UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

SYLLABUS DEL CORSO

Elementi di Bioinformatica

2021-3-E3101Q116

Obiettivi

Alla fine del corso lo studente avrà compreso le principali problematiche computazionali e tecniche algoritmiche in bioinformatica. Sarà in grado di scrivere programmi di modeste dimensioni per risolvere problemi in bioinformatica utilizzando anche dati presenti su basi di dati pubbliche

Conoscenza e comprensione

Questo insegnamento fornisce le conoscenze basilari e capacità di comprensione relativamente a:

- Algoritmi su sequenze biologiche.

- Strutture dati per l'indicizzazione di sequenze
- Algoritmi per la ricostruzione di storie evolutive

- Programmazione in C in campo bioinformatico
- Shell di Unix
- Controllo di versione
- Programmazione in Pyhton in campo bioinformatico
- Formati di file usati in bioinformatica

Capacità di applicare conoscenza e comprensione

Alla fine dell'insegnamento gli studenti saranno in grado di:

- Comprendere quali algoritmi e strutture dati utilizzare per affrontare alcuni problemi in bioinformatica
- Scrivere programmi efficienti in C in campo bioinformatico
- Scrivere programmi in Python in campo bioinformatico

Contenuti sintetici

Principali problemi e algoritmi in bioinformatica. Pattern Matching. Allineamento di sequenze. Sequenziamento di DNA. Storie evolutive. Gestione di dati e basi di dati biologici

Programma esteso

1. Algoritmi e strutture dati per il pattern matching. Algoritmi di Karp-Rabin, Dömölki.

2. Suffix tree e suffix array: gestione, pattern matching e loro utilizzo per la ricerca della sottostringa più lunga.

3. Allineamento di due sequenze. Allineamento globale, locale, con banda. Costi di gap lineare e generico. Allineamento multiplo.

4. Sequenziamento di DNA. Grafi di overlap e grafi di de Bruijn.

5. Alberi evolutivi. Modelli a partire da caratteri: algoritmo di Gusfield per la filogenesi perfetta. Modelli a partire da distanze: distanze ultrametriche e distanze additive. Algorithmi UPGMA e Neighbor Joining. Cenni di ricostruzione tramite massima verosimiglianza
6. Genotipi e Aplotipi. Distinzione fra singolo individuo e pedigree.

7. Formati di file di dati biologici
8. Banche dati biologiche
9. Metodologie di sviluppo software open source per la bioinformatica
10. Cenni di shell
11. Cenni di git

Prerequisiti

Algoritmi e strutture dati; Linguaggi di programmazione;

Modalità didattica

Lezioni in aula e attività di laboratorio. Utilizzo della piattaforma di e-learning per integrare lo studio individuale

tramite arricchimento delle attività in aula e per autovalutazioni in itinere del livello di preparazione ottenuto. Durante l'emergenza le lezioni si tengono tramite videolezioni sincrone.

Il corso è erogato in Italiano.

Materiale didattico

Il libro di testo seguito per quasi tutte le lezioni è “Algorithms on Strings, Trees and Sequences”, di Daniel Gusfield, Cambridge Univ. Press. La biblioteca tiene alcune copie del libro di testo, anche come ebook.

Il libro "An Introduction to Bioinformatics Algorithms" di N. Jones, P. Pevzner viene usato esclusivamente come approfondimento per la parte di ricostruzione di filogenesi e di sequenziamento.

Il libro "Theoretical Evolutionary Genetics" di J. Felsenstein viene usato esclusivamente come approfondimento per la parte di ricostruzione di filogenesi. Un libro analogo è Population and Quantitative Genetics by Graham Coop.

Periodo di erogazione dell'insegnamento

primo semestre

Modalità di verifica del profitto e valutazione

La verifica dell'apprendimento consiste di una prova scritta e di una parte progettuale.

La prova scritta è individuale, basata su domande a risposta aperte relativa alle nozioni presentate nel corso relative ai contenuti di natura algoritmica. La prova scritta dura un'ora e contiene 4 domande, ma bisogna rispondere solo a 3 delle 4 domande.

La parte progettuale può essere superata in due modi:

1. project work da svolgere individualmente o in piccoli gruppi (max 3 persone), riguardanti gli aspetti di programmazione. Ogni studente decide se svolgerlo in Python o C, mentre il testo viene deciso dai docenti. I gruppi sono formati autonomamente dagli studenti. Ogni studente (o gruppo) riceve un testo personalizzato che viene comunicato al termine del corso. Il progetto viene poi discusso dopo avere superato la prova scritta.
2. svolgendo con successo tre esercizi in itinere durante il corso. Gli esercizi devono essere svolti individualmente e consistono nello scrivere programmi in Python (2 esercizi) e C (1 esercizio) entro delle scadenze fissate. Per svolgere con successo l'esercizio, il programma deve superare alcuni test che possono includere un controllo su tempi di calcolo o occupazione di memoria. Chi supera tutte e tre le prove riceve una valutazione di 30/30 per la parte progettuale. Non vengono riconosciuti superamenti parziali (ad esempio, avere svolto 2 esercizi su 3).

La valutazione finale viene ottenuta tramite media pesata delle votazioni ottenute nelle due parti, con peso 50% per la prova scritta e 50% per il project work, ma entrambe le parti devono avere valutazione positiva.

Per la parte scritta non sono previste prove in itinere.

Orario di ricevimento

Su appuntamento per email. Durante l'emergenza il ricevimento viene garantito e gestito tramite videoconferenza.