

# GA - DPAMSA

Algoritmo genetico e Deep Reinforcement Learning  
per l'allineamento multiplo di sequenze

Studente:

**Gabriele Tuozzo**

Docente:

**Rocco Zaccagnino**

# INDICE

01

OBIETTIVI DEL  
PROGETTO

04

RISULTATI

02

BACKGROUND

05

CONSIDERAZIONI

03

L'ALGORITMO  
GENETICO

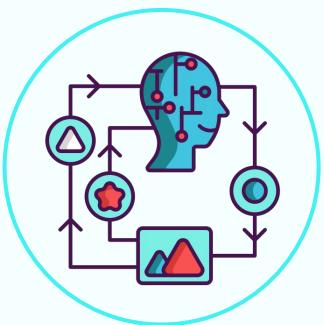
06

CONCLUSIONI

# OBIETTIVI DEL PROGETTO

- E' possibile utilizzare un modello di **Reinforcement Learning (RL)** addestrato per risolvere un problema di **allineamento multiplo di sequenze P**, per risolvere in maniera **efficace ed efficiente** un problema **P' più complesso** ?
- Gli **algoritmi genetici (AG)** possono aiutarci in questo contesto?
- E' possibile **migliorare i tool di allineamento multiplo** basati sul RL con gli AG ?

# BACKGROUND



## REINFORCEMENT LEARNING

Ambito del Machine Learning in cui un agente impara a comportarsi in un ambiente, **compiendo azioni e osservando i risultati** di queste azioni



## ALGORITMI GENETICI

Gli algoritmi genetici sono una classe di algoritmi di ottimizzazione ispirati ai **processi di evoluzione naturale**, la **mutazione** e il **crossover**.

# REINFORCEMENT LEARNING

- Il RL è un'area del **Machine Learning**, in cui un agente impara a compiere delle azioni in un ambiente per massimizzare una ricompensa cumulativa.
- Si basa sul **principio di rinforzo**, cioè sull'idea che le azioni che portano a risultati positivi tendono a essere **ripetute**, mentre quelle che portano a risultati negativi tendono a essere **evitate**
- Si differenzia dall'apprendimento supervisionato perché non richiede che siano presentate coppie di input/output etichettate. Invece, l'attenzione è rivolta a trovare un equilibrio tra **l'exploration e l'exploitation**.

# REINFORCEMENT LEARNING

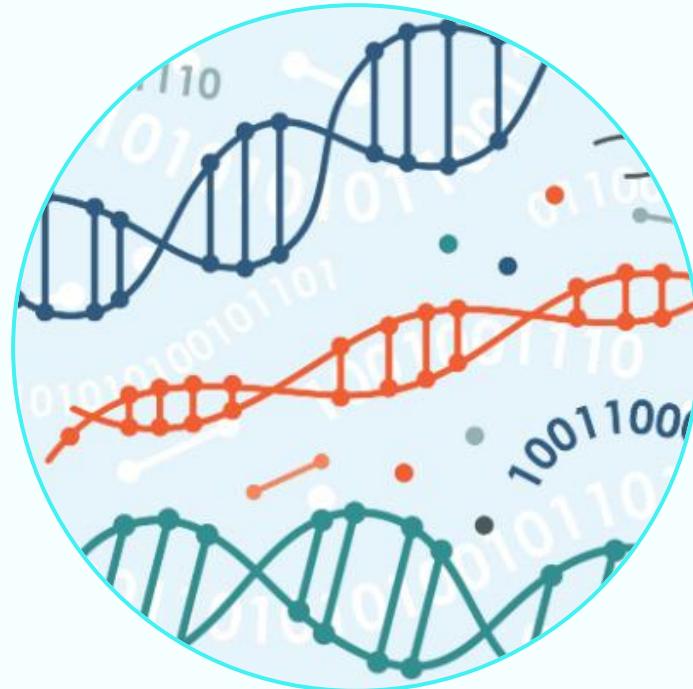
Le componenti fondamentali sono:

- **L'ambiente**: tutto ciò con cui interagisce l'agente. L'ambiente è descritto da uno stato **S** che **cambia** in risposta alle azioni dell'agente.
- **L'agente**: è l'entità che prende decisioni e **interagisce con l'ambiente**.
- **Stato**: rappresenta la **situazione attuale** dell'ambiente.
- **Azione**: sono le scelte disponibili all'agente. Ogni azione influisce sullo stato dell'ambiente.
- **Ricompensa**: è il feedback ricevuto dall'agente dopo aver compiuto un'azione
- **Q-function**: funzione valore che specifica la qualità di una coppia stato-azione.

# REINFORCEMENT LEARNING - ALGORITMI PRINCIPALI

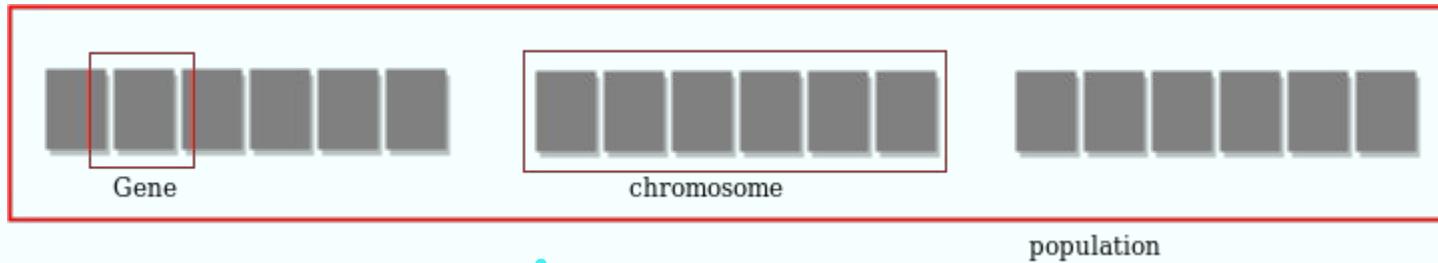
- **Q-Learning:** è un algoritmo di apprendimento (model free) che mira ad apprendere la Q-function ottimale, cioè quella che massimizza la ricompensa cumulativa.
- **Deep Q-Networks (DQN):** utilizza le deep neural network per approssimare la Q function, permettendo di gestire stati e azioni dinamiche e complesse.

# Gli algoritmi genetici



# Algoritmi genetici

- Gli algoritmi genetici sono una classe di algoritmi di ottimizzazione ispirati ai **processi di selezione naturale**
- Si tratta di uno sfruttamento intelligente delle ricerche casuali per dirigere la ricerca nella **regione di migliori prestazioni** nello spazio delle soluzioni
- **Simulano la sopravvivenza del più adatto** tra gli individui di generazioni consecutive.

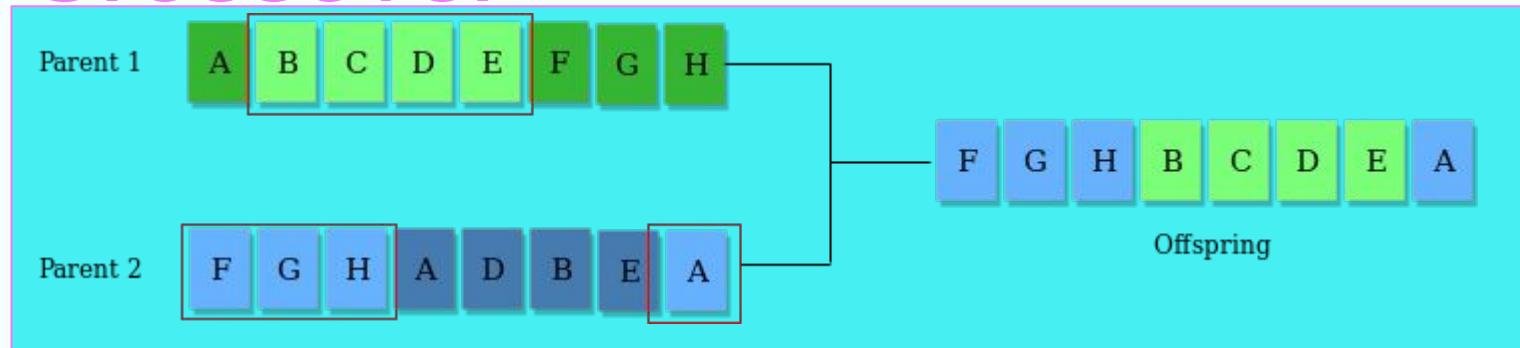


# Algoritmi genetici

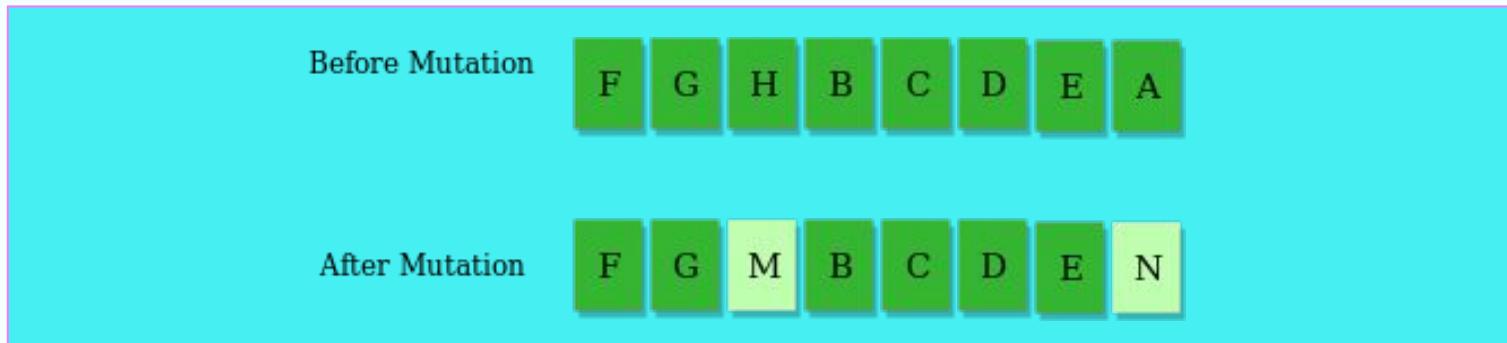
- Ad ogni individuo viene assegnato un **punteggio di fitness** e si cercano gli individui con un punteggio di fitness **ottimale (o quasi)**
- Una volta creata la generazione iniziale, l'algoritmo fa evolvere la popolazione usando i seguenti operatori: **selezione, crossover e mutazione.**
  - **Selezione:** si da la preferenza agli individui con un buon fitness score e si permette loro di passare i propri geni alle generazioni successive.

# Algoritmi genetici

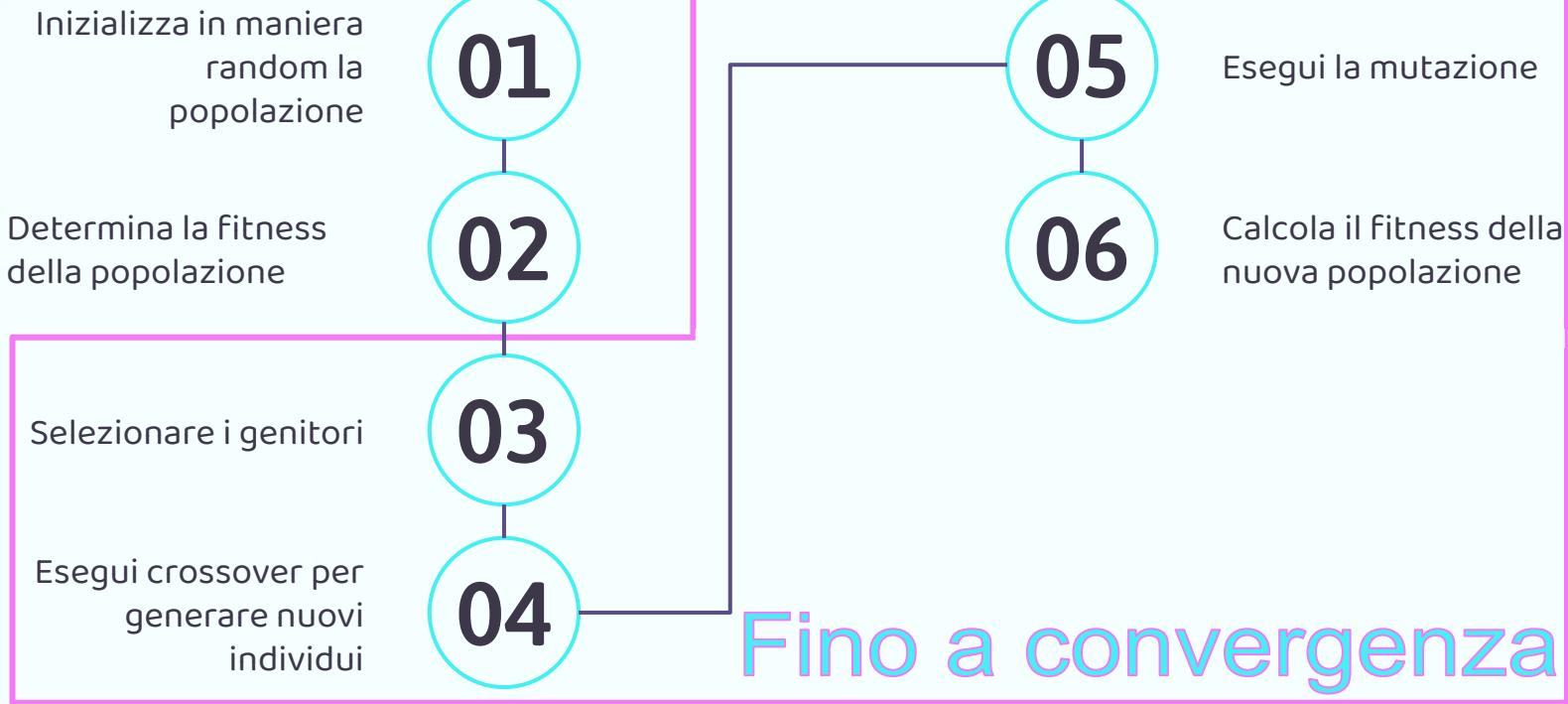
## Crossover



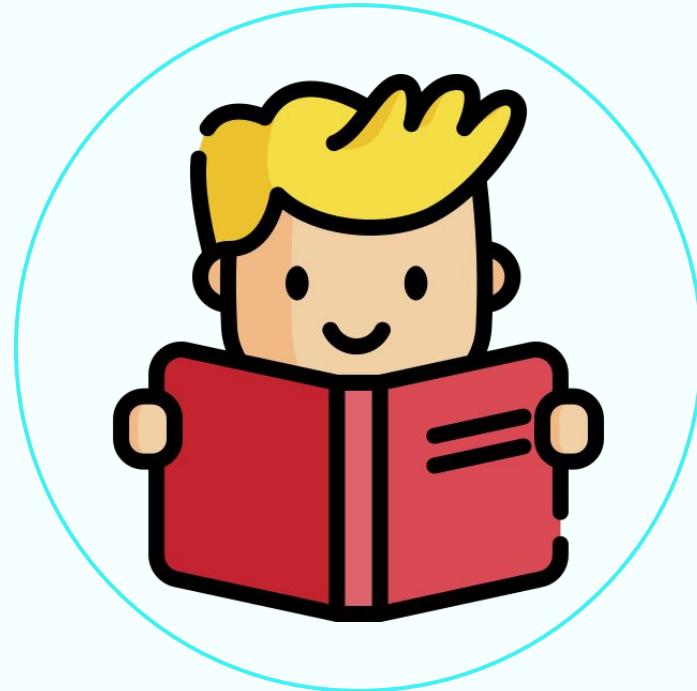
## Mutazione

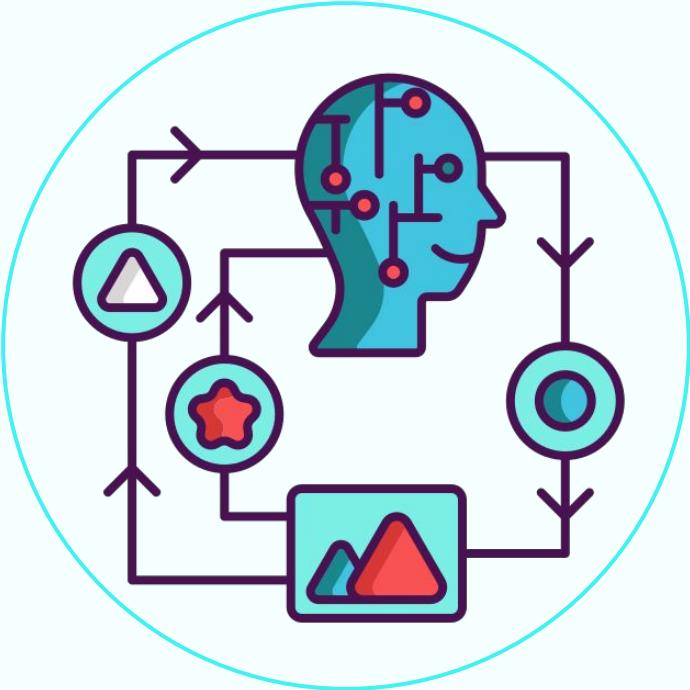


# Passaggi di un algoritmo genetico



# Stato dell'arte

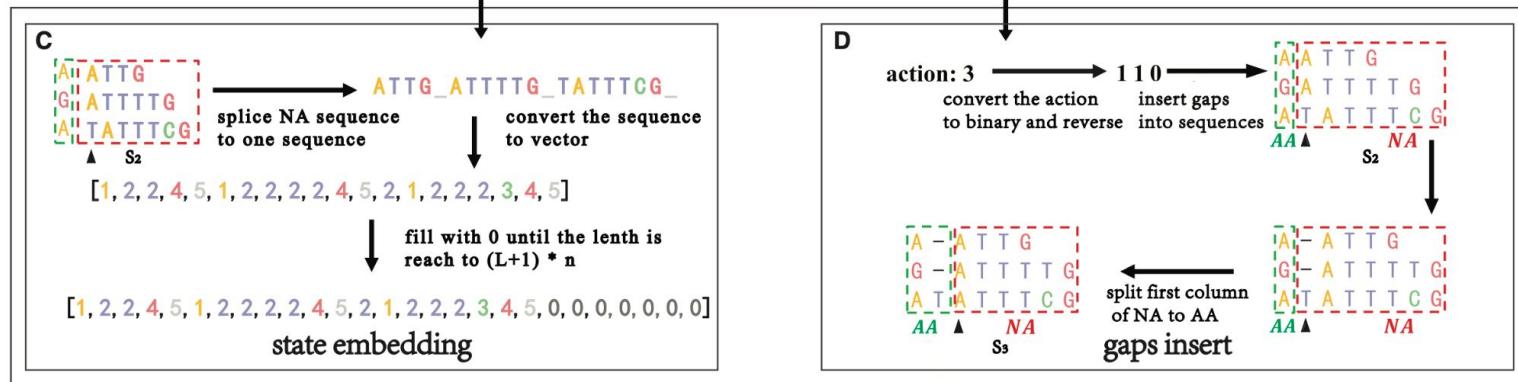
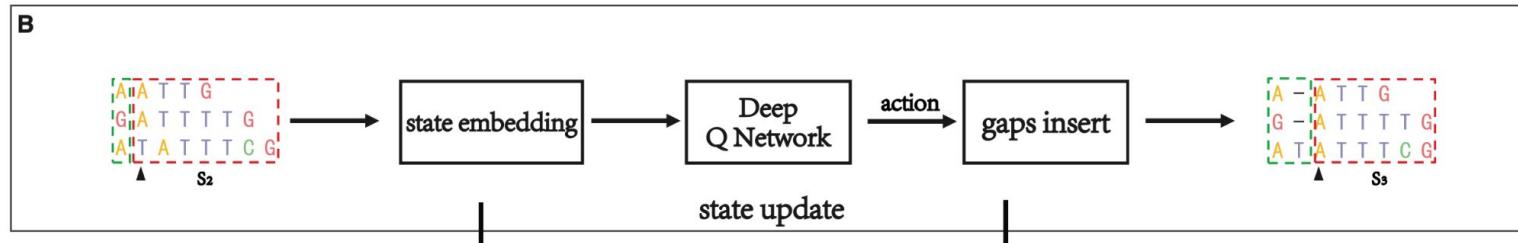
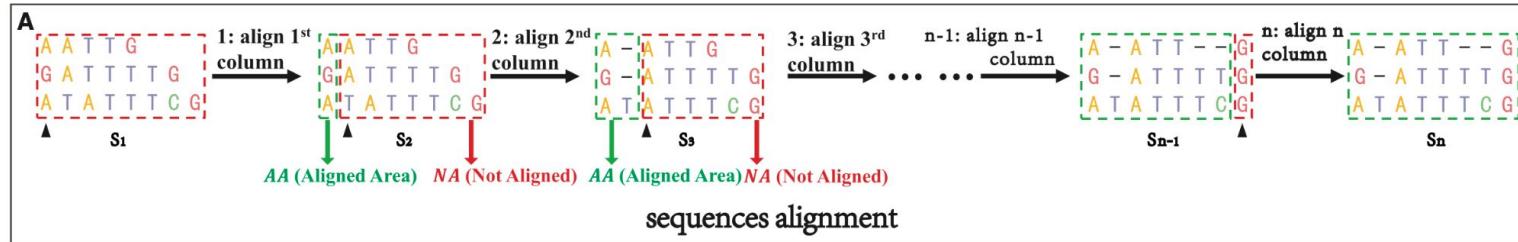




# DPAMSA

Deep reinforcement learning,  
Positional encoding e self Attention  
per il Multi Sequence Alignment

# DPAMSA





03

# L'ALGORITMO GENETICO E IL RL PER IL MSA

# L'algoritmo genetico e il RL per il MSA

CCGACAAT  
GCCCTTAA  
TAGGGAGA  
CTTTCCGC

L'allineamento  
da eseguire  
**(board)**

CCGACAAT  
GCCCTTAA  
TAGGGAGA  
CTTTCCGC

**Quello che sa  
risolvere il  
modello di RL**

# 1. Generazione della popolazione

Population

Individual 0

33413112  
43332211  
21444141  
32223343

Individual 1

33413112  
43332211  
21444141  
32223343

Individual 3

33413112  
43332211  
21444141  
32223343

Individual 4

33413112  
43332211  
21444141  
32223343

A:1

T:2

C:3

G:4

- :5

## 2. Fitness score e Selection

- Il **Fitness score** di ogni individuo è calcolato utilizzando la **sum-of-pairs**.
- La fase di **Selection** quindi, consisterà nel selezionare gli ***n*** individui con score della sum-of-pairs più alto e saranno loro a **generare nuovi individui** ed a passare all'iterazione successiva.

$$p(x, y) = \begin{cases} -4 & x = gap | y = gap \\ 4 & x = y \\ -4 & x \neq y \end{cases}$$

## 2. Mutazione

- Abbiamo previsto due tipi di mutazione:
  - **Random mutation:** si seleziona in maniera random l'individuo da mutare, in base al numero di possibile sub-board diverse, eseguiamo lo stesso numero di mutazioni.
  - **Best fitted individual:** vado ad operare la mutazione sull'individuo con score della sum-of-pairs più alta, sulla sub-board con sum-of-pairs più bassa.
- Il primo è chiaramente più **efficiente**, nel secondo caso invece, si vuole una **qualità dell'allineamento maggiore** a discapito delle prestazioni.

## 2. Mutazione

### Mutation

Reinforcement  
Learning Agent

Individual

33413112  
43332211  
21444141  
32223343

## 2. Dopo la Mutazione

After the Mutation

Reinforcement  
Learning Agent

Individual

533413112  
43332211  
21444141  
32223343

## 4. Crossover

Due metodi di crossover diversi implementati:

- **Vertical crossover**
- **Horizontal crossover**

Prestazioni simili, ma Horizontal crossover **leggermente più veloce**

## 4. Vertical Crossover

Parent 1

33413112
43332211
21444141
32223343

Parent 2

5334135112
4333221155
2144414155
3222334355

Offspring

3341351125
4333221155
2144414155
3222334355

## 4. Horizontal Crossover

Parent 1

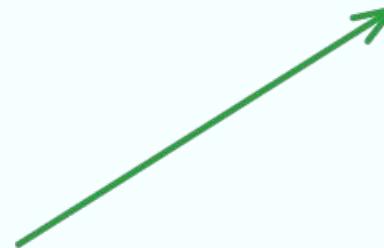
334135112  
433322115  
214441415  
322233435

Parent 2

5334135112  
4333221155  
2144414155  
3222334355

Offspring

334135112  
433322115  
214441415  
322233435



# L'algoritmo genetico - Recap

RL con Random  
o Worst fitted

sum-of-pairs

n individui con sum-of-pairs più alta

Vertical o Horizontal



# RISULTATI

Nome Dataset	Numero di test	Numero di sequenze per ogni test	Lunghezza di ogni sequenza
dataset1_3x30	50	3	30(bp)
dataset1_3x60	50	3	60(bp)
dataset1_6x60	50	6	60(bp)



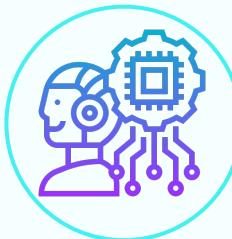
ClustalW



ClustalΩ



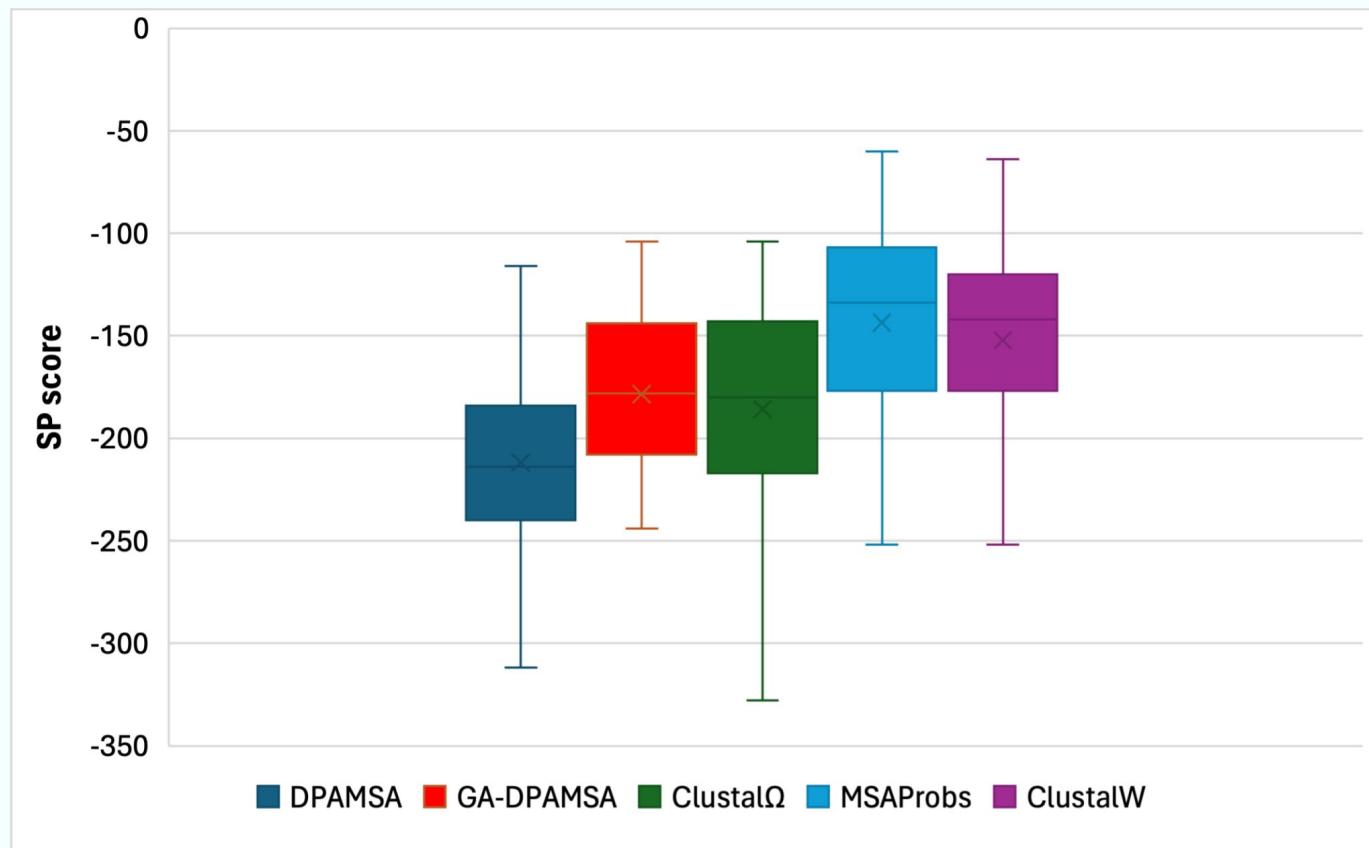
MSAProbs



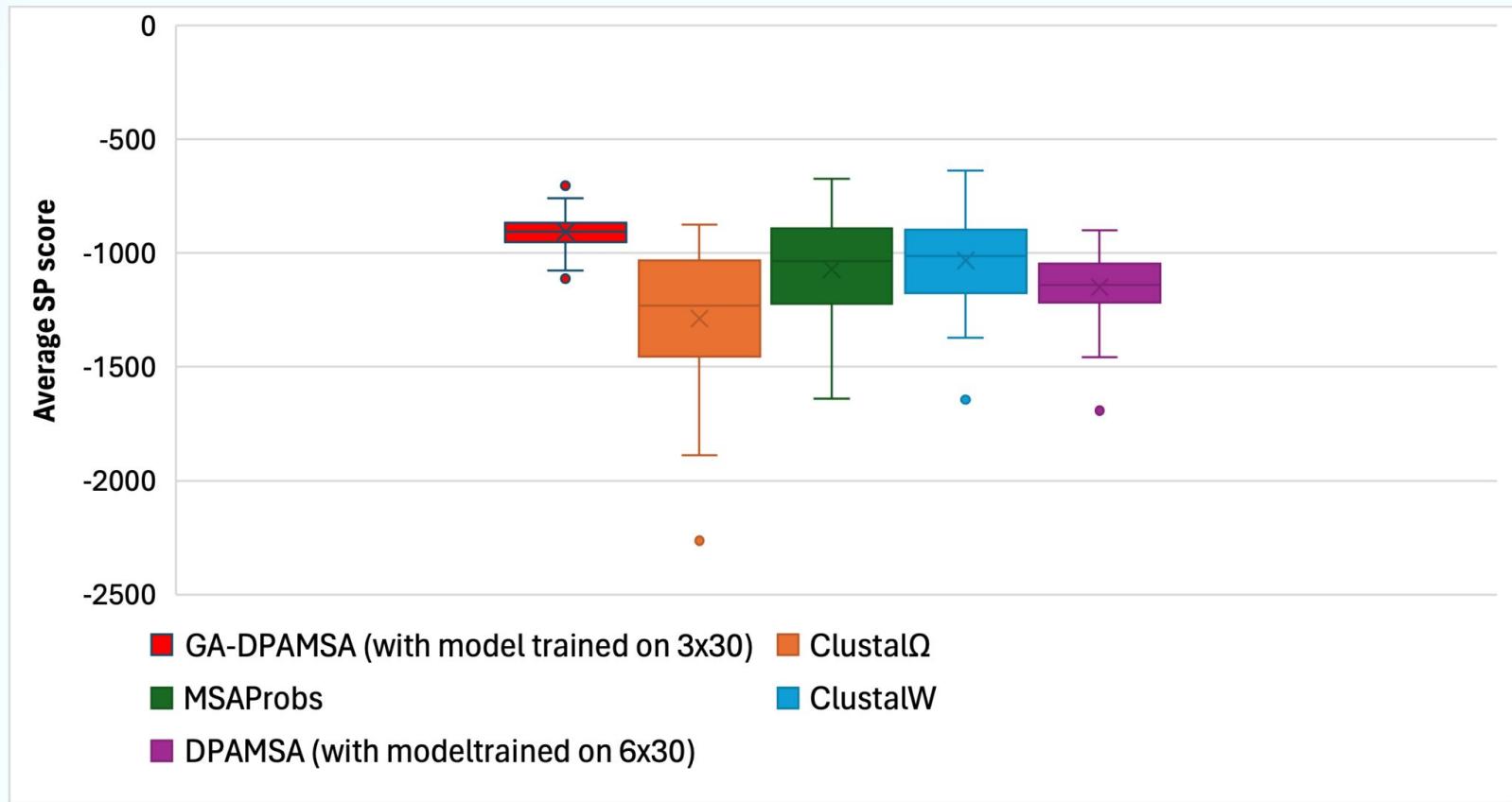
DPAMSA  
(Senza AG)

Parametro AG	Valore
Size popolazione	5
Numero di iterazioni	3
percentuale di individui da mutare	20%

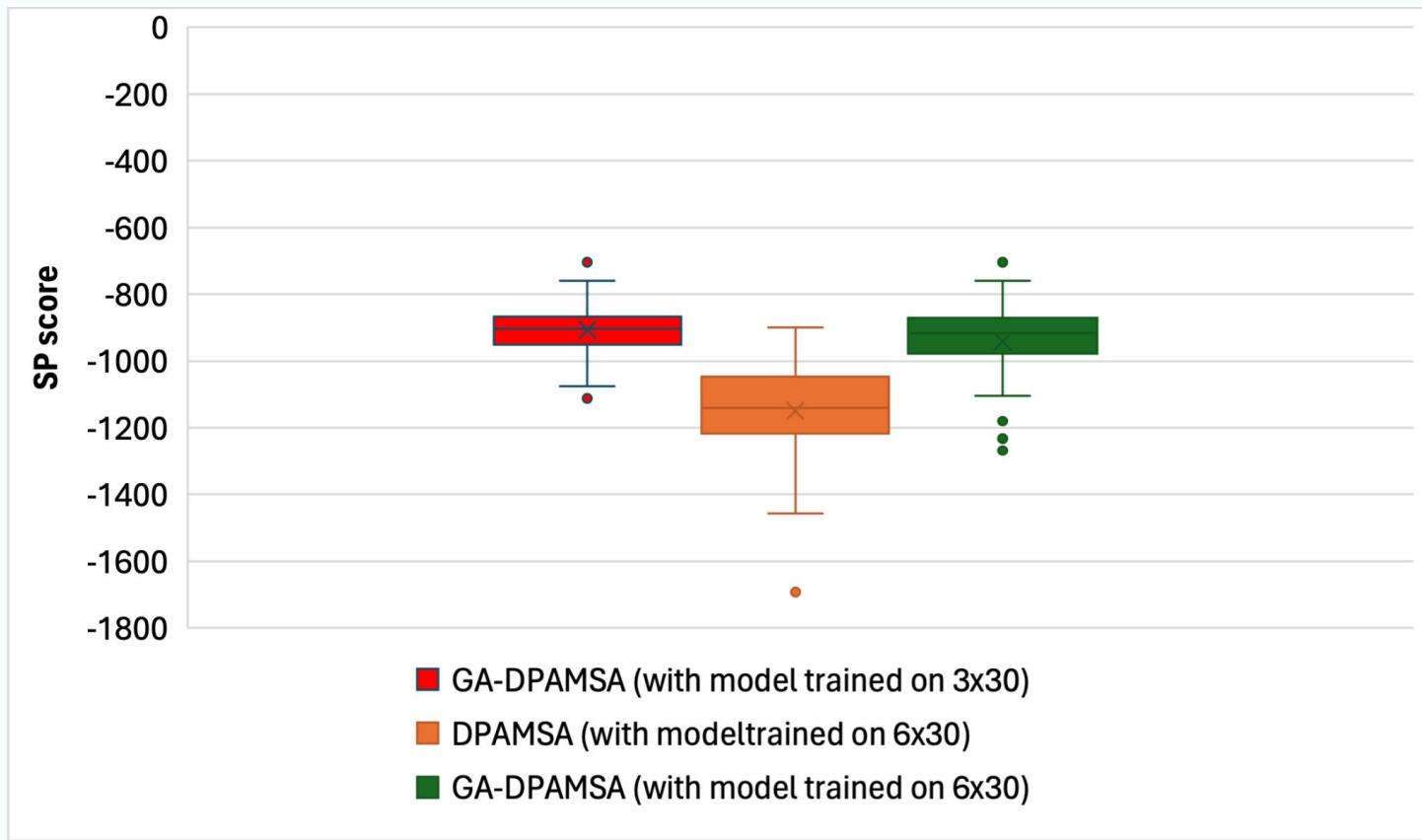
# RISULTATI: dataset1\_3x30



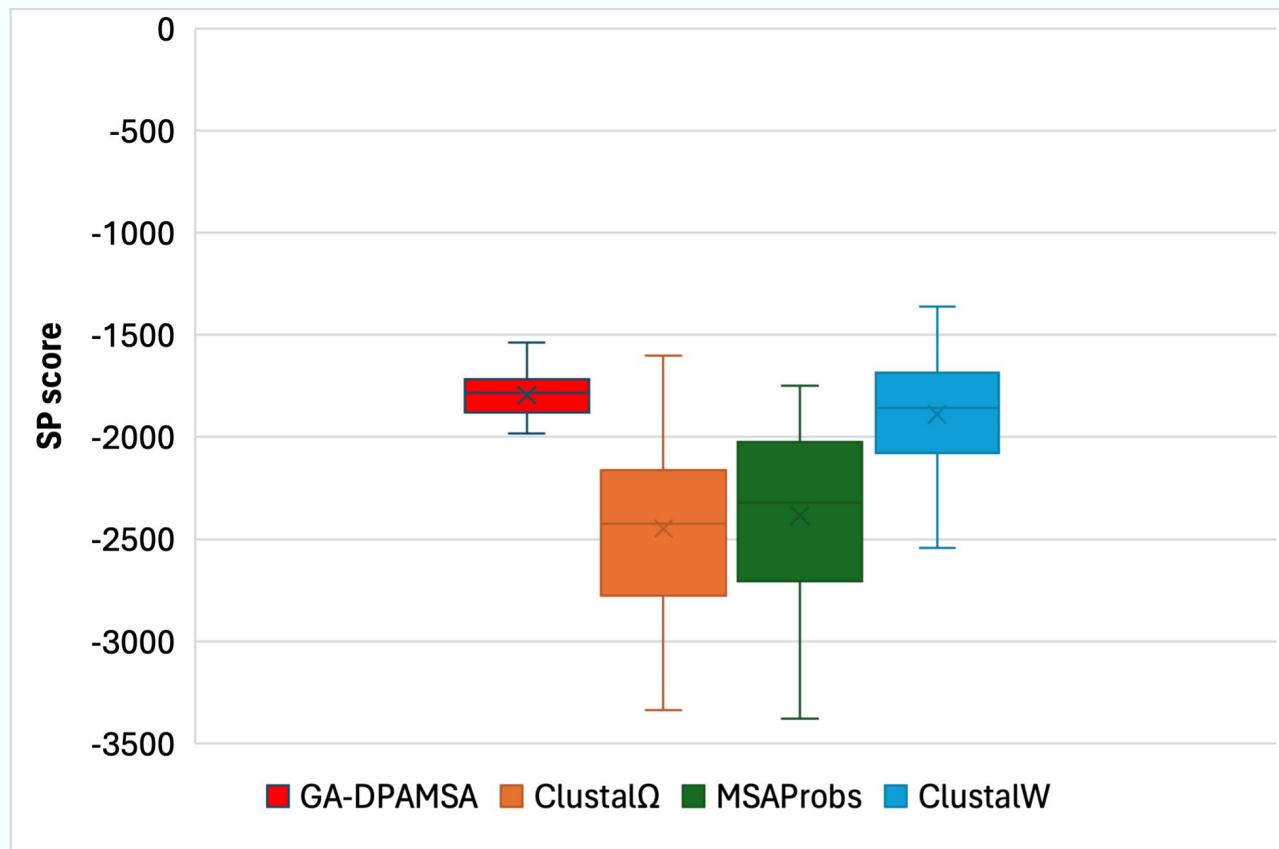
# RISULTATI: dataset1\_6x30



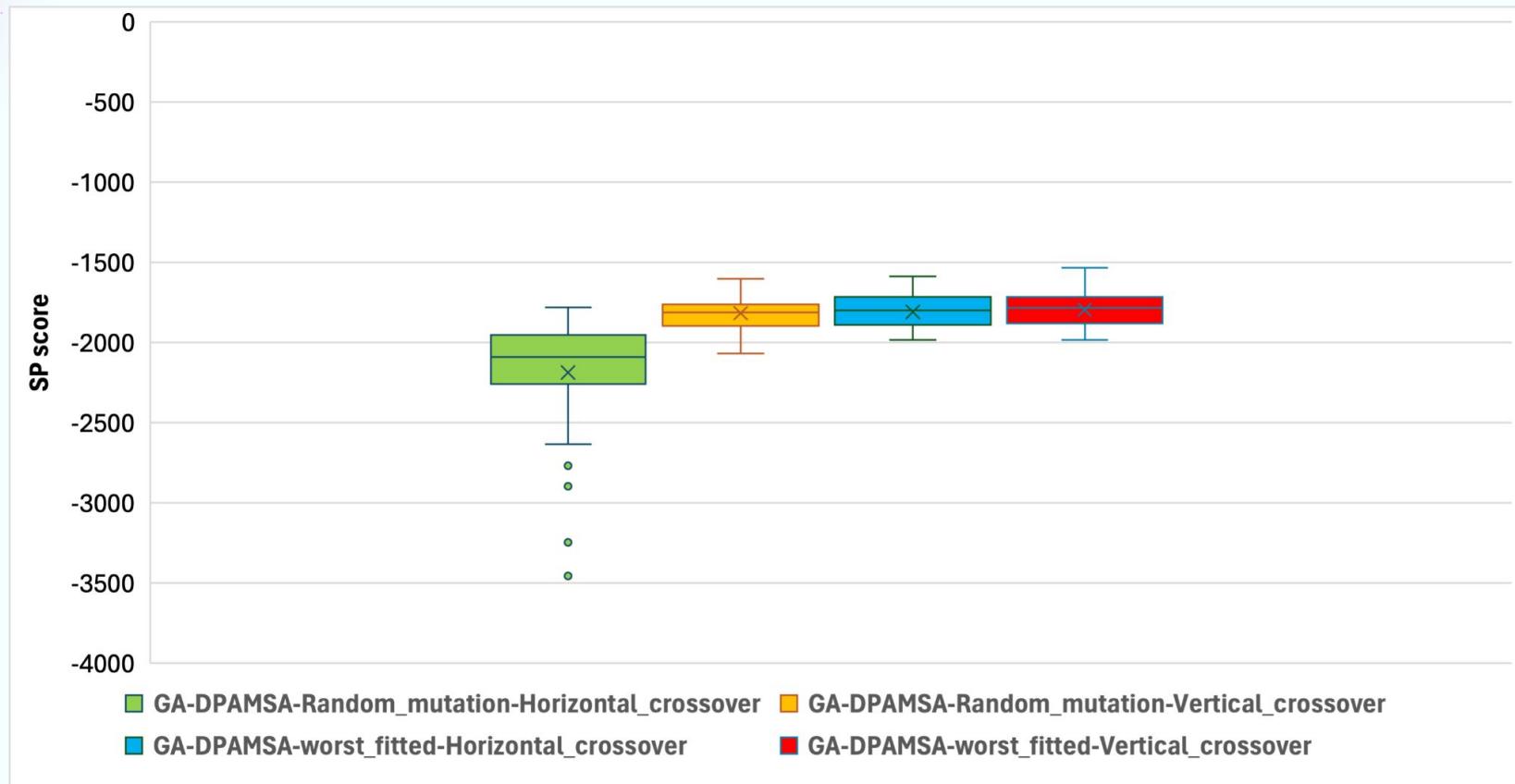
# dataset1\_6x30 DPAMSA vs GA-DPAMSA



# RISULTATI: dataset1\_6x60



# RISULTATI: GA-DPAMSA su diverse configurazioni



# Considerazioni

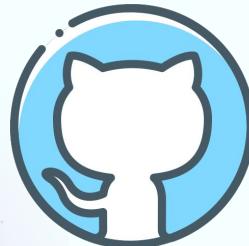
- **Solamente** nel caso del dataset 3x30bp non siamo riusciti ad essere migliori, ma **all'aumentare della size del problema**, gli allineamenti prodotti risultano essere **migliori rispetto agli altri 3 tool**
- L'algoritmo genetico ci da effettivamente dei **miglioramenti** rispetto al **modello base**
- Riusciamo ad avere prestazioni paragonabili (anche leggermente migliori) con un modello addestrato per un **problema più piccolo**, a **risolvere uno più grande**.
- Possiamo concludere che l'esperimento è risultato **efficace** e gli obiettivi predisposti all'inizio sono stati **raggiunti**

# Sviluppi futuri

- Testare il tool con dataset ancora **più grandi** e con sequenze di DNA reali
- Misurare le performance dell'AG utilizzando un **altro modello di RL**
- Adattare il codice per l'allineamento delle **proteine** ed **RNA** (ovviamente andrebbe cambiato anche il modello di RL)
- Testare l'algoritmo utilizzando un **approccio ibrido** per la **mutazione** ed il **crossover**
  - Ad esempio, potrei **scegliere a caso**, lanciando una moneta, se in una **determinata iterazione** utilizzare il vertical crossover o l'horizontal, e stessa cosa per la mutazione, se usare la random mutation o il best fitted

# Grazie per l'attenzione

Domande?



Il codice dell'applicazione con i dataset di training ed inferenza, insieme ai file di report dei risultati, sono disponibili su **GitHub** nell'organizzazione del corso:  
**<https://bit.ly/ga-dpamsa>**