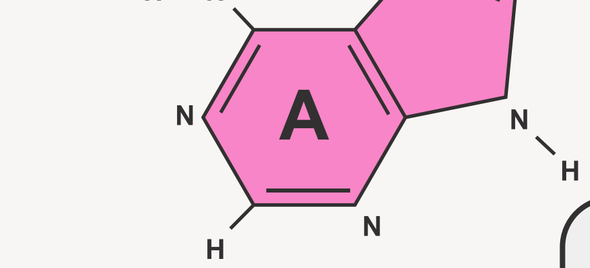
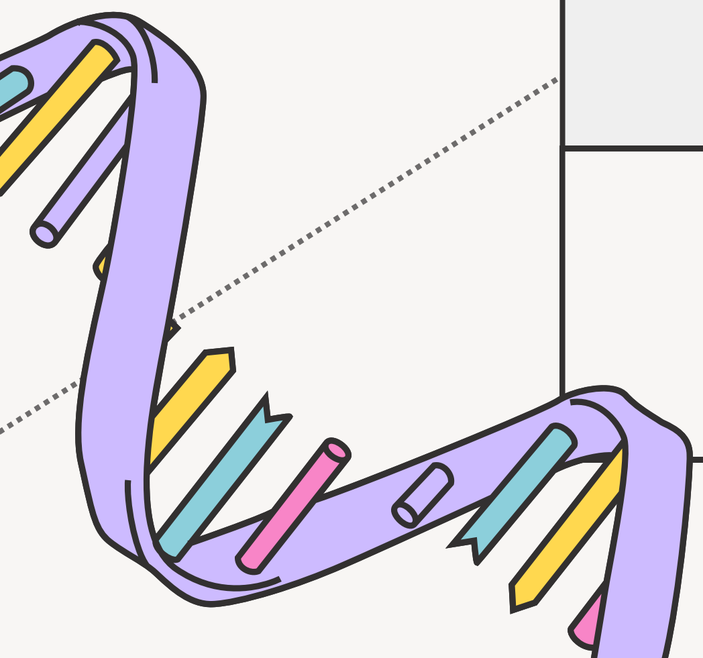
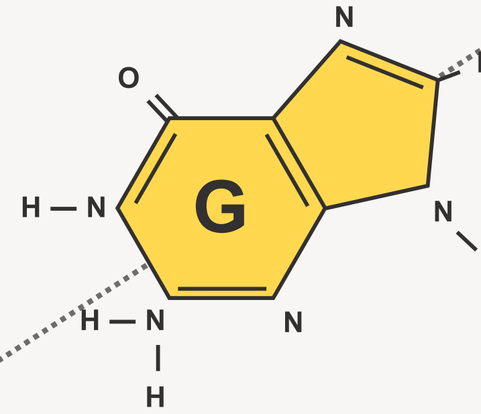


UNO STUDIO PRELIMINARE PER LA FATTORIZZAZIONE DI LYNDON PER L'ALLINEAMENTO MULTIPLO CON GLI ALGORITMI GENETICI

Mihail Purice
Maria Lombardi

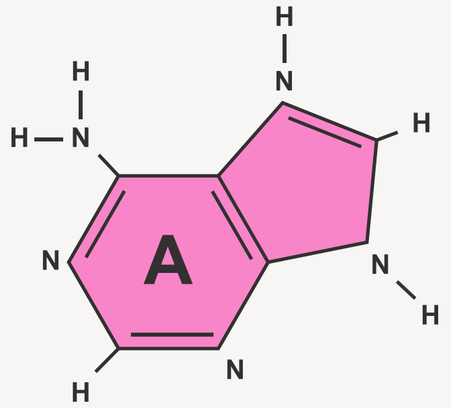




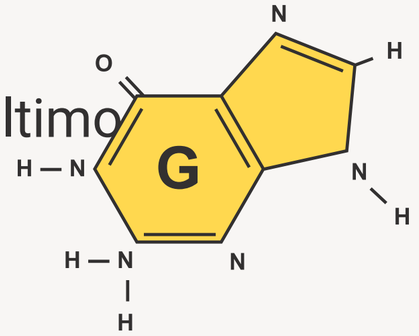
Approccio utilizzato

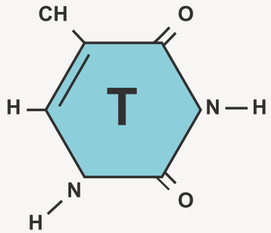
1. Studio del dominio del problema
2. Analisi del codice
3. Modifiche implementate
4. Testing
5. Sviluppi futuri

DOMINIO DEL PROBLEMA



La bioinformatica ha visto una rapida crescita grazie ai progressi nei metodi di sequenziamento e assemblaggio del genoma però quest’ultimo rimane ancora una sfida computazionale molto complessa.

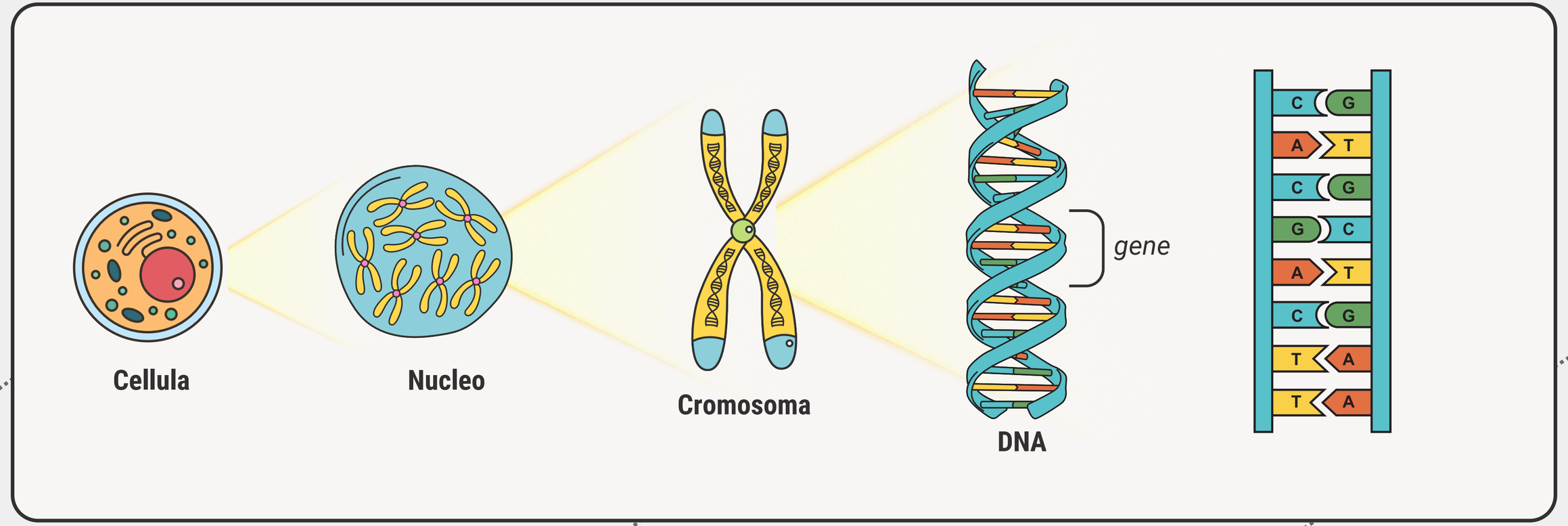




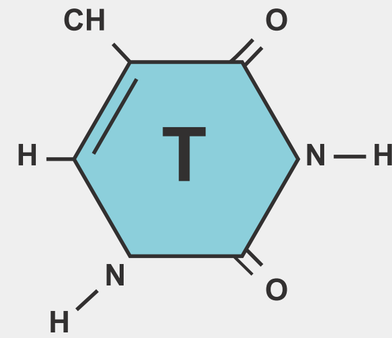
Il nostro obiettivo è combinare le tecniche di machine learning e la Fattorizzazione di Lyndon (CFL) per migliorare l'assemblaggio del genoma

COS'È UN GENOMA?

Il genoma è il materiale genetico completo di un organismo. Questo è composto da DNA che trasporta le informazioni genetiche degli organismi, comprese sequenze codificanti e non codificanti.



INDIVIDUAZIONE MARCATORI



Marcatori

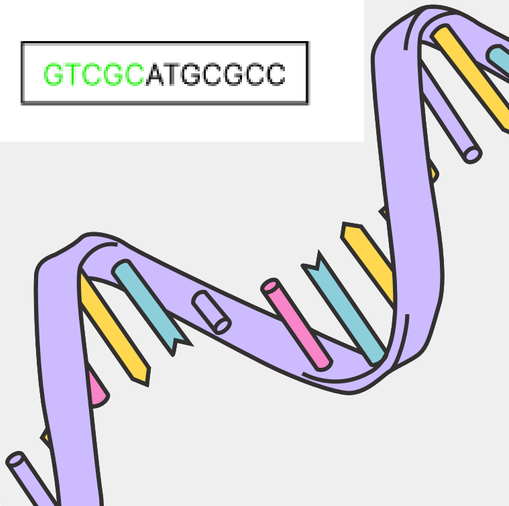
AAATT | TCCCT | GTCGC | TTTTC

Lettura

TCATATCCCTAGAGTGCAATAGCTGAGTGAGTAGCCGTAGGTTCTGCGCGATGCAGTGTCCCTGAATAATCCAAACAACCTCGCCGCGGTCGCATGCGCC

Applicazione Marcatori

TCATA | TCCCTAGAGTGCAATAGCTGAGTGAGTAGCCGTAGGTTCTGCGCGATGCAGTG | TCCCTGAATAATCCAAACAACCTCGCCGCG | GTCGCATGCGCC



FATTORIZZAZIONE DI LYNDON

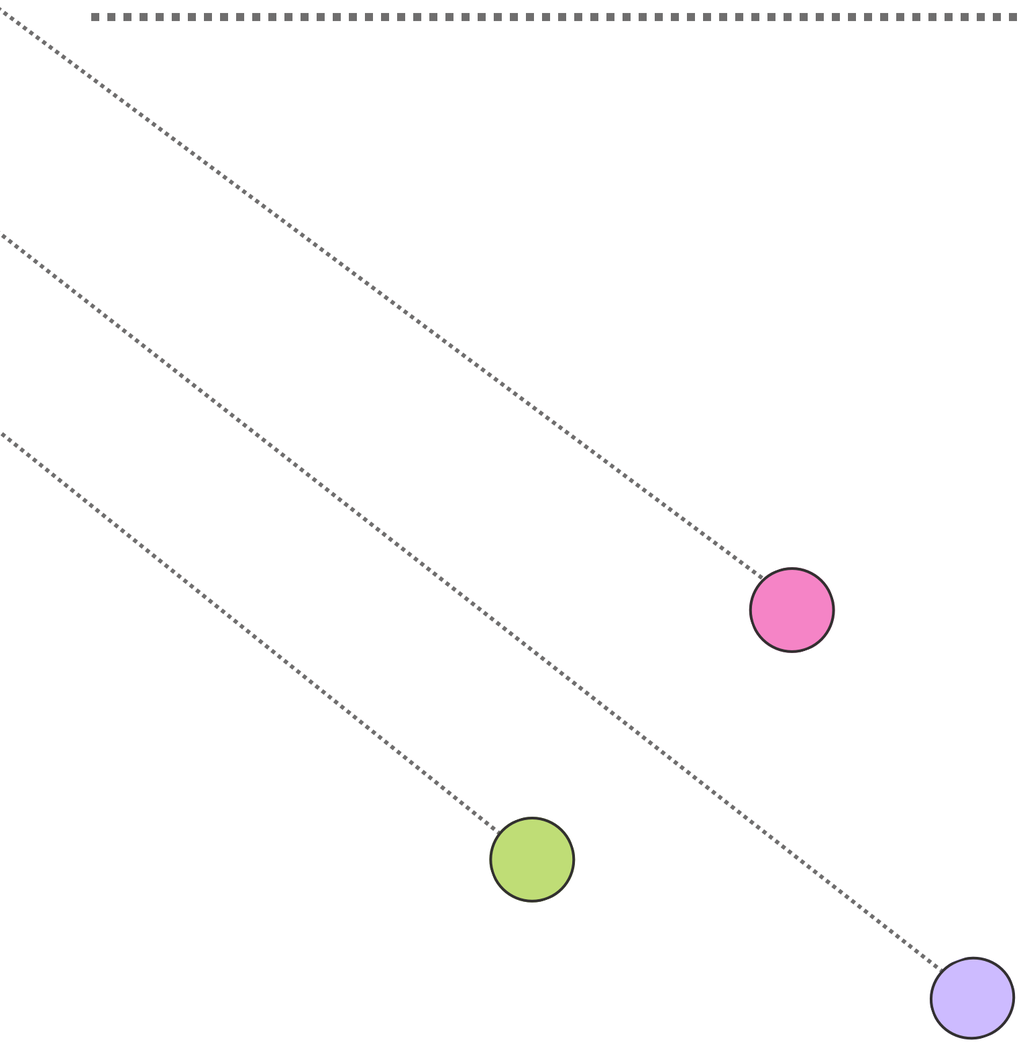
La Fattorizzazione di Lyndon diretta di una sequenza genera una sequenza di parole di Lyndon ⟨f1, f2, . . . , fn⟩, dove ogni fi è una parola di Lyndon. Questo processo è essenziale per scomporre una sequenza in elementi fondamentali che conservano le proprietà di Lyndon, fornendo una rappresentazione compatta e unica della sequenza

Questo tipo di fattorizzazione permette di ottenere una rappresentazione ordinata e standardizzata delle sequenze, facilitando la loro manipolazione.

ALGORITMI GENETICI

Gli algoritmi genetici (GA) rappresentano una classe di algoritmi di ottimizzazione e ricerca ispirati ai processi di selezione naturale e genetica. Introdotti da John Holland negli anni ’70, questi algoritmi simulano l’evoluzione delle specie per trovare soluzioni ottimali o quasi ottimali a problemi complessi. Gli GA utilizzano popolazioni di individui, rappresentati da cromosomi, che evolvono nel tempo attraverso operazioni di selezione, crossover e mutazione

ALGORITMI GENETICI



```
def run_ga(self, env, population,iterazioni):
    pop_fitness = self._evaluatePopulation(population)

    for generation in range(iterazioni):
        population = sorted(pop_fitness, key=lambda x: x[0],
           reverse=True)

        best_ind = population[0]

        if len(population) < 2:
            return best_ind
        else:
            lenPop = len(population) // 2
            population = population[:lenPop]

        # Crossover
        for i in range(0, len(population) - 1, 2):
            if np.random.rand() < self.crossover_prob:
                if i < len(population) - 1:
                    child1, child2 = self._crossover(population[i
                       ], population[i + 1])
                    population.append(child1)
                    population.append(child2)

        # Mutation
        for i in range(len(population)):
            if np.random.rand() < self.mutation_prob:
                temp = self._mutation(population[i])
                population[i] = temp

        if best_ind not in population:
            population.append(copy.deepcopy(best_ind))

        pop_fitness = self._evaluatePopulation(population, gen)

    return best_ind
```

TESTING PRELIMINARI

Abbiamo testato l'algoritmo genetico sui seguenti parametri, **(Numero_Marcatori e Dim_Marcatori),** ottenendo i seguenti risultati:

Per quanto riguarda Test_6Marks_8Dim:

NumIndividui: 400, NumIterazioni: 400, Distanza di Levenshtein: 809
NumIndividui: 400, NumIterazioni: 400, Distanza di Levenshtein: 837
NumIndividui: 800, NumIterazioni: 800, Distanza di Levenshtein: 848
NumIndividui: 500, NumIterazioni: 500, Distanza di Levenshtein: 861
NumIndividui: 400, NumIterazioni: 400, Distanza di Levenshtein: 866

Per quanto riguarda Test_8Marks_7Dim:

NumIndividui: 300, NumIterazioni: 300, Distanza di Levenshtein: 818
NumIndividui: 400, NumIterazioni: 400, Distanza di Levenshtein: 849
NumIndividui: 800, NumIterazioni: 800, Distanza di Levenshtein: 854
NumIndividui: 400, NumIterazioni: 400, Distanza di Levenshtein: 885
NumIndividui: 800, NumIterazioni: 800, Distanza di Levenshtein: 885

Per quanto riguarda Test_5Marks_7Dim:

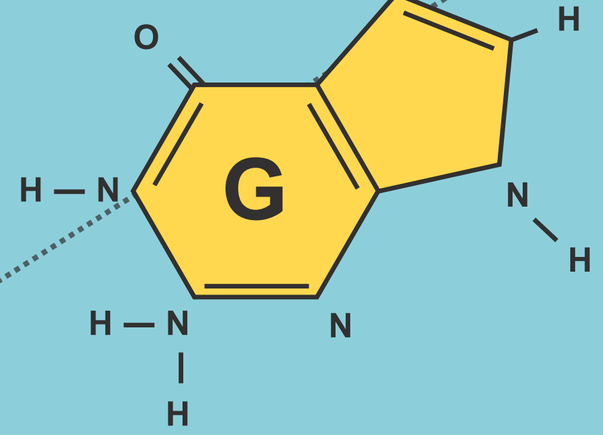
NumIndividui: 200, NumIterazioni: 200, Distanza di Levenshtein dati: 827
NumIndividui: 400, NumIterazioni: 400, Distanza di Levenshtein dati: 836
NumIndividui: 500, NumIterazioni: 500, Distanza di Levenshtein dati: 849
NumIndividui: 500, NumIterazioni: 500, Distanza di Levenshtein dati: 852
NumIndividui: 300, NumIterazioni: 300, Distanza di Levenshtein dati: 856

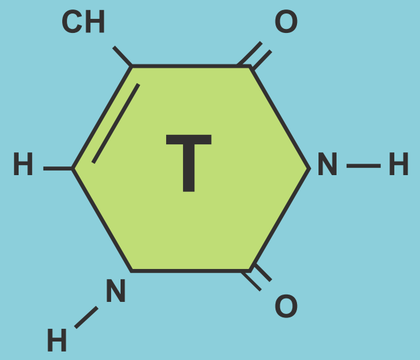
CONCLUSIONI E SVILUPPI FUTURI

Proposte:

Crediamo che l’approccio che abbiamo utilizzato può migliorare le tecniche di assemblaggio del genoma esistente, andando a soffermarsi sui possibili miglioramenti, tra cui:

- Una soluzione per i falsi positivi;
- Miglior gestione dei marcatori;





Grazie per l’attenzione