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

Allineamento wavefront per l'individuazione di ricombinazioni

VOLPATO MATTIA

MATRICOLA: 866316

RELATORE: GIANLUCA DELLA VEDOVA

CO-RELATORE: PAOLA BONIZZONI

ANNO ACCADEMICO 2022-2023

Panoramica del lavoro

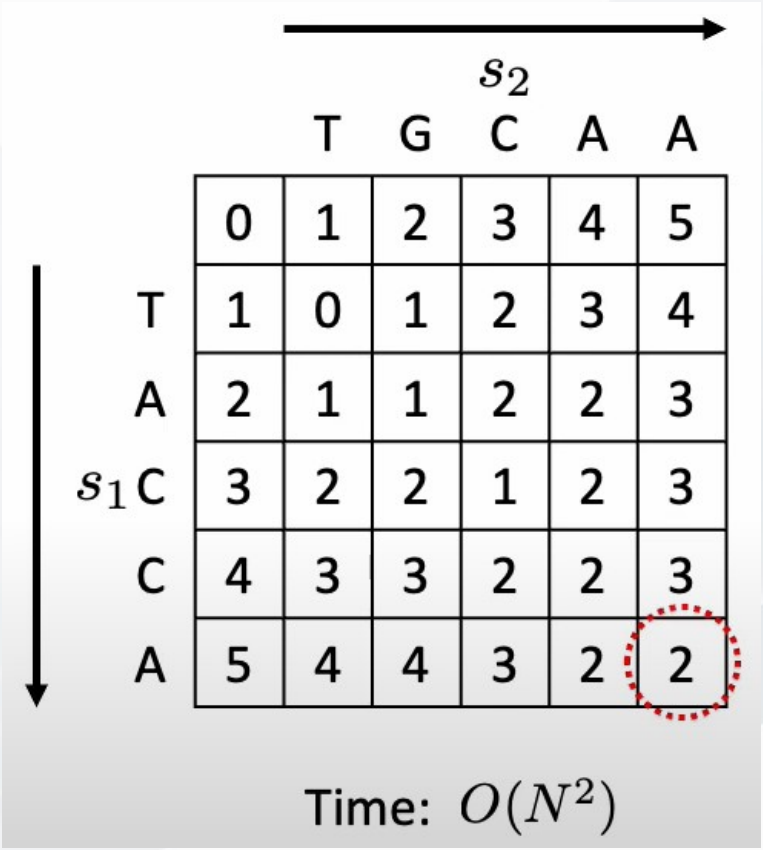
I. **Allineamento di sequenze**
    i. Allineamento con **grafi di variazione**
    ii. Allineamento con **ricombinazione**

II. **Algoritmo *wavefront* per il calcolo della distanza di edit**
    i. Estensione a **grafi di variazione**

III. **Implementazione e sperimentazione**
    i. Prototipo per migliorare le prestazioni di ***RecGraph***
    ii. Confronto sperimentale con ***RecGraph***

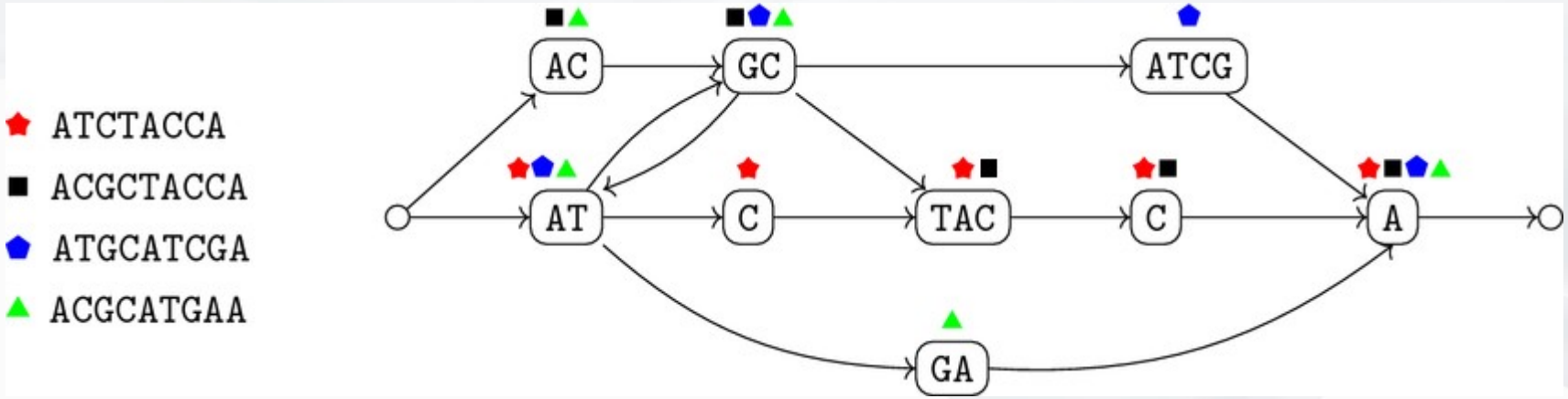
Distanza di edit

- Misurare la **similarità** di due sequenze
- **Programmazione dinamica**
- Generalizzabile a **strutture** a **grafo**?



POA e grafi di variazione

- **Grafi di variazione**



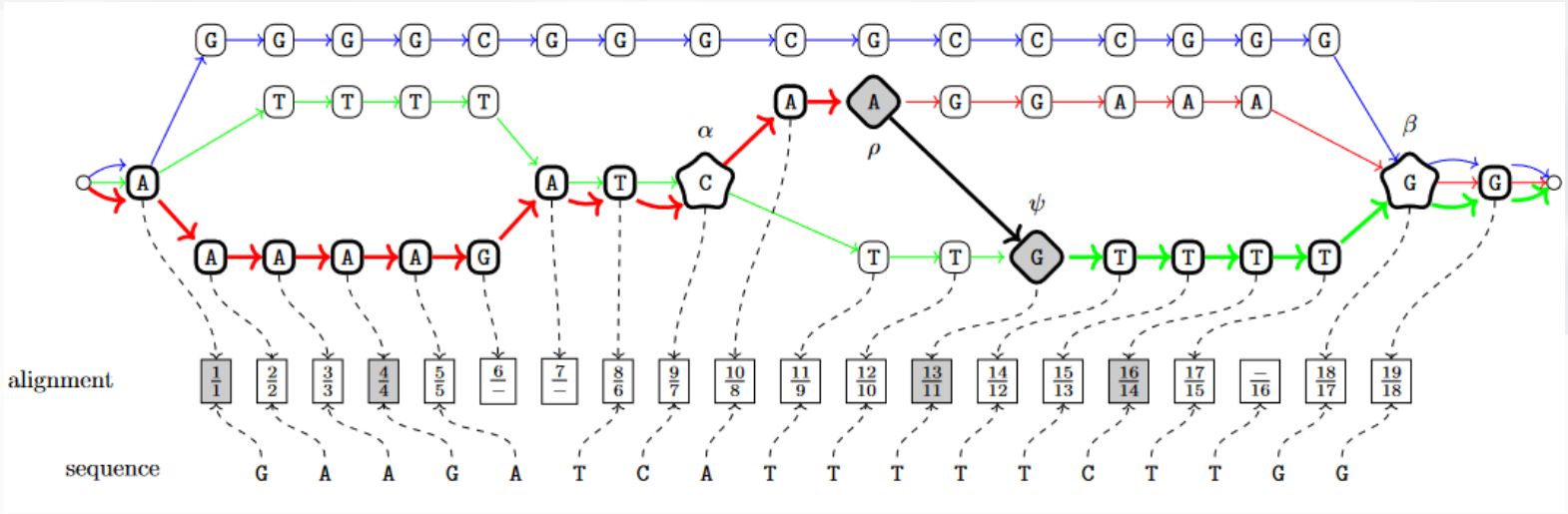
Esempio di un **grafo di variazione**

Allineamento con ricombinazione

- **Ricombinazione**: scambio (parziale) di materiale genetico tra cromosomi omologhi, che può portare a variazioni genetiche

- **Allineamento** con **al più una ricombinazione**:
  - Grafi di variazione
  - $T(n, m, p) = O(n^2 \cdot m \cdot p^2)$

- Obiettivo: migliorare le **prestazioni** con l'**algoritmo *wavefront***

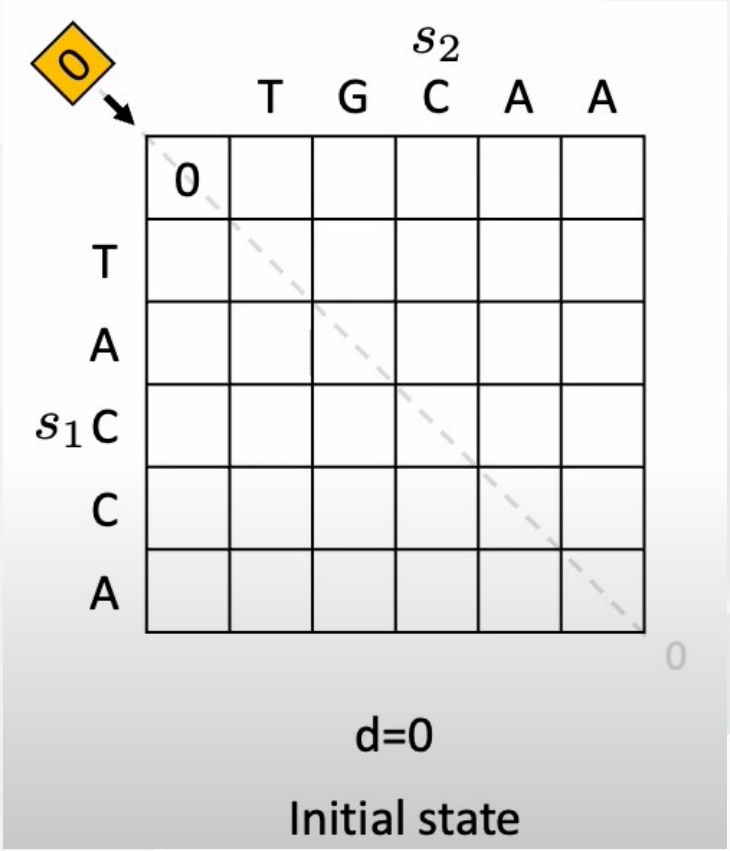


Algoritmo *wavefront*

- Calcola la **distanza di edit ottimale** tra due sequenze

- Basato su:
  i. **Diagonali** della matrice **crescenti**
  ii. **Fronte d'onda**: insieme delle celle della matrice con la **stessa distanza di edit**

- Complessità
  i. **Tempo:** $T(n, m, d) = O(min\{n, m\} \cdot d + d^2)$
  ii. **Spazio:** $M(n, m, d) = O(d^2)$

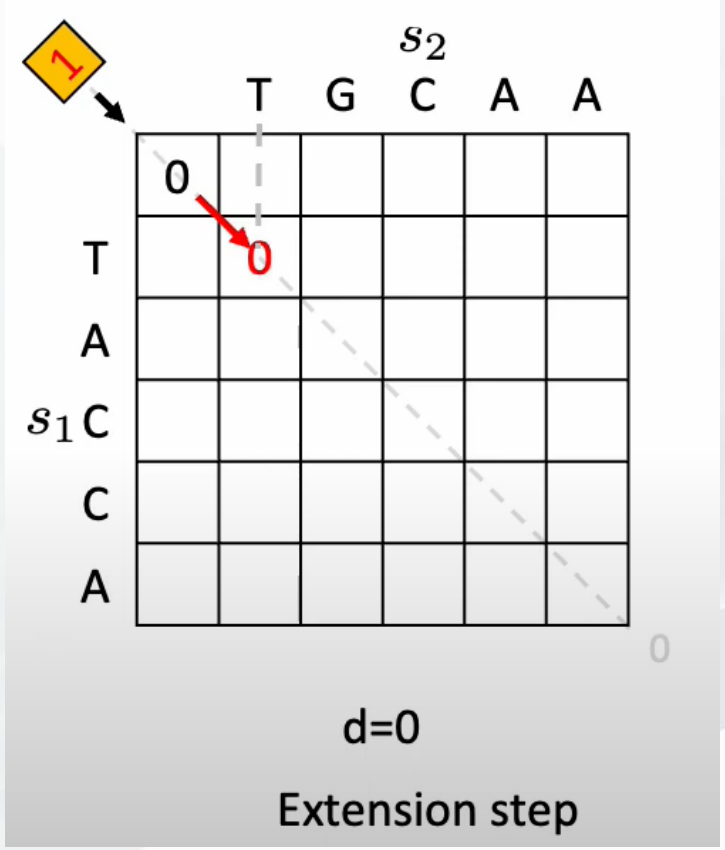


Algoritmo *wavefront*

- Calcola la **distanza di edit ottimale** tra due sequenze

- Basato su:
  i. **Diagonali** della matrice **crescenti**
  ii. **Fronte d'onda**: insieme delle celle della matrice con la **stessa distanza di edit**

- Complessità
  i. **Tempo:** $T(n, m, d) = O(min\{n, m\} \cdot d + d^2)$
  ii. **Spazio:** $M(n, m, d) = O(d^2)$

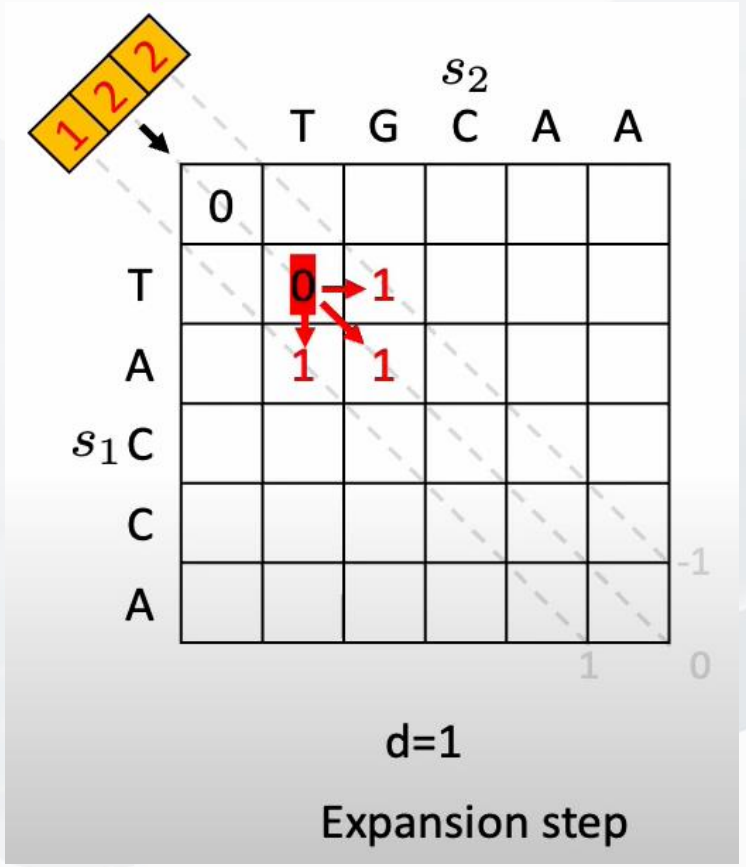


Algoritmo ***wavefront***

- Calcola la **distanza di edit ottimale** tra due sequenze

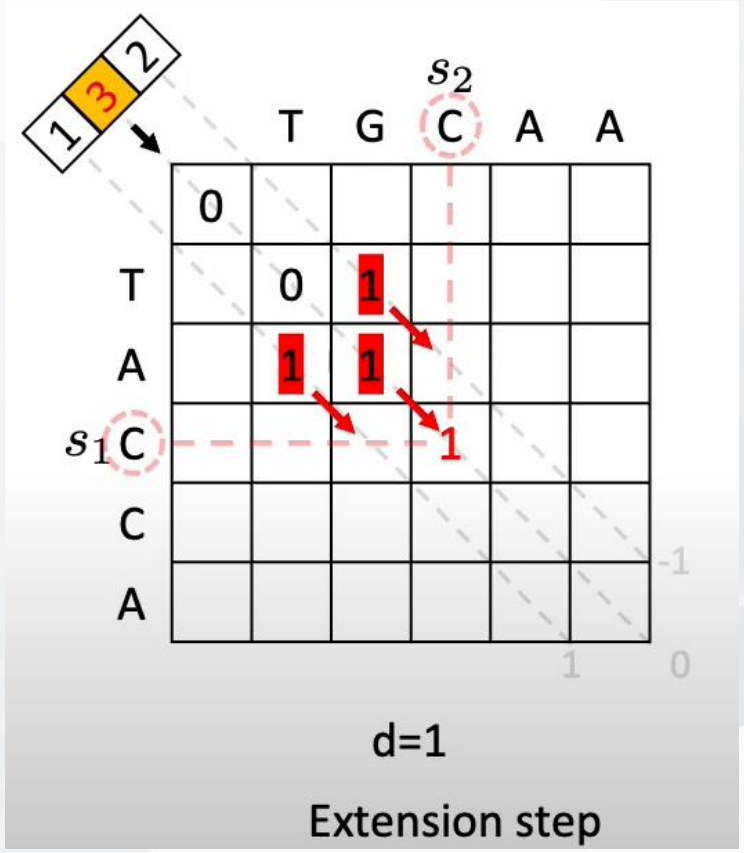
- Basato su:
  i. **Diagonali** della matrice **crescenti**
  ii. **Fronte d'onda**: insieme delle celle della matrice con la **stessa distanza di edit**

- Complessità
  i. **Tempo:** $T(n, m, d) = O(min\{n, m\} \cdot d + d^2)$
  ii. **Spazio:** $M(n, m, d) = O(d^2)$



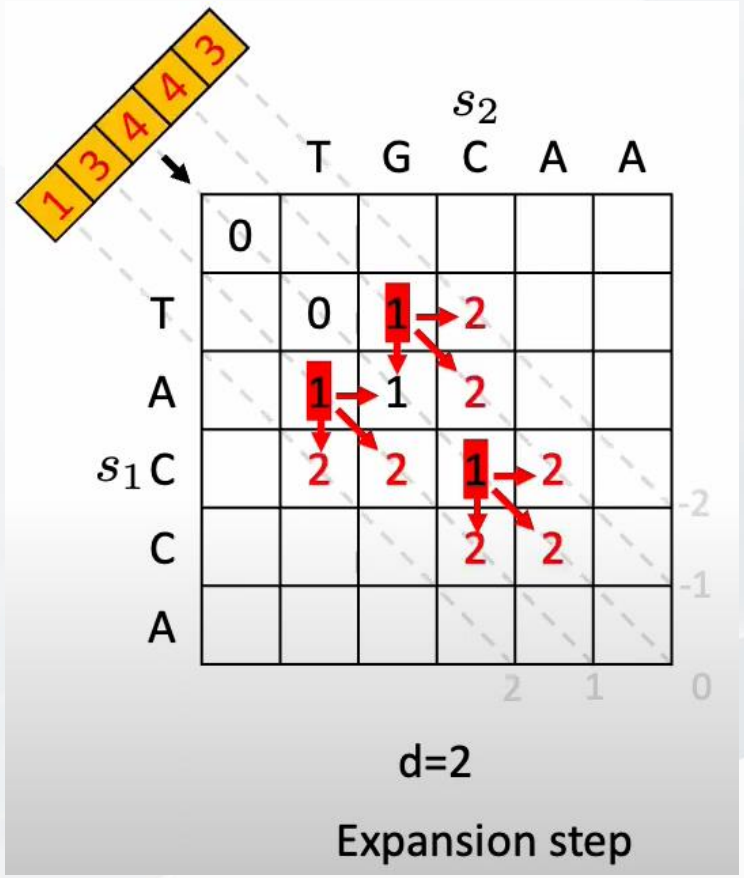
Algoritmo *wavefront*

- Calcola la **distanza di edit ottimale** tra due sequenze
- Basato su:
  i. **Diagonali** della matrice **crescenti**
  ii. **Fronte d'onda**: insieme delle celle della matrice con la **stessa distanza di edit**
- Complessità
  i. **Tempo:** $T(n, m, d) = O(min\{n, m\} \cdot d + d^2)$
  ii. **Spazio:** $M(n, m, d) = O(d^2)$



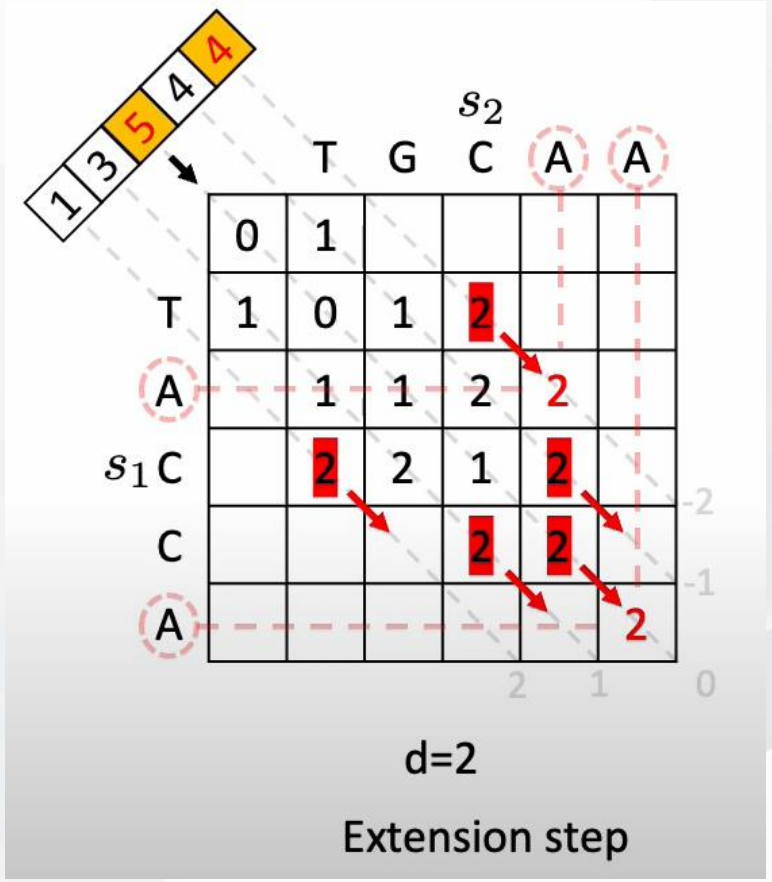
Algoritmo *wavefront*

- Calcola la **distanza di edit ottimale** tra due sequenze
- Basato su:
  i. **Diagonali** della matrice **crescenti**
  ii. **Fronte d'onda**: insieme delle celle della matrice con la **stessa distanza di edit**
- Complessità
  i. **Tempo:** $T(n, m, d) = O(min\{n, m\} \cdot d + d^2)$
  ii. **Spazio:** $M(n, m, d) = O(d^2)$



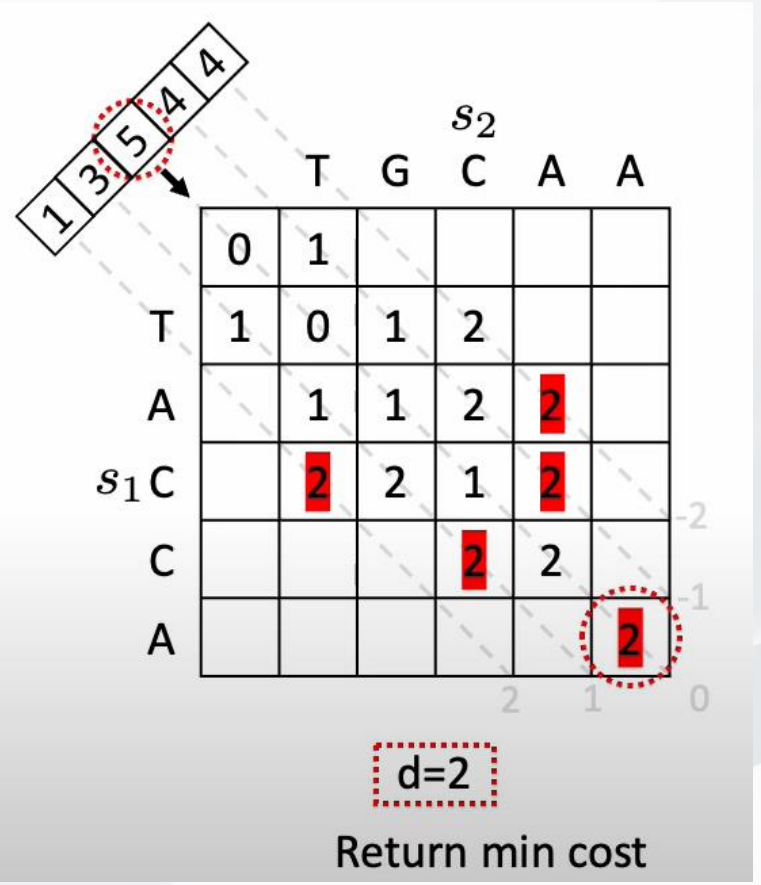
Algoritmo *wavefront*

- Calcola la **distanza di edit ottimale** tra due sequenze
- Basato su:
  i. **Diagonali** della matrice **crescenti**
  ii. **Fronte d'onda**: insieme delle celle della matrice con la **stessa distanza di edit**
- Complessità
  i. **Tempo:** $T(n, m, d) = O(min\{n, m\} \cdot d + d^2)$
  ii. **Spazio:** $M(n, m, d) = O(d^2)$



Algoritmo ***wavefront***

- Calcola la **distanza di edit ottimale** tra due sequenze
- Basato su:
  i. **Diagonali** della matrice **crescenti**
  ii. **Fronte d'onda**: insieme delle celle della matrice con la **stessa distanza di edit**
- Complessità
  i. **Tempo:** $T(n, m, d) = O(min\{n, m\} \cdot d + d^2)$
  ii. **Spazio:** $M(n, m, d) = O(d^2)$

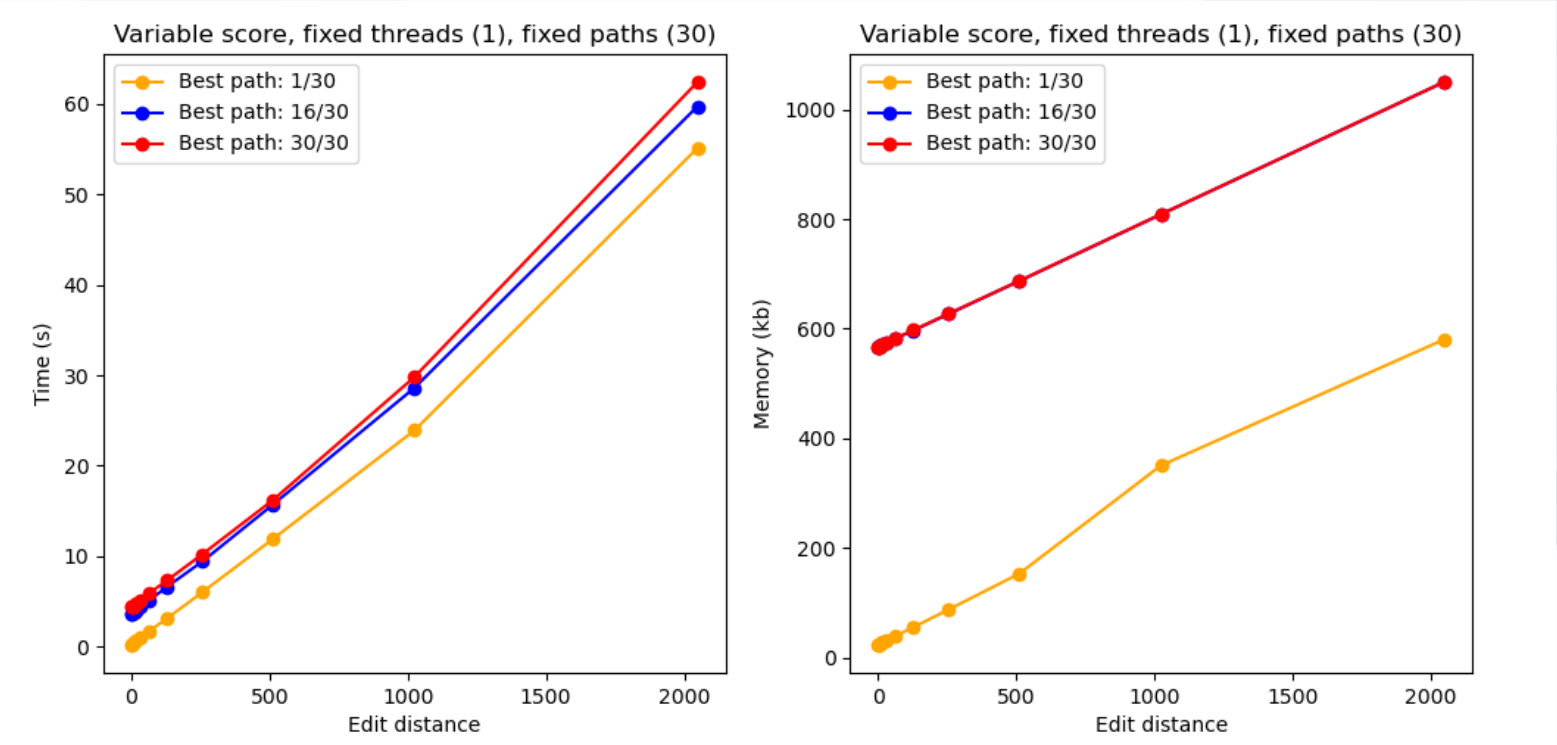


Implementazione

- Prototipo per effettuare allineamenti tra **grafi di variazione** e **sequenze** con **algoritmo *wavefront***
  - Linguaggio di programmazione **Rust**
  - https://github.com/iFoxz17/WF_Recgraph
- Primo approccio: adattamento di ***wavefront*** a **POA**
  - **Problema:** troppo complesso
- Secondo approccio: estrazione dei percorsi e **multithreading**
  - **Vantaggio: esecuzione parallela** di allineamenti su più cammini
  - **Svantaggio: ridondanza** dei calcoli su vertici che appartengono a più cammini

Sperimentazione

- **Tre gradi di libertà**
  1. **Distanza di edit**
  2. Numero di **threads**
  3. Numero di **percorsi**
- **Risultati:** tempo
  1. $\Theta(d)$
  2. $\Theta(\frac{1}{t})$
  3. $\Theta(p)$



Confronto con *RecGraph* (1)

- **Grafo**:
  - $\approx$ 50000 bp
  - **Percorsi** (p): 30
  - **Lunghezza media percorsi** (n)**:** $\approx$ 29000 bp

- **Reads** (m):
  - 150 bp
  - 1000 bp
  - 10000 bp
  - 25000 bp

- Genoma appartenente al virus **SARS-CoV-2**

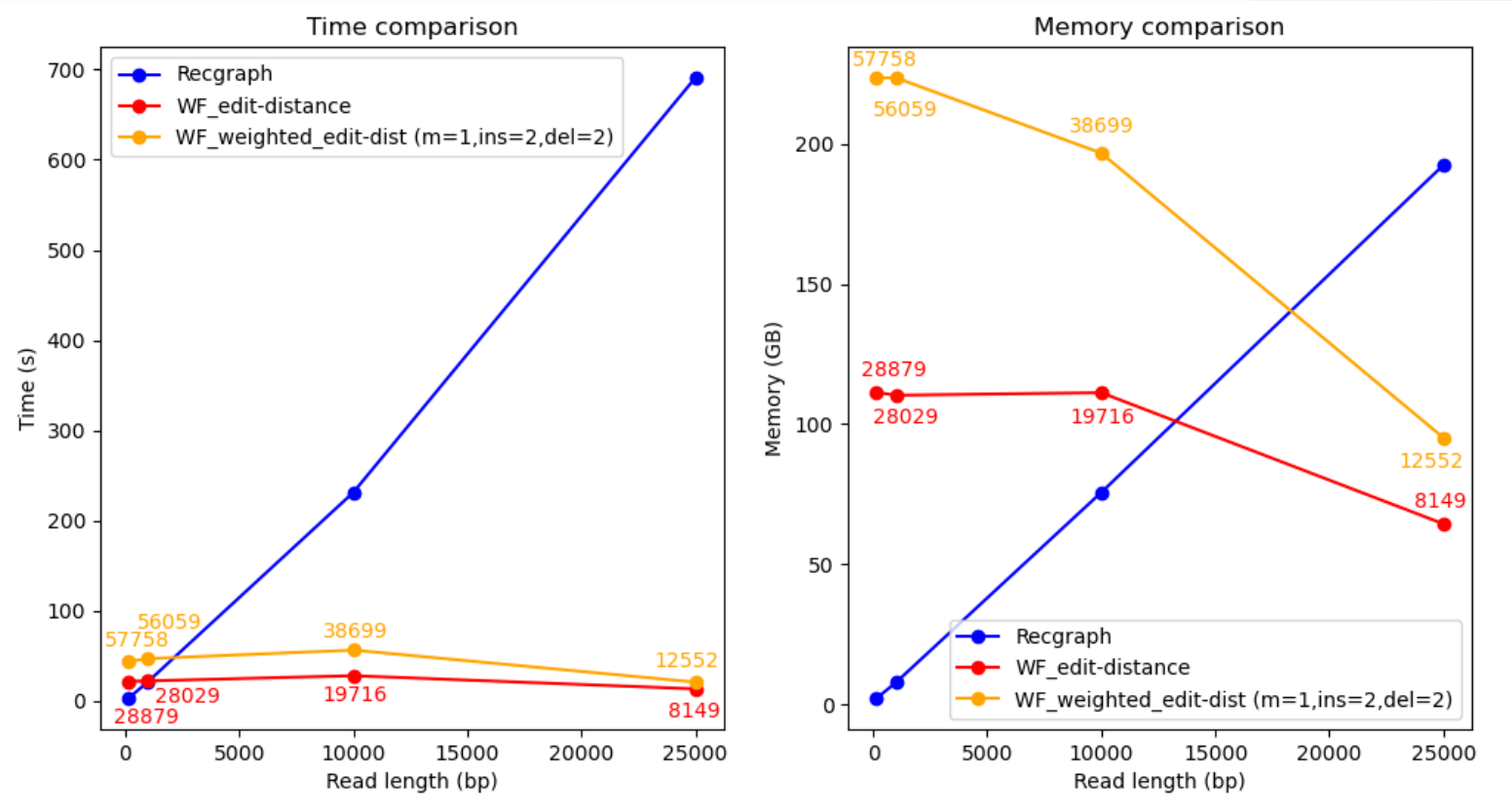
Sperimentazione eseguita su un server con 64 **core**, 256 GB di **RAM**

Confronto con *RecGraph* (2)

Modalità **globale**

$$T(n,m,d,p) = O(\min\{n,m\} \cdot d \cdot p)$$

$$M(n,m,d,p) = O(max\{n,m\} \cdot d \cdot p)$$



Riepilogo

- **Allineamento di sequenze:**
  i. Studio della **generalizzazione** di **Needleman-Wunsch** su **grafi**
  ii. Studio dell'**algoritmo *wavefront***
  iii. Generalizzazione di ***wavefront*** su **grafi di variazione**

- **Implementazione**
  i. Implementazione di un **prototipo** per migliorare le prestazioni di ***RecGraph***
  ii. Linguaggio di programmazione **Rust**
  iii. **Confronto sperimentale** con ***RecGraph***