DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Algoritmi per la Bioinformatica 2019/2020

Progetto 8

Alignment-free and mapping-free frameworks to detect variants

Barisan Anna, Milia Mikele

Introduzione: il problema studiato

Problema: ricerca di mutazioni (SNP) all'interno del genoma, utilizzando metodi che non usano l'allineamento di sequenze (alignment-free/mapping-free).

Genotipizzazione: processo di definizione delle differenze nel corredo genetico o nel genotipo di un individuo tramite l'esame della sequenza individuale del suo DNA.

Introduzione: le mutazioni

Le differenze tra i genomi di diversi individui di una determinata specie sono note come **varianti genomiche**.

- SNP: Single Nucleotide Polymorphisms, variazione rispetto ad un unico nucleotide;
- SNV: Single Nucleotide Variant;
- Indel: inserzioni o eliminazioni di una o più basi consecutive;
- Varianti *de novo*: alterazioni presenti per la prima volta in un membro della famiglia (figlio) e non nei genitori.

Introduzione: le mutazioni

Un **allele** è una delle possibili versioni di una variante.

Il **genotipo** di una variante è dato dalla coppia di alleli della variante presenti nei due aplotipi (alleli ereditati dai due genitori):

- genotipo omozigote se i due alleli sono identici
- genotipo eterozigote se i due alleli differiscono.

Genotipizzare è il compito di calcolare il genotipo di tutte le varianti all'interno del campione in input.

INPUT: read brevi (sequenze di nucleotidi) prodotte da tecnologia NGS.

Introduzione: la genotipizzazione

La **pipeline standard** utilizzata per la chiamata delle varianti include l'**allineamento** delle read con una sequenza del genoma di riferimento: identifica la posizione più probabile lungo il genoma di riferimento da cui proviene ciascuna read e assegna i genotipi alle varianti nelle read.

↳ computazionalmente costosi, molto tempo, difficoltà non risolte.

I metodi **alignment-free** effettuano la genotipizzazione senza utilizzare l'allineamento.

Metodi *alignment-free*

I metodi **alignment-free** effettuano la genotipizzazione senza utilizzare l'allineamento: sfruttano diversi modelli, **algoritmi**, e **strutture dati efficienti**

- tempo di esecuzione ridotto
- accuratezza comparabile a tool alignment-based.

Possibile effettuare una distinzione all'interno dei tool alignment-free:

Reference-based	Reference-free o de novo
utilizzano genoma di riferimento e lista preassegnata di varianti note in input oltre alle read;	non utilizzano la reference.

Outline

- Introduzione del problema
- Pipeline standard (allineamento)
- Metodi alignment-free
  - **VarGeno**
  - **MALVA**
  - FastGT
  - COBASI
  - **Kevlar**
  - **DiscoSNP++**
- Benchmark
- Conclusioni

VarGeno (Sun & Medvedev, 2018)

Idea: la corrispondenza approssimativa di *k*-mer di medie dimensioni riesce a identificare univocamente i loci nel genoma senza allineamento completo delle read.

Traits:

- Alignment-free
- Reference-based (input: sample di read, genoma di riferimento, lista di SNP)
- *k*-mer count (k = 32)
- diretto miglioramento di LAVA (Shajii et al., 2016)

Goal: genotipizzare e rilevare principalmente SNP (no indel).

VarGeno: il Bloom Filter

Bloom Filter: struttura dati probabilistica efficiente in termini di spazio che rappresenta un insieme di elementi e consente di effettuare query di appartenenza approssimative, per determinare se un elemento appartiene o no all'insieme dei dati del Bloom filter.

- verificare se un elemento è "possibilmente nel set" (possono esserci falsi positivi, casi in cui si pensa erroneamente che l'elemento appartenga);
- verificare se un elemento è "sicuramente non nel set" (mai falsi negativi).

VarGeno: la struttura dati

Viene creato un **dizionario** contenenti tuple *<k-mer, puntatore-dati-associati>* ordinate in ordine crescente rispetto al valore intero dei *k*-mer codificati, con i *k*-mer della lista di SNP in input.

Hash table che mappa ogni numero intero senza segno di r bit *u* alla prima posizione in D in cui vi è un *k*-mer codificato i cui bit superiori sono maggiori o uguali a *u*.

Bloom filter B che contiene, per ogni i *k*-mer presente, un elemento corrispondente ai $(2k - r)$ bit inferiori.

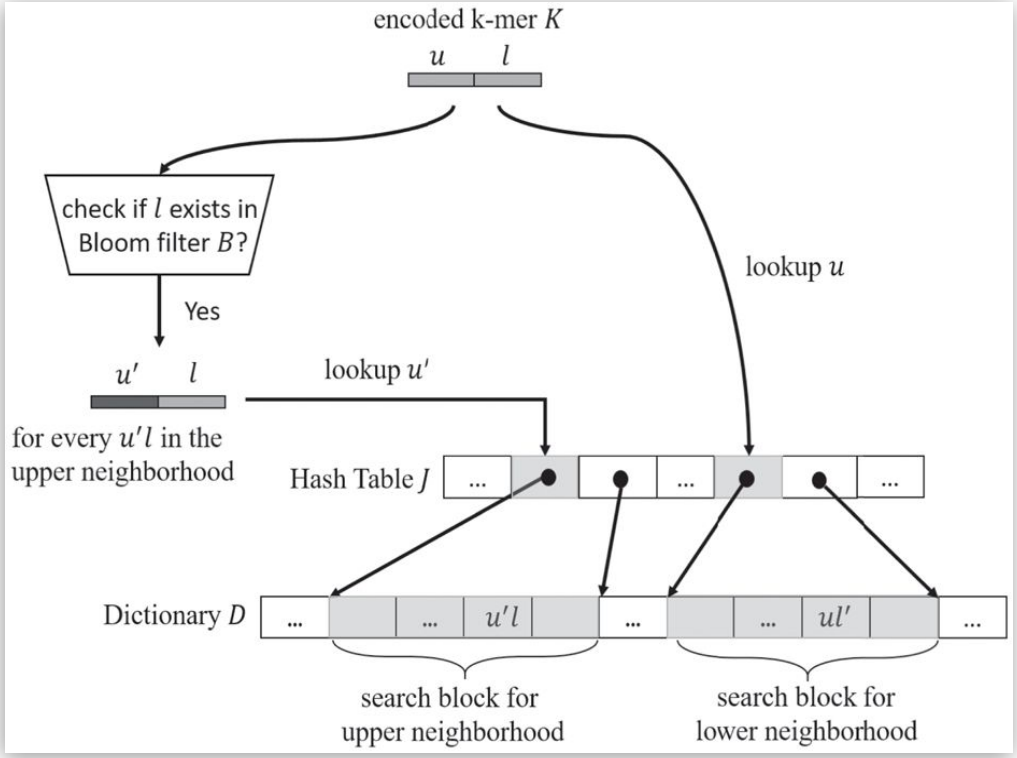
VarGeno: l'algoritmo

Vengono creati due indici, uno per tutti i *k*-mer presenti nella sequenza di riferimento e l'altro con *k*-mer da posizioni che si sovrappongono agli SNP della lista, con l'allele di riferimento sostituito da un allele alternato.

Partendo dagli indici, si **suddivide ciascuna read in *k*-mer non sovrapposti** e si interrogano gli indici, controllando la presenza del *k*-mer e dei suoi vicini a distanza di Hamming 1.

Vengono esplorati solo i vicini che differiscono in una posizione il cui punteggio di qualità è inferiore a una soglia c.

VarGeno: le query dei *k*-mer



Step 1

Query vicinato superiore.

Step 2

Query vicinato inferiore.

VarGeno: l'algoritmo

Effettuate le ricerche dei *k*-mer da una read, si **determina la singola posizione di mappatura per la read**: la posizione deve avere il maggior numero di corrispondenze, almeno due *k*-mer che corrispondono devono provenire da posizioni diverse e almeno un *k*-mer deve essere non modificato.

Decisa la posizione di corrispondenza migliore della read sul genoma di riferimento, la read viene utilizzata per **conteggiare** l'**allele** di riferimento o l'allele alternato degli SNP all'interno della posizione corrispondente.

Un **modello probabilistico**, basato sul teorema di Bayes, utilizza i conteggi per determinare il genotipo più probabile per ciascun SNP.

MALVA (Bernardini et al., 2019)

Idea: la signature (insieme di *k*-mer) di un allele modella con efficienza e permette di rilevare indel e varianti.

Traits:

- Alignment-free
- Reference-based
- *k*-mer count (**signature**)

Goal: genotipizzare e individuare, oltre a SNP bi-allelici, anche SNP multi-allelici e indel, sia brevi che lunghi.

MALVA: le signature e la struttura dati

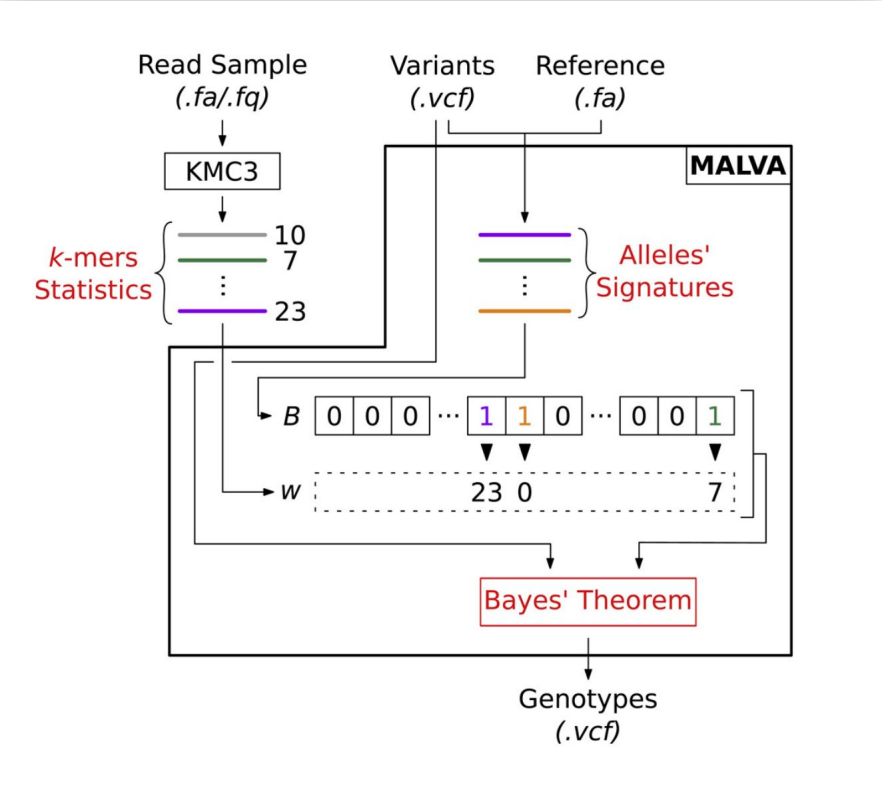
La firma dell'allele *a* di una variante *v* è il ***k*-mer centrato in *a*** in qualche genoma *g* che include *a*. Se sono note altre varianti a meno di *k* basi di distanza dall'allele, esso potrebbe avere **più** firme.

Se la stringa di basi che rappresenta a è più lunga di k la sua firma è l'insieme delle sue sottostringhe di lunghezza k.

Malva sfrutta 3 set per memorizzare le signature:

- REFSIG (contiene le firme di alleli di riferimento)
- ALTSIG (contiene firme di alleli alternati)
- REPCTX (contesto attorno a firme di alleli alternati che compaiono anche in altre regioni del genoma)

MALVA: l'algoritmo



4 Step

1. Calcolo delle firme.
2. Rilevamento delle firme ripetute.
3. Calcolo dei pesi delle firme degli alleli.
4. Chiamata dei genotipi.

MALVA: l'algoritmo

1. Calcolo delle firme.

Calcolo dei set REFSIG e ALTSIG che contengono le firme (*k*-mer) degli alleli di riferimento e alternati (controllo alleli distanti meno di *k*/2).

2. Rilevamento delle firme ripetute.

Controllo che le firme di alleli alternati non appaiano anche in altre posizioni nel genoma di riferimento: se si, tali firme vengono "ampliate", comprendendo il contesto intorno e inserite nel terzo set REPCTX.

MALVA: l'algoritmo

3. Calcolo dei pesi delle firme degli alleli.

Vengono estratti i *k*-mer delle read ed effettuato il conteggio delle loro occorrenze. Viene aumentato il peso dell'allele di riferimento ogni volta che un *k*-mer apparte nel set REFSIG e quello dell'allele alternato quando il *k*-mer appare in ALTSIG ma non in REPCTX.

4. Chiamata dei genotipi.

Dati i pesi calcolati, viene calcolata la probabilità a priori di tutti i possibili genotipi (per i possibili alleli): teorema di Bayes.
Output: il genotipo predetto è quello che ha la maggiore probabilità.

Kevlar (Standage et al., 2019)

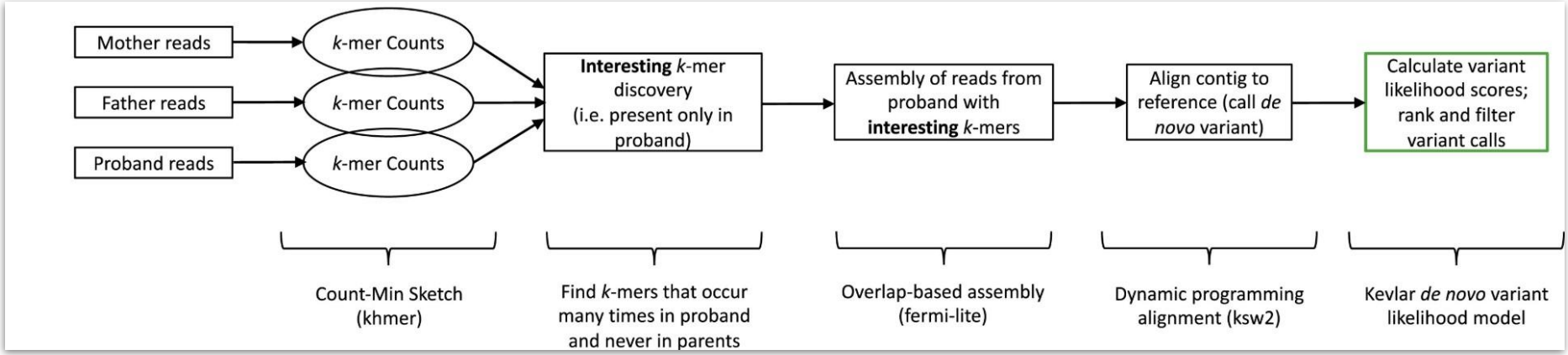
Idea: una mutazione cellulare all'interno di un trio genitore-figlio, dovrebbe comportare una nuova sequenza nel figlio rispetto ai genitori.

Traits:

- mapping-free
- trio genitore-figlio
- *k*-mer count

Goal: rilevare simultaneamente SNV *de novo* e indels

Kevlar: l'algoritmo



5 step

1. calcolo della frequenza dei *k*-mer
2. identificazione *k*-mer interessanti
3. assemblaggio dei *k*-mer interessanti
4. allineamento contig
5. chiamata delle varianti

Kevlar: l'algoritmo

1. **Calcolo della frequenza dei *k*-mer**

   Un **conteggio approssimativo** dei *k*-mer viene memorizzato all'interno di Count-Min sketch, struttura dati che favorisce l'efficienza alla precisione.

   La precisione di CM sketch, dipende dalla dimensione e dal numero degli elementi distinti che vengono tracciati.

   Se i *k*-mer sono presenti nei genomi di riferimento (genitori) e/o in un genoma di contaminanti (batteri, virus, ...) sono ignorati.

Kevlar: l'algoritmo

2. Identificazione *k*-mer interessanti

Ogni read del figlio è scansionata e per ciascun *k*-mer viene richiesta la sua frequenza alle CM sketch generate.

Se un *k*-mer è frequente nel genoma figlio ed è assente dai genomi genitori è identificato come "interessante".

Kevlar: l'algoritmo

3. Assemblaggio dei *k*-mer interessanti

Ogni read che contiene *k*-mer interessanti viene filtrata prima di qualunque altra analisi. Per i seguenti motivi:

- permettere il ricalcolo esatto delle frequenze di ogni *k*-mer interessante (scartandolo se non soddisfa più la frequenza di soglia).

- opportunità di scartare *k*-mer presenti nel genoma di riferimento e contaminanti che non sono stati ignorati.

Kevlar: l'algoritmo

3. Assemblaggio dei *k*-mer interessanti

Read interessanti che condividono numerosi k-mer sono raggruppate in insiemi disgiunti, ognuno rappresentante una mutazione.

Come? Viene definito un grafo di G nel seguente modo:

- ogni **nodo** identifica una **read** contenente uno o più *k*-mer interessanti
- una coppia di nodi è connessa da un **arco se** le rispettive **read hanno** uno o più **k-mer interessanti in comune**

Se due read condividono un *k*-mer interessante, allora sono parte della stessa componente connessa p di G.

Kevlar: l'algoritmo

3. Assemblaggio dei *k*-mer interessanti

Per ogni p ∈ G vengono assemblate la read corrispondenti tramite un algoritmo basato su **overlap** e viene prodotto il contig adatto per effettuare chiamate delle varianti.

Kevlar: l'algoritmo

4. **Allineamento contig**

   Dai genomi dei genitori vengono selezionate delle sequenze obiettivo di riferimento per i contig.

   Ogni contig viene allineato a ciascuna sequenza, e vengono mantenuti solo gli allineamenti con il punteggio più alto, classificandoli SNV o indel tramite due pattern.

   Qualsiasi allineamento che non corrisponde ai pattern, è identificato come **no-call**.

Kevlar: l'algoritmo

5. Chiamata delle varianti

Per assegnare un punteggio alle mutazioni *de novo*, Kevlar usa un modello probabilistico che considera la frequenza dei *k*-mer interessanti per calcolare la probabilità che siano *de novo*, ereditati o semplicemente dei falsi positivi.

La classificazione delle varianti predette è basata su un'euristica.

DiscoSNP++ (Peterlongo et al., 2017)

Idea: utilizzare grafi di *de Bruijn* probabilistici per migliorare l'identificazione e la categorizzazione delle varianti

Traits:

- hybrid
- *grafi di de Bruijn*

Goal: individuare e classificare, tutte le tipologie di SNP, indel compresi

DiscoSNP++: la struttura dati

Minia (Chikhi & Rizk, 2013) permette di costruire grafi di de Bruijn probabilistici, inserendo i nodi all'interno di Bloom Filter posti in cascata e deducendo implicitamente gli archi dalle interrogazione senza la necessità di memorizzarli.

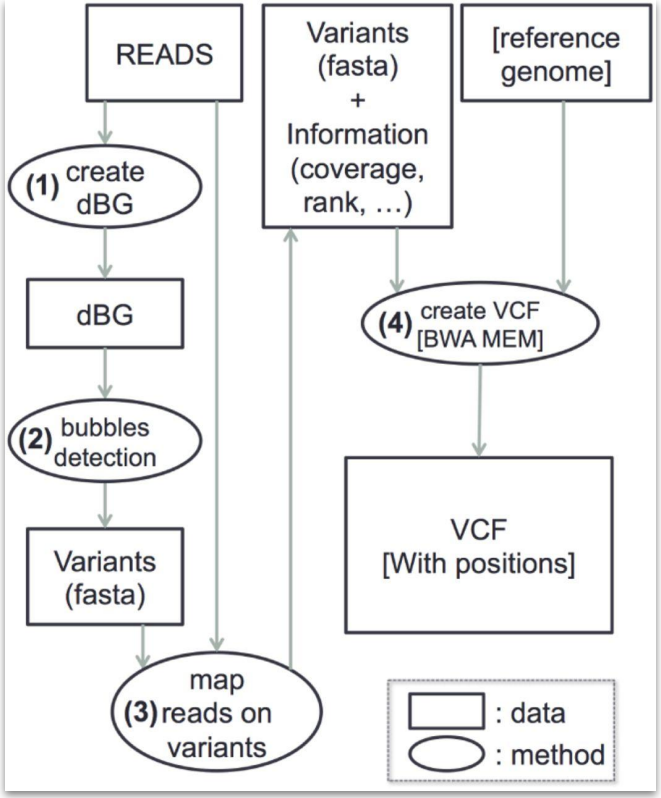
Problema: Un grafo di de Bruijn probabilistico è un'approssimazione eccessiva del grafo originale. Interrogando il Bloom Filter sull'esistenza o meno di un arbitrario nodo del grafo, si può ottenere un elemento falso positivo.

DiscoSNP++: la struttura dati

Soluzione: individuare e memorizzare i *k*-mer falsi positivi per evitare false diramazioni all'interno di una struttura separata, cFP (critical False Positive)

Modificare ogni interrogazione fatta al Bloom Filter in modo tale che produca **true** se e solo se il Bloom Filter risponde **true** e l'elemento su cui si sta interrogando non è presente all'interno dell'insieme cFP.

DiscoSNP++: l'algoritmo



4 Step

1. Creazione del grafo di *de Bruijn* probabilistico
2. Individuazione delle bolle
3. Chiamata dei genotipi
4. Mapping su genoma di riferimento

DiscoSNP++: l'algoritmo

1. **Creazione del grafo di *de Bruijn* probabilistico**

   Tramite Minia viene costruito il grafo di *de Bruijn* probabilistico.

   A causa degli errori di sequenziamento prodotti dagli NGS, tutti i *k*-mer che hanno un numero di occorrenze non superiore ad una threshold vengono scartati.

DiscoSNP++: l'algoritmo

2. Individuazione delle bolle

Individua e classifica le bolle generate dalla presenza di SNP (vicini o meno) e/o indel all'interno del grafo tramite due sottomoduli:

- il primo attraversa tutti i *k*-mer con nodi di diramazione destri e li propone come potenziali nodi di partenza per bolle che identificano SNP o indel.
- il secondo ricorsivamente espande congiuntamente due cammini e controlla che i vincoli imposti sui parametri di diramazione siano rispettati

DiscoSNP++: l'algoritmo

3. Chiamata dei genotipi

Le read iniziali vengono mappate sulle sequenze delle mutazioni trovate, e il rank viene calcolato basandosi sulla loro frequenza.

I genotipi sono individuati in modo indipendente per ogni organismo, tramite un modello binomiale basato sulla probabilità che una read venga mappata erroneamente su un allele.

DiscoSNP++: l'algoritmo

4. **Mapping su genoma di riferimento**

   Se viene fornito un genoma di riferimento, utilizzando BWA-mem le mutazioni predette vengono mappate sul genoma di riferimento.

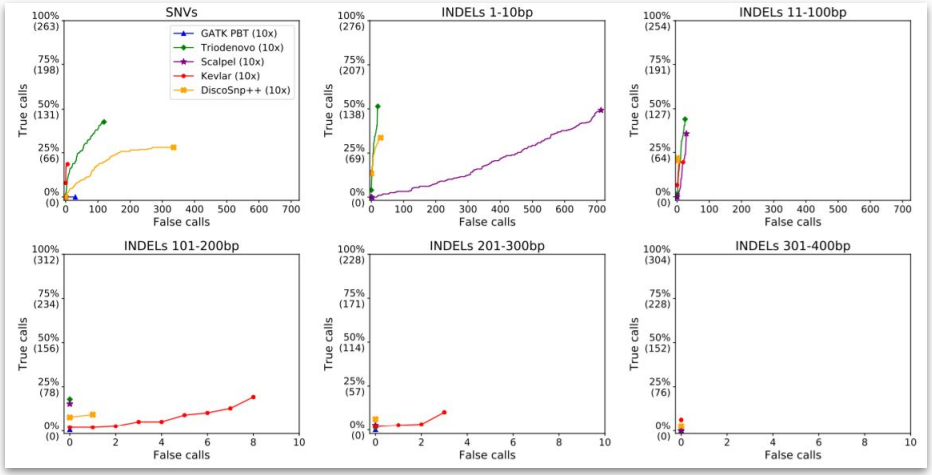
   Se una variante ha più di una posizione ottima di mappatura, ne viene scelta una una random.

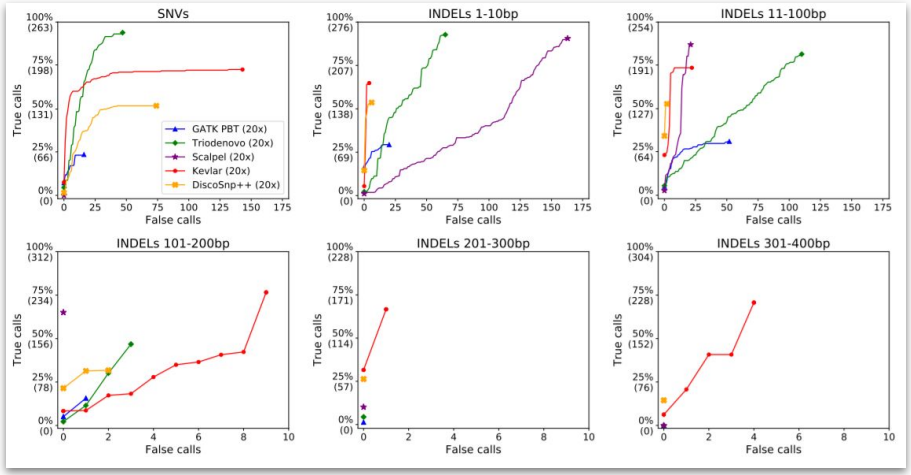
Benchmark

Confronti:

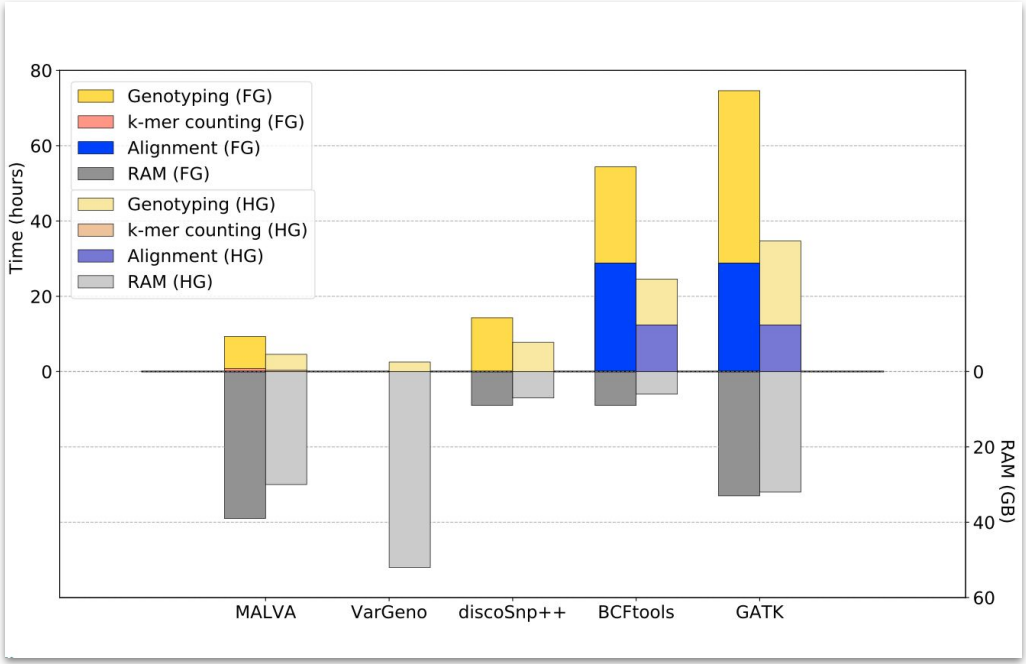
- Kevlar vs DiscoSNP++
- Malva vs VarGeno vs DiscoSNP++

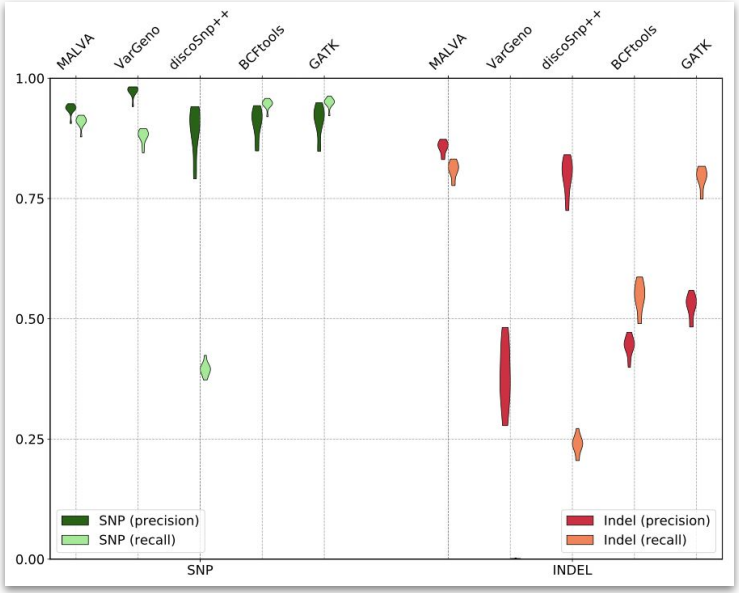
Benchmark: Kevlar vs DiscoSNP++





Benchmark: MALVA vs VarGeno vs DiscoSNP++





Conclusioni

I metodi **alignment-free** effettuano la genotipizzazione propongono:

- una soluzione efficiente a problemi dei tool alignment-based
- tempo di esecuzione ridotto
- accuratezza comparabile ai tool alignment-based
- modelli e strutture dati efficienti
- modello probabilistico per la chiamata delle varianti

Sviluppi futuri

Considerazione: allo stato attuale delle cose non c'è un tool alignment-free che prevale sull'altro.

Chiaramente ognuno dei tool analizzati risolve un determinato task e ognuno può essere migliorato per coprire task più generali e/o essere più performante.

Come?

- ottimizzando le strutture dati utilizzate
- variando il modello probabilistico utilizzato durante la fase di genotipizzazione

Grazie per l’attenzione.