, partendo da i nella prima sequenza e da un punto qualsiasi della seconda sequenza. La lunghezza media di queste coppie di sottostringhe viene quindi utilizzata per definire la distanza tra le sequenze.
- kr: stima la distanza evolutiva tra i genomi calcolando il numero di sostituzioni per sito. L'estimator del tasso di sostituzioni tra due sequenze non allineate dipende da un modello matematico di evoluzione delle sequenze di DNA e dalla lunghezza media della sottostringa unica più breve (shustring).
- Underlying Approach: stima le distanze filogenetiche tra interi genomi utilizzando le statistiche di corrispondenza delle parole comuni tra due sequenze. Le statistiche di corrispondenza sono derivate da un piccolo insieme di sottoparole indipendenti di lunghezza variabile.

F. micro-alignments

- andi: stima le distanze filogenetiche tra genomi di specie strettamente correlate identificando coppie di corrispondenze di parole uniche massime a una certa distanza l'una dall'altra e sulla stessa diagonale nella matrice di confronto di due sequenze.
- co-phylog: stima le distanze evolutive tra sequenze genomiche assemblate o non assemblate di organismi microbici strettamente correlati. Lo strumento trova allineamenti brevi, senza gap, di lunghezza fissa e costituiti solo da coppie di nucleotidi corrispondenti, ad eccezione della posizione centrale di ciascun allineamento, in cui sono consentite le mancate corrispondenze. Le distanze filogenetiche sono stimate dalla frazione di tali allineamenti per i quali la posizione centrale è un mismatch.
- FSWM/Read-SpaM: stima la distanza filogenetica tra due sequenze di DNA. Il programma definisce innanzitutto un modello binario fisso P di lunghezza l che rappresenta le "posizioni di corrispondenza" e le posizioni "don't care". Quindi, identifica tutti gli "Spaced-word Matches" (SpaM) rispetto a P, cioè allineamenti locali senza lacune delle sequenze di input di lunghezza l, con nucleotidi corrispondenti nelle "posizioni di corrispondenza" di P e possibili mismatch nelle posizioni "non importanti". Per stimare la distanza tra due sequenze di DNA, gli SpaM con bassa somiglianza complessiva vengono scartati e gli SpaM rimanenti vengono utilizzati per stimare la distanza tra le sequenze, in base al rapporto di mismatch nelle posizioni "don't care".
- Multi-SpaM: analogamente a FSWM, inizia con un modello binario P di lunghezza l che rappresenta le "posizioni di corrispondenza" e le "posizioni non interessate".
- phylonium: stima le distanze filogenetiche tra genomi strettamente correlati. Lo strumento seleziona un riferimento da un dato insieme di sequenze e trova segmenti di sequenza corrispondenti di tutte le altre sequenze rispetto a questo riferimento.

G. Number of word matches

- mash: stima la distanza evolutiva tra sequenze di nucleotidi o aminoacidi. Lo strumento utilizza l'algoritmo MinHash per ridurre le sequenze in ingresso a piccoli "sketch", che consentono di stimare rapidamente la distanza con bassi requisiti di memoria e archiviazione. Per creare uno sketch, ogni k-mer di una sequenza viene sottoposto a hash.

- Slope-SpaM: Slope-SpaM stima la distanza filogenetica tra due sequenze di DNA calcolando il numero N_k di corrispondenze k-mer per una serie di valori di k. Il tool può anche utilizzare le corrispondenze di parole spaziate (SpaM)
- Skmer: stima le distanze filogenetiche tra campioni di letture di sequenziamento grezze. Skmer esegue internamente mash per calcolare il profilo k-mer degli skim del genoma e la loro intersezione, e stima le distanze genomiche correggendo l'effetto della bassa copertura e dell'errore di sequenziamento.

H. Return time distribution

- RTD-Phylogeny: calcola le distanze filogenetiche tra sequenze nucleotidiche o proteiche in base al tempo necessario per la ricomparsa di k-meri. Il tempo si riferisce al numero di residui per la comparsa successiva di particolari k-mer. Pertanto, l'occorrenza di ciascun k-mer in una sequenza viene calcolata sotto forma di distribuzione del tempo di ritorno (RTD).

I. SNP count

- kSNP3: identifica i polimorfismi a singolo nucleotide (SNP) in un insieme di sequenze genomiche senza la necessità di un allineamento del genoma o di un genoma di riferimento.

J. Variable length-word count

- EP-sim: EP-sim calcola una distanza AF tra sequenze nucleotidiche o aminoacidiche in base ai profili entropici (EP). L'EP è una funzione della posizione genomica che cattura l'importanza di quella regione rispetto all'intero genoma. Per ogni posizione, calcola un punteggio basato sull'entropia di Shannon della distribuzione delle parole e sul conteggio delle parole a lunghezza variabile.

VII. Risultati per dataset

A. Protein Sequence Classification

Il riconoscimento delle relazioni strutturali ed evolutive tra le sequenze di aminoacidi è fondamentale per la comprensione della funzione e dell'evoluzione delle proteine. L'area sotto la curva (AUC) caratteristica operativa del ricevitore (ROC) , che indica se un metodo è in grado di discriminare tra sequenze proteiche omologhe e non omologhe, ha mostrato che il tool AFKS ha le prestazioni migliori. AFKS, con parametri impostati sulla distanza simratio e una lunghezza di parola di k = 2, è lo strumento più performante sia per i dataset a bassa che ad alta identità di sequenza. Per quest'ultimo tipo di dataset, il metodo produce i valori di AUC più elevati in tutti e quattro i livelli strutturali, con un'AUC media di 0,798 ± 0,139. alfpy–google si posiziona al secondo (0,738 ± 0,091) e al quarto posto (0,778 ± 0,142) rispettivamente per i dataset a bassa e alta identità di sequenza. In particolare, le prime sette posizioni in classifica in entrambi i dataset a bassa e alta identità di sequenza sono occupate, anche se in ordine diverso, dalle stesse misure di AFKS e alfpy. In generale, gli strumenti testati raggiungono un maggiore potere discriminatorio nel riconoscimento delle relazioni strutturali (AUC medie più elevate) nel dataset ad alta identità di sequenza rispetto al dataset a bassa identità.

B. Gene tree inference

A causa della limitata quantità di informazioni di sequenza disponibili, i gene trees sono tipicamente più difficili da ricostruire rispetto agli species tree. L'obiettivo è dedurre le relazioni filogenetiche di sequenze omologhe sulla base di una collezione di filogenesi SwissTree ad alta confidenza che rappresentano diversi tipi di sfide per la previsione dell'omologia, ad esempio, numerose duplicazioni di geni e HGT (Horizontal Gene Transfer). Come misura dell'accuratezza, viene impiegata la distanza normalizzata Robinson-Foulds (nRF) tra gli alberi ricostruiti con i metodi AF in esame e gli alberi di riferimento. La distanza nRF ha valori compresi tra 0 e 1, con 0 che indica topologie di alberi identiche e 1 che indica le topologie più diverse. Nessuno dei metodi AF testati è stato in grado di dedurre perfettamente la rispettiva topologia dell'albero di riferimento per nessuna delle 11 famiglie di geni. jD2Stat (con i valori dei parametri n = 1 e k = 5) è stato lo strumento più accurato nel test. Questo metodo ha ottenuto i valori più bassi di nRF (massima accuratezza) tra tutti i metodi testati in media su tutte le 11 famiglie di geni di riferimento (nRF = 0,3296 ± 0,1511).

C. Regulatory Sequences (CRM)

L'analisi delle sequenze regolatorie dei geni è un altro ambito in cui i metodi AF sono popolari, poiché la somiglianza tra questi elementi è solitamente bassa e gli allineamenti non riescono a rilevarla correttamente. Si impiega una procedura di benchmarking e dataset di riferimento di moduli cis-regolatori (CRM), dimostrando che gli algoritmi di allineamento producono risulatti peggiori rispetto ai metodi AF nel riconoscere i CRM funzionalmente correlati. Un CRM può essere definito come una sequenza non codificante contigua che contiene più siti di legame per i fattori di trascrizione e regola l'espressione di un gene. Tuttavia, nessuno dei metodi AF ha prodotto risultati perfetti per nessuna delle sette combinazioni di dati tessuti/specie. Il tool alfpy impostato su tre misure di distanza - Canberra, Chebyshev e divergenza Jensen-Shannon - ha catturato il maggior numero (in media su 7 campioni di tessuto) di elementi regolatori funzionalmente correlati. La selezione della distanza di Canberra (lunghezza della parola k = 2) ha riconosciuto correttamente il 73,6% ± 10,54% dei CRM, cogliendo la più alta correlazione funzionale in tre dei sette dataset.

D. Genome-based phylogeny (assembled genomes, Raw sequencing reads)

I metodi AF sono particolarmente popolari negli studi filogenetici basati sui genomi a causa di diversi motivi:

- notevoli dimensioni dei dati di input
- tassi variabili di evoluzione tra i genomi

- complessa corrispondenza delle parti di sequenza, spesso risultante da riarrangiamenti del genoma come inversioni, traslocazioni, fusioni cromosomiche, fissazioni cromosomiche e traslocazioni reciproche

È stata valutata la capacità dei metodi AF di dedurre alberi di specie utilizzando dati di riferimento di diversi gruppi tassonomici, tra cui batteri, animali e piante. Sono stati utilizzati genomi completamente assemblati e letture simulate di sequenziamento di nuova generazione non assemblate a diversi livelli di coverage

1) assembled genomes: Sono stati testati 23 tool AF (70 varianti in totale) nell'inferenza filogenetica utilizzando il mtDNA completo di 25 specie ittiche del sottordine Labroidei. La migliore accuratezza è stata raggiunta da nove strumenti AF (19 varianti), che ha generato topologie di alberi quasi identiche all'albero di riferimento dei Labroidei (nRF = 0,05). I risultati differiscono solo nell'ordine di speciazione di tre specie ittiche strettamente correlate appartenenti alla tribù Tropheini della famiglia Pseudocrenilabrinae. Le stesse specie sono state collocate erroneamente nelle topologie generate da altre 39 varianti di strumenti che occupavano tutte il secondo posto nella classifica di riferimento (nRF = 0,09). Questi metodi hanno inoltre sbagliato a collocare le specie all'interno delle famiglie Pomacentridae ed Embiotocidae. Questi risultati indicano che la maggior parte dei metodi AF deduce alberi in generale accordo con l'albero di riferimento dei genomi mitocondriali. Inoltre, sono state confrontate le prestazioni dei metodi AF nell'inferenza filogenetica con genomi batterici più grandi di Escherichia coli/Shigella e con genomi nucleari di specie vegetali. Sette strumenti (nove varianti) non hanno potuto essere testati su tutte e tre le serie di genomi completi, poiché i programmi non hanno completato le analisi. I restanti 16 strumenti (61 varianti) determinano distanze nRF maggiori, quindi risultati peggiori. Sebbene gli strumenti testati mostrino distanze nRF simili per i genomi batterici e vegetali in generale, gli strumenti più performanti sono diversi tra i due dataset. Ad esempio, phylonium e andi, sviluppati per il confronto filogenetico di organismi strettamente imparentati, sono gli strumenti con le migliori prestazioni per i dataset di E. coli/Shigella, mentre sui dataset delle piante entrambi gli strumenti hanno prestazioni scarse. I tool più performanti per dataset sulle piante sono co-phylog, mash e Multi-SpaM, che hanno recuperato quasi perfettamente la topologia dell'albero di riferimento delle specie vegetali (con un nRF = 0,09 per tutti e tre i programmi).

2) raw sequencing reads: È stata ancha testata l'accuratezza degli strumenti AF nell'inferenza filogenetica basata su letture di sequenziamento simulate e non assemblate, rappresentate da sette diversi livelli di copertura di sequenziamento, provenienti da E. coli/Shigella e da un insieme di specie vegetali. Non sono state osservate differenze nei valori di nRF tra i risultati basati sui genomi di E. coli/Shigella non assemblati e assemblati, indicando che gli strumenti AF hanno mostrato prestazioni uguali per i genomi non assemblati e assemblati. Al contrario, gli strumenti testati hanno mostrato prestazioni inferiori (cioè valori di nRF più elevati) nella ricostruzione filogenetica senza assemblaggio delle specie vegetali. I tool andi e co-phylog sono gli strumenti più accurati nel dataset di E. coli/Shigella, con una distanza media nRF di 0,21 ± 0,14. Per i dataset vegetali non assemblati, mash è lo strumento più accurato, ovvero quello con la distanza nRF più breve tra gli alberi dedotti e l'albero di riferimento. Per il livello di coverage più basso (0,015625), mash permette ancora di dedurre alberi con distanze nRF medie di 0,27 dall'albero di riferimento. In generale, il mash mostra le migliori prestazioni a sei dei sette livelli di coverage (da 0,015625 a 0,5). Per il dataset di E. coli/Shigella non assemblato, mash si classifica in seconda posizione. In particolare, per una coverage 0,25 nel dataset sulle piante, mash ha inferito la topologia dell'albero in perfetto accordo con l'albero di riferimento (nRF = 0). Tuttavia, le sue prestazioni diminuiscono leggermente per livelli di coverage più elevati (con nRF di 0,09 e 0,18 per copertura 0,5 e 1, rispettivamente). La migliore accuratezza al livello di coverage più alto (1×) è stata ottenuta da co-phylog (nRF = 0,09). Se si considerano gli strumenti più generali applicati a tutti i dataset di riferimento testati, mash si posiziona al primo e al secondo posto rispettivamente per la filogenesi senza assemblaggio delle piante e di E. coli/Shigella.

E. Horizontal gene transfer

Sono stati simulati cinque insiemi di 33 genomi, ciascuno con diverse estensioni di HGT determinate dal numero medio di eventi HGT per iterazione (l = 0, 250, 500, 750 e 1.000; l è il numero di eventi HGT tentati nell'insieme ad ogni iterazione del processo di simulazione dell'evoluzione del genoma). Gli strumenti AFKS (misura di Markov, con una lunghezza di parola di k = 12) e mash (k = 17-24) hanno raggiunto la massima accuratezza generale ottenendo l'nRF medio più basso (0,05 ± 0,05) e un perfetto accordo topologico con gli alberi di riferimento alle due frequenze più basse di HGT simulato (l = 0 e 250). Per la maggior parte dei metodi AF, l'accuratezza dell'inferenza filogenetica diminuisce con l'aumentare dell'estensione dell'HGT. Tuttavia, le sette applicazioni software con le migliori prestazioni, AFKS, mash, CAFE, alfpy, FFP, jD2Stat e ALFRED-G, sono state in grado di ricostruire l'albero di riferimento con poche incongruenze a quasi tutti i livelli di frequenza di HGT (nRF $\leq$ 0,1 a l $\leq$ 750), tranne che per le frequenze più elevate di HGT simulato, dove la distanza nRF era nell'intervallo 0,13-0,17. È interessante notare che le misure di distanza AF di base (distanze euclidee, Manhattan, Canberra e LCC) implementate in alfpy raggiungono un nRF medio più basso (0,07 ± 0,06) e un nRF minimo a un livello di frequenza HGT più alto (nRF = 0,13) rispetto agli strumenti AF progettati per la ricostruzione filogenetica di interi genomi (co-phylog, FSWM, Multi-SpaM e kr), che sorprendentemente sono risultati relativamente imprecisi (nRF maggiore di 0,2 per diversi valori di l). Per valutare le prestazioni dei metodi AF con i dati di sequenza del mondo reale, è stato utilizzato innanzitutto un superalbero di riferimento di 27 genomi di E. coli e Shigella, generato

sulla base di migliaia di alberi di proteine a copia singola. Per questo dataset, gli strumenti progettati per la filogenetica dell'intero genoma hanno ottenuto valori di nRF inferiori rispetto alle misure di distanza AF di base; undici strumenti per la filogenetica dell'intero genoma hanno occupato le prime sei posizioni. Tre di questi metodi, andi, co-phylog e phylonium, hanno raggiunto la massima accuratezza, con un nRF minimo di 0,08. La maggior parte delle misure di AF implementate in AFKS, alfpy e CAFE si è classificata al 10° posto e ha portato alla ricostruzione di alberi di specie imprecisi, in cui metà delle bipartizioni non erano presenti nell'albero di riferimento (nRF = 0,5).

F. Risultati

I risultati hanno mostrato che nessun singolo metodo ha ottenuto le migliori prestazioni in tutti i dataset testati. Tuttavia, alcuni strumenti si sono classificati tra i primi cinque più spesso di altri. Ad esempio, se si considerano i benchmark su scala genomica, che comprendono 8 dataset delle categorie filogenesi dell'intero genoma e HGT, gli strumenti sviluppati per i confronti genomici sono risultati tra i primi cinque: mash (8 volte), co-phylog e Skmer (7 volte), FFP (6 volte) e FSWM/Read-SpaM (5 volte). Poiché mash è l'unico metodo che si colloca tra i primi 5 strumenti con le migliori prestazioni su tutti i dataset di benchmarking su scala genomica, è particolarmente adatto per i confronti di sequenze genomiche, indipendentemente dall'intervallo filogenetico e dalla tecnologia utilizzata per ottenere i dati (ad esempio, short read o contig assemblati). La maggior parte degli approcci AF (14 su 21 tool o, più specificamente, 56 su 68 varianti di strumenti) ha ottenuto risultati particolarmente buoni, anche se non perfetti, nell'inferenza filogenetica di genomi mitocondriali di diverse specie ittiche, producendo alberi generalmente coerenti (nRF ¡ 0,1) con la filogenesi di riferimento. Tuttavia, i risultati sul confronto delle sequenze dell'intero genoma per procarioti ed eucarioti mostrano una diminuzione significativa delle prestazioni dei tool di AF testati. Nel complesso, i metodi basati su statistiche convenzionali hanno ottenuto risultati migliori rispetto agli approcci che utilizzano statistiche più complesse. È interessante notare che la distanza di Canberra di base implementata in alfpy è la misura di distanza più efficace nel riconoscere sequenze regolatorie funzionalmente correlate. Riassumento, solo tre strumenti (AFKS, FFP e mash) sono sufficientemente generici da poter essere applicati a tutti i 12 dataset di benchmarking; i restanti strumenti possono gestire solo sottoinsiemi dei dataset di riferimento, perché sono stati progettati solo per uno scopo specifico (ad esempio, per gestire solo alcuni tipi di sequenze, come nucleotidi, proteine e sequenze genomiche non assemblate o assemblate).

I risultati completi possono essere consultati al seguente sito afproject.org/app/tools/performance/

VIII. MASH

Mash estende la tecnica di riduzione delle dimensioni MinHash per includere una distanza di mutazione a coppie e un test di significatività del valore P, consentendo di raggruppare e cercare in modo efficiente collezioni di sequenze molto grandi. Mash riduce le sequenze e gli insiemi di sequenze di grandi dimensioni a piccoli sketch rappresentativi, dai quali è possibile stimare rapidamente le distanze di mutazione globali. Si tratta di un metodo estremamente veloce che utilizza la strategia MinHash bottom sketch per stimare l'indice di Jaccard dei multi-insiemi di k-mer di due sequenze in ingresso. In altre parole, stima il rapporto tra le corrispondenze di k-mer e il numero totale di k-mer delle sequenze. Utilizzando solo gli sketch, che possono essere migliaia di volte più piccoli, è possibile stimare rapidamente la somiglianza delle sequenze originali con un errore limitato. È importante notare che l'errore di questo calcolo dipende solo dalla dimensione dello sketch ed è indipendente dalla dimensione del genoma. La tecnica MinHash è una forma di locality-sensitive hashing (LSH) che serve a stimare velocemente quanto simili due insiemi sono. L'LSH è una tecnica di hashing che calcola l'hash di oggetti simili ponendoli nello stesso "bucket" con un'alta probabilità. Poiché gli elementi simili finiscono negli stessi bucket, questa tecnica può essere utilizzata per il clustering dei dati e la ricerca dei vicini più prossimi. Si differenzia dalle tecniche di hashing convenzionali in quanto le collisioni di hash sono massimizzate e non minimizzate. In alternativa, la tecnica può essere vista come un modo per ridurre la dimensionalità dei dati ad alta densità; gli elementi di input ad alta densità possono essere ridotti a versioni a bassa densità, preservando le distanze relative tra gli elementi.

A. Funzioni mash per il confronto

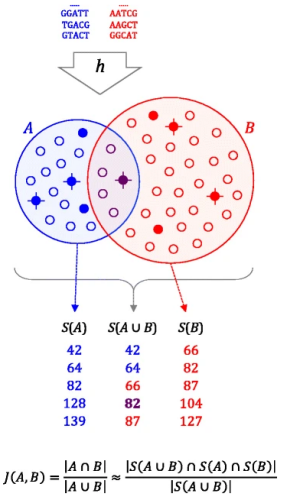
Mash fornisce due funzioni di base per il confronto delle sequenze: sketch e dist. La funzione sketch converte una sequenza o un insieme di sequenze in uno sketch MinHash. La funzione dist confronta due sketch e restituisce una stima dell'indice di Jaccard (cioè la frazione di k-meri condivisi), un valore P e la distanza Mash, che stima il tasso di mutazione delle sequenze secondo un semplice modello evolutivo. Poiché Mash si basa solo sul confronto di sottostringhe di lunghezza k, o k-mer, gli input possono essere interi genomi, metagenomi, sequenze nucleotidiche, sequenze di amminoacidi o letture di sequenziamento "raw".

B. jaccard index - minhash bottom sketch

L'indice Jaccard è un valore comunemente usato per mostrare la somiglianza tra due insiemi. Sia U un insieme ed A e B sottoinsiemi di U, l'indice Jaccard è definito come il rapporto tra il numero di elementi della loro intersezione e il numero di elementi della loro unione:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Il valore è 0 quando i due insiemi sono disgiunti, 1 quando sono uguali. Altrimenti varia tra 0 ed 1. Due insiemi sono simili quando l'indice Jaccard tende ad 1. L'obiettivo di minhash è calcolare J(A,B) velocemente, senza calcolare esplicitamente l'intersezione e l'unione.



In primo luogo, le sequenze di due insiemi di dati vengono scomposte nei loro k-mer costituenti e ogni k-mer viene passato attraverso una funzione di hash h per ottenere un hash a 32 o 64 bit, a seconda della dimensione del k-mer in input. Gli insiemi di hash risultanti, A e B, contengono ciascuno $|A|$ e $|B|$ hash distinti (cerchietti). L'indice di Jaccard è semplicemente la frazione di hash condivisi (viola) su tutti gli hash distinti in A e B. Questo può essere approssimato considerando un campione casuale molto più piccolo dall'unione di A e B. Gli sketch MinHash S(A) e S(B) di dimensione s = 5 sono mostrati per A e B, comprendendo i cinque valori hash più piccoli per ciascuno (cerchi pieni). Unendo S(A) e S(B) per recuperare i cinque valori hash più piccoli in assoluto per $A \cup B$ (cerchi con croce) si ottiene S($A \cup B$). Poiché S($A \cup B$) è un campione casuale di $A \cup B$, la frazione di elementi in S($A \cup B$) che sono condivisi sia da S(A) sia da S(B) è una stima imparziale di J(A,B)

C. mash sketch

Per poter essere confrontate con mash, le sequenze devono essere prima sketched, il che crea rappresentazioni molto ridotte. Questo avviene automaticamente se a mash dist vengono fornite sequenze raw. Tuttavia, se verranno eseguiti più confronti, è più efficiente creare prima gli sketch con mash sketch e fornirli a mash dist al posto delle sequenze raw.

Per costruire uno sketch MinHash, Mash determina innanzitutto l'insieme dei k-mer costituenti facendo scorrere una finestra di lunghezza k sulla sequenza. Mash supporta alfabeti arbitrari (ad esempio nucleotidi o amminoacidi) e sequenze assemblate e non assemblate. Viene calcolato l'hash di ogni k-mer in una sequenza, in modo tale da creare un identificatore pseudo-casuale. Ordinando questi identificatori (hash), un piccolo sottoinsieme dalla cima dell'elenco ordinato può rappresentare l'intera sequenza (si tratta di min-hash). Più un'altra sequenza è simile, più min-hash è probabile che condivida. Per una data dimensione dello sketch s, Mash restituisce gli s hash più piccoli prodotti da h su tutti i k-mer della sequenza. Per uno sketch di dimensioni s e un genoma di dimensioni n, uno sketch "bottom"può essere calcolato in modo efficiente in tempo O(n log s) mantenendo una lista ordinata di dimensioni s e aggiornando lo sketch corrente solo quando un nuovo hash è più piccolo dello sketch corrente.

1) k-mer size: Come in qualsiasi metodo basato sui k-mer, i k-mer più grandi forniranno maggiore specificità, mentre i k-mer più piccoli forniranno maggiore sensibilità. I genomi più grandi richiedono anche k-mer più grandi per evitare k-mer condivisi per caso. La dimensione del k-mer viene specificata con -k e i file di sketch devono avere la stessa dimensione del k-mer per essere confrontati con mash dist. Quando *mash sketch* viene eseguito, valuta automaticamente la dimensione del k-mer specificata rispetto alle dimensioni dei genomi in input, stimando la probabilità di una corrispondenza casuale come:

$$p = \frac{1}{\frac{(\sum)^k}{g} + 1}$$

dove g è la dimensione del genoma e $\sum$ è l'alfabeto (ACGT di default). L'alfabeto può essere specificato con -a che modifica anche la dimensione predefinita dei k-mer per riflettere le informazioni più dense. Se questa probabilità supera una soglia (specificata da -w; 0,01 per impostazione predefinita) per qualsiasi genoma in input, verrà fornito un avviso con la dimensione minima del k-mer necessaria per rientrare nella soglia. Per le collezioni di sketch di grandi dimensioni, anche la memoria e lo spazio di archiviazione possono essere considerati nella scelta della dimensione del k-mer. Mash utilizzerà hash a 32 bit, piuttosto che a 64 bit, se può includere l'intero spazio k-mer per l'alfabeto in uso. In questo modo si dimezza (all'incirca) la dimensione del file di sketch su disco e la memoria utilizzata quando viene caricato per mash dist. Il criterio per utilizzare un hash a 32 bit è:

$$(\overline{\sum})^k \leq 2^{32}$$

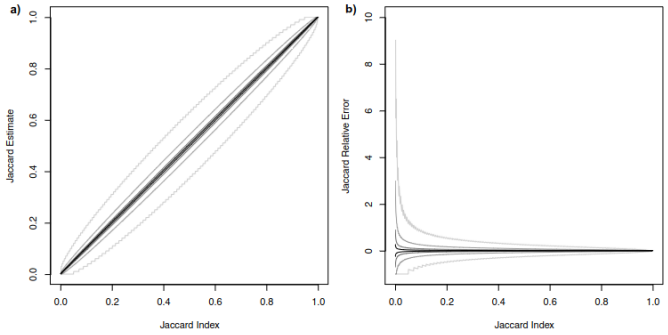
2) sketch size: La dimensione dello sketch corrisponde al numero di min-hash (non ridondanti) che vengono mantenuti. Sketch più grandi rappresenteranno meglio la sequenza, ma al costo di file di sketch più grandi e tempi di confronto più lunghi. L'errore limite di una stima della distanza per una data dimensione dello sketch s è formulato come:

$$\sqrt{\frac{1}{s}}$$

D. mash distance

Mash fa un merge-sort di due sketch bottom S(A) e S(B) per stimare l'indice di Jaccard. Il merge termina dopo che sono stati elaborati s hash unici (o entrambi gli sketch sono terminati) e la stima di Jaccard viene calcolata come per x hash condivisi trovati dopo aver elaborato gli hash di s. Poiché gli sketch sono memorizzati in maniera ordinata, questa operazione richiede solo O(s) tempo e viene effettivamente calcolato J(A,B). Il limite dell'errore della stima di Jaccard

$e = (1\sqrt{(s)})$ si basa solo sulla dimensione dello sketch ed è indipendente dalla dimensione del genoma.



Le dimensioni crescenti degli sketch sono progressivamente colorate da s=100 (grigio chiaro), s=1.000, s=10.000 e s=100.000 (nero). I limiti superiori e inferiori sono tracciati utilizzando la funzione di distribuzione binomiale inversecumulativa, con gli stessi parametri dell'equazione 8, in modo tale che per un dato indice di Jaccard vi sia una probabilità dello 0,99 che la corrispondente stima di Jaccard (a) o l'errore relativo (b) rientri nei limiti. Questi grafici illustrano che l'errore relativo può crescere notevolmente quando si stimano piccoli valori di Jaccard. Pertanto, è consigliabile utilizzare sketch di grandi dimensioni quando si confrontano sequenze divergenti con pochi k-mer condivisi. Questi grafici illustrano solo l'errore della stima di Jaccard e sono indipendenti dalla dimensione del k-mer. Tuttavia, l'errore relativo può diventare molto grande per valori di Jaccard molto piccoli (cioè genomi divergenti). In questi casi, per compensare, è necessario uno sketch di dimensioni maggiori o un k più piccolo. Mash può anche confrontare sketch di dimensioni diverse, ma tali confronti sono vincolati dal più piccolo dei due sketch s ¡ u e vengono considerati solo i valori più piccoli di s. La Mash distance D cerca di stimare direttamente il tasso di mutazione nell'ambito di un semplice processo di Poisson di mutazione casuale dei siti. Data la probabilità d di una singola sostituzione, il numero atteso di mutazioni in un k-mer è $\lambda = kd$. Pertanto, secondo un modello di Poisson (assumendo k-mer unici e mutazioni casuali e indipendenti), la probabilità che non si verifichi alcuna mutazione in un dato k-mer è e^{-kd}, con un valore atteso pari alla frazione di k-mer conservati w rispetto al numero totale di k-mer t nel genoma, w/t. Risolvendo l'equazione $e^{-kd} = \frac{w}{t}$ risulta $d = -(\frac{1}{k})ln\frac{w}{t}$. Mash imposta t sulla dimensione media del genoma n, penalizzando così le differenze di dimensione del genoma e misurando la somiglianza. Infine, poichè la stima di Jaccard j può essere inquadrata in termini di dimensione media del genoma $j = \frac{w}{(2n-w)}$, la frazione di k-meri condivisi può essere considerata in termini di indice di Jaccard $\frac{w}{n} = \frac{2j}{1+j}$, ottenendo la Mash distance:

$$D = -\frac{1}{k}ln\frac{2j}{1+j}$$

La figura 1 mostra i limiti di errore della Mash distance col variare della dimensione degli sketch. Questo grafico illustra che distanze di Mash più grandi richiedono sketch di grandi dimensioni per essere stimati con precisione. Tuttavia, con una dimensione dello sketch adeguatamente grande, è possibile stimare con precisione la distanza di Mash in un'ampia gamma di valori. La scelta di una dimensione k-mer più piccola può anche migliorare l'accuratezza per le sequenze divergenti, ma la scelta del k-mer dipende anche dalla dimensione del genoma.

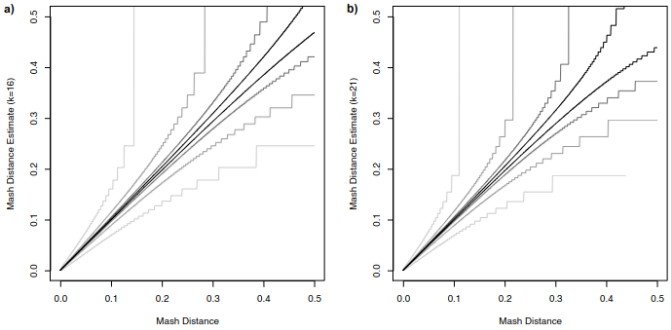


Fig. 1: limiti di errore della Mash distance col variare della dimensione degli sketch

1) Relazione tra dimensione k-mer e della dimensione del genoma sulla distanza di Mash: La figura 2 mostra (a) La relazione tra l'indice di Jaccard e la distanza di Mash per dimensioni di k-mer di 15 (rosso), 21 (nero) e 27 (blu). Per una distanza Mash fissa (ad esempio, 0,2), le dimensioni dei k-mer più grandi comportano valori di Jaccard più bassi perché un numero minore di k-mer lunghi è condiviso tra sequenze divergenti. Pertanto, può essere utile utilizzare una piccola dimensione di k-mer per evitare l'errore più elevato che si verifica con valori Jaccard piccoli. (b) illustra l'effetto di k-mer non unici e delle dimensioni del genoma e regola la distanza Mash prevista in base al numero di k-mer casuali che saranno condivise per caso tra due genomi da 1 Gbp (Gigabase pairs). In questo caso, l'asse delle ascisse mostra un ipotetico indice di Jaccard, ipotizzando che tutti i k-mer siano unici, mentre l'asse delle ordinate mostra la distanza

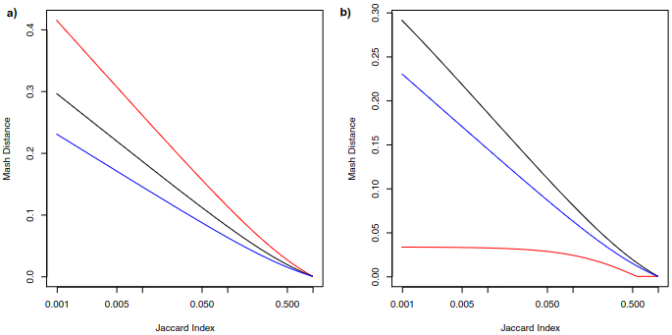


Fig. 2: relazione tra l'indice di Jaccard e la distanza di Mash per dimensioni di k-mer

Mash che tiene conto di tali collisioni. In generale, la scelta più piccola di k che elimina la maggior parte delle collisioni k-mer è la migliore, perché massimizza la sensibilità senza alterare la distanza Mash risultante.

E. p-value

Nel caso di genomi molto distanti tra loro, può essere difficile giudicare il significato di un determinato indice Jaccard o di una Mash distance. La probabilità che un dato k-mer K appaia in un genoma casuale X di lunghezza n è:

$$\mathrm{P(K \in X)} = 1 - (1 - |\Sigma|^{-k})^n$$

quindi per un k piccolo e un n grande la probabilità che appaia un k-mer casuale è alta. Il numero di k-mer corrispondendi in sketch di due genomi non correlati dipende dalla lunghezza dello sketch e dalla probabilità che un k-mer casuale appaia in un genoma, dove l'indice jaccard r tra due genomi casuali X e Y è dato da:

$$r = \frac{P(K \in X)P(K \in Y)}{P(K \in X) + P(K \in Y) - P(K \in X)P(K \in Y)}$$

per la dimensione dello sketch s, il massimo valore di hash nello sketch v e i bit di hash b, il numero di k-mer distinti nel genoma è stimato come n = 2^b s/v. Per la dimensione della popolazione m di tutti i k-mer distinti in X e Y e il numero di k-meri condivisi w, dove:

$$m = |X \cup Y| = |X| + |Y| - w$$

la probabilità p di osservare x o più corrispondenze tra gli sketch di questi due genomi può essere calcolata utilizzando la funzione di distribuzione cumulativa ipergeometrica. Per la dimensione dello sketch s, la dimensione condivisa w e la dimensione della popolazione m:

$$p(x; s; w; m) = 1 - \sum_{i=0}^{x-1} \frac{\binom{w}{i}\binom{m-w}{s-i}}{\binom{m}{s}}$$

Tuttavia, poiché m è tipicamente molto grande e la dimensione dello sketch è relativamente molto più piccola, è più pratico approssimare la distribuzione ipergeometrica con la distribuzione binomiale in cui il valore atteso di può essere calcolato utilizzando l'equazione mostrata in precedenza per il calcolo della r:

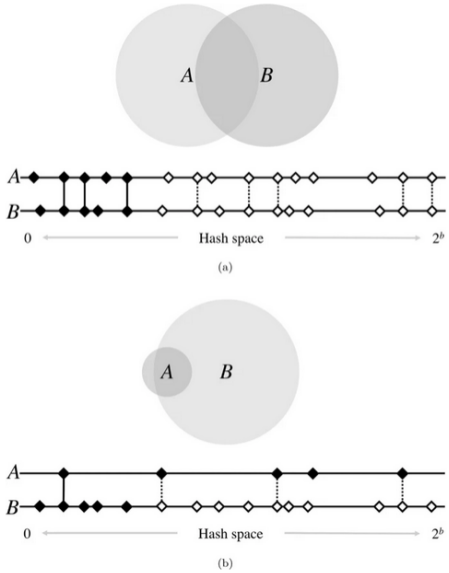
$$p(x; s; r) = 1 - \sum_{i=0}^{x-1} \binom{s}{i} r^i (1 - r)^{s-i}$$

Quindi riassumendo, poiché le distanze MinHash sono stime probabilistiche, è importante considerare la probabilità di vedere una data distanza per caso. *mash dist* fornisce quindi i valori p con le stime di distanza. I valori p più bassi corrispondono a stime di distanza più affidabili e spesso vengono arrotondati a 0 a causa dei limiti in virgola mobile. Se i valori di p sono elevati (ad esempio, superiori a 0,01), la dimensione del k-mer è probabilmente troppo piccola per le dimensioni dei genomi da confrontare.

F. mash screen

L'algoritmo MinHash si è dimostrato efficace per stimare rapidamente la somiglianza di due genomi o metagenomi. Tuttavia, questo metodo non è in grado di stimare in modo affidabile il contenimento di un genoma all'interno di un metagenoma. Questo perché le applicazioni tipiche di MinHash approssimano la somiglianza piuttosto che il contenimento. Mediante Mash Screen, è possibile misurare qualitativamente la rappresentazione di un genoma o di un proteoma all'interno di un metagenoma. Le applicazioni spaziano da un rapido screening della contaminazione alla ricerca di ceppi batterici e virali specifici in tutti i metagenomi disponibili.

1) perchè non si può usare indice Jaccard: abbiamo visto in precedenza che per due genomi a e b, la loro somiglianza è definita come l'indice j di Jaccard tra i loro insieme di k-mer A e B. Pertanto, la somiglianza è sensibile alle diverse lunghezze delle sequenze. Se si considera un singolo genoma a e un metagenoma molto più grande b, si può prevedere che la somiglianza tra a e b sia bassa, anche se b contiene interamente a, perché b conterrà probabilmente molti k-mer non presenti in a, rendendo l'unione di A e B più grande dell'intersezione.



Nell'immagine che segue, in a, i genomi di dimensioni simili sono adatti alla stima della somiglianza, poiché i loro hash sono distribuiti in modo simile nello spazio degli hash. Tuttavia, se i genomi sono di dimensioni molto diverse, come in b, il genoma più grande saturerà lo spazio in modo più denso. Ciò fa sì che una frazione più elevata di hash corrispondenti sia contenuta solo nello sketch dell'insieme più piccolo, sottostimando il contenimento di A in B. Pertanto, tutti gli hash di B devono essere considerati per stimare accuratamente il contenimento di A. La somiglianza varia da 0 a 1, è simmetrica e 1-j è una metrica.

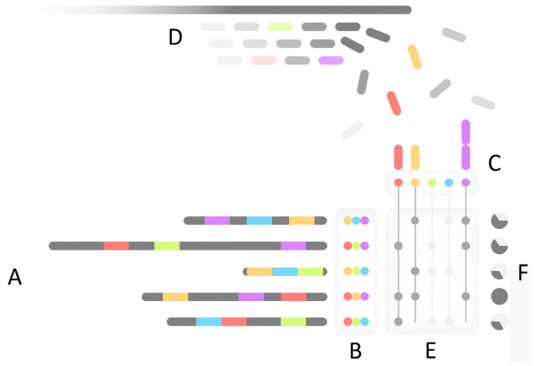
Al contrario, l'indice di contenimento c è asimmetrico e misura quanto un genoma è contenuto in un altro:

$$c_k(a, b) = \frac{|A \cap B|}{|A|}$$

Se tutti i k-mer di A sono contenuti in B, a prescindere da quanti altri k-mer contenga B, l'indice di contenimento sarà uguale a 1. Per stimare la somiglianza, Mash utilizza una strategia di "bottom sketch", proposta originariamente da Broder nel 1997. Da allora sono emerse tecniche più efficienti per la stima della somiglianza, ma il bottom sketching è elegante nella sua semplicità. In breve, tutti i k-mer di un genoma A vengono passati attraverso una singola funzione di hash h, ma solo i valori di hash più piccoli di m vengono memorizzati come sketch S(A), dove $|S(A)| \ll |A|$. Poiché la probabilità che S(A) e S(B) condividano un valore minimo è legata all'indice di Jaccard di A e B, la somiglianza degli insiemi completi di k-mer A e B può essere rapidamente approssimata dagli sketch S(A) e S(B), relativamente più piccoli. Mash converte ulteriormente la somiglianza nella distanza Mash, che è una stima della distanza mutazionale tra le due sequenze. A differenza della somiglianza, che può essere stimata in modo affidabile utilizzando sketch di dimensioni fisse, la stima del contenimento richiede una dimensione dello sketch proporzionale alla dimensione del genoma. I metodi precedenti di stima del contenimento hanno cercato di comprimere e/o indicizzare una grande collezione di reads, con l'obiettivo di consentire una rapida ricerca di sequenze tra di esse (ad esempio, dato un database di riferimento di trascritti, isolati o metagenomi, identificare tutti quelli che contengono una sequenza di query). Si tratta di un problema importante, ma nel caso di mash screen si considera l'opposto: dato un database di riferimento di genomi, identificare tutti quelli contenuti in un metagenoma di query. Lo screening stima anche, senza assemblaggio, la somiglianza tra i genomi di riferimento e quelli contenuti nel metagenoma.

2) mash screen funzionamento: A differenza di *dist*, screen deve ricevere un file di sketch multiplo pre-elaborato con cui effettuare lo screening delle letture. Questo file di sketch può essere creato con il comando sketch nello stesso modo di dist. Mediante *mash screen*, viene testato rapidamente un database di molti genomi per verificarne il contenimento all'interno di un metagenoma più grande e non assemblato. Per ogni genoma di riferimento, Mash Screen calcola un punteggio di contenimento che misura la somiglianza del genoma di riferimento con una sequenza contenuta nel metagenoma. Ciò consente di svolgere attività di routine, come lo screening rapido della contaminazione, la selezione di genomi di riferimento appropriati per le analisi basate sulla mappatura e lo screening efficiente di interi database metagenomici per rintracciare singole specie e ceppi in tutti i campioni.

(A) Il minimo di m hash (in questo caso 3, colorati) per ogni sequenza di riferimento viene determinato durante il processo di sketching per produrre (B) una libreria di sketch MinHash di riferimento. Per lo screening, gli hash distinti di tutti gli sketch di riferimento vengono raccolti e utilizzati come chiavi per (C) una mappa di conteggi osservati per hash, che viene popolata da (D) hashing k-mer dalla miscela di sequenze durante il flusso. (E) I conteggi della mappa vengono interrogati per ogni sketch per produrre (F) una stima del contenimento per ogni costituente della miscela.



3) streaming: Poiché la miscela è destinata a essere un insieme arbitrariamente grande non è auspicabile caricare l'intero insieme in memoria. Al contrario, gli sketch di riferimento sono piccoli e possono essere facilmente indicizzati in memoria. Per calcolare il contenimento di ogni sketch di riferimento, la miscela viene eseguita in streaming rispetto a questo indice. In primo luogo, tutti gli hash distinti presenti negli sketch costituenti vengono raccolti in un insieme di riferimento. Questo insieme viene utilizzato come chiave di una mappa (implementata come Robin-Hood hashtable nella versione 2.3 di Mash), dove i valori associati sono i conteggi di quante volte l'hash compare nella miscela, inizialmente impostati a zero. Le sequenze della miscela vengono quindi inviate in streaming, tradotte in sei fotogrammi, se applicabile, e tutti i loro k-mer vengono sottoposti ad hashing, incrementando i contatori nella mappa se è presente una chiave per quell'hash. Infine, si determina una distanza per ogni costituente, contando quanti hash del suo sketch hanno conteggi non nulli nella mappa. Si noti che il numero di voci in questa mappa, e quindi la memoria utilizzata, dipenderà dal numero di hash distinti nell'insieme completo dei possibili costituenti. Poiché molti genomi di riferimento saranno molto simili tra loro, la memorizzazione di uno sketch MinHash per ogni riferimento è più efficiente della memorizzazione di una selezione casuale di k-mer.

G. esecuzione tool

parametri di default: k-mer size= 21 (k può variare da 1 a 32), sketch size = 1000. Ogni sketch richiede 8kB. Un k piccolo migliora anche la sensibilità, che è utile quando si confrontano dati rumorosi come quelli del sequenziamento di una singola molecola. Le esecuzioni del tool sono avvenute su macchina GNU/Linux avente come CPU AMD Ryzen 5 5600H e 16GB di RAM.

1) Stima della distanza: Si confrontano due genomi di Escherichia coli:

$> gi|49175990|ref|NC_{0}00913.2|Escherichiacolistr.K-12substr.MG1655$

$> gi|47118301|dbj|BA000007.2|EscherichiacoliO157 : H7str.SakaiDNA$

Si esegue *mash dist*:

```
./mash dist genome1.fna genome2.fna
```

Output:

```
genome1.fna genome2.fna 0.0222766 0 456/1000
```

genome1.fna è il reference-ID, genome2.fna il query-ID, il primo valore numerico rappresenta la mash distance, quello successivo la P-value ed infine il numero hash corrispondenti. Tempo di esecuzione (media aritmetica su 10 run del tool):

real	0,2s
user	0,182s
sys	0,017s

- **Real** è il tempo trascorso dall'inizio alla fine della chiamata. È tutto il tempo trascorso, compresi lassi di tempo utilizzati da altri processi e il tempo che il processo trascorre bloccato.
- **User** è la quantità di tempo CPU speso nel codice in modalità utente (al di fuori del kernel) all'interno del processo. Si tratta solo del tempo effettivo di CPU utilizzato per l'esecuzione del processo.
- **Sys** è la quantità di tempo CPU spesa nel kernel all'interno del processo. Ciò significa l'esecuzione del tempo di CPU speso nelle chiamate di sistema all'interno del kernel, in contrapposizione al codice di libreria, che è ancora in esecuzione nello spazio utente. Come per 'user', si tratta solo del tempo di CPU utilizzato dal processo.

2) Confronti a coppie con file di sketch composto: Si utilizza un altro genoma di E.coli:
$> gi|682117612|gb|CP009273.1|EscherichiacoliBW25113$
Si effettua prima lo sketch dei due genomi in modo tale da creare uno sketch combinato:

```
./mash sketch -o reference
genome1.gna genome2.fna
```

Poi si stima la distanza da ciascuna query con il terzo genoma:

```
./mash dist reference.msh genome3.fna
```

Output:

```
genome1.fna genome3.fna 0 0 1000/1000
genome2.fna genome3.fna 0.0222766 0 456/1000
```

3) Esame di un read set per il contenimento dei genomi RefSeq: Se un read set contiene potenzialmente più genomi, può essere "esaminato" rispetto ad un database per stimare quanto ciascun genoma sia contenuto nel read set. È possibile utilizzare *sra-tools* per scaricare il dataset ERR024951. Dopo aver installato il tool, è possibile scaricare il dataset utilizzando *prefetch*, in tal caso il file avrà l'estensione .sra. È necessario convertire il file in .fastq mediante *sff-dump*. Altrimenti, il tool permette di scaricare il dataset direttamente convertito:

```
fastq-dump ERR024951
```

Successivamente, possiamo analizzare il dataset scelto andando a rimuovere la ridondanza con -w con i genomi RefSeq e salvare l'output in un file apposito:

```
./mash screen -w -p 12 refseq.genomes.k21s1000.msh
ERR024951.fastq > screen.tab
```

Il flag -p indica il numero di thread da impiegare nell'esecuzione. Infine, possiamo leggere il file in ordine decrescente di identità:

```
sort -gr screen.tab | head
```

Quindi, i risultati dello screening:

identity	shared-hashes	median-multiplicity	p-value	query-ID
0.99957	991/1000	24	0	GCF_002054545.1_ASM205454v1_genomic.fna.gz
0.99899	979/1000	26	0	GCF_000841985.1_ViralProj14228_genomic.fna.gz
0.998844	976/1000	101	0	GCF_900086185.1_12082_4_85_genomic.fna.gz
0.923964	190/1000	49	0	GCF_000900935.1_ViralProj181984_genomic.fna.gz
0.900615	111/1000	100	0	GCF_001876675.1_ASM187667v1_genomic.fna.gz
0.887722	82/1000	31	3.16322e-233	GCF_001470135.1_ViralProj306294_genomic.fna.gz
0.873204	58/1000	22	1.8212e-156	GCF_000913735.1_ViralProj227000_genomic.fna.gz
0.868675	52/1000	57	6.26251e-138	GCF_001744215.1_ViralProj344312_genomic.fna.gz
0.862715	45/1000	1	1.05185e-116	GCF_001882095.1_ViralProj353688_genomic.fna.gz
0.856856	39/1000	21	6.70643e-99	GCF_000841165.1_ViralProj14230_genomic.fna.gz

real	33s
user	1m55s
sys	4,6s

REFERENCES

[1] Zielezinski, A., Girgis, H.Z., Bernard, G. et al. Benchmarking of alignment-free sequence comparison methods. Genome Biol 20, 144 (2019). https://doi.org/10.1186/s13059-019-1755-7

[2] Zielezinski, A., Vinga, S., Almeida, J. et al. Alignment-free sequence comparison: benefits, applications, and tools. Genome Biol 18, 186 (2017). https://doi.org/10.1186/s13059-017-1319-7

[3] Ondov, B.D., Treangen, T.J., Melsted, P. et al. Mash: fast genome and metagenome distance estimation using MinHash. Genome Biol 17, 132 (2016). https://doi.org/10.1186/s13059-016-0997-x

[4] Ondov, B., Starrett, G., Sappington, A. et al. Mash Screen: high-throughput sequence containment estimation for genome discovery. Genome Biol 20, 232 (2019). https://doi.org/10.1186/s13059-019-1841-x