UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

COURSE SYLLABUS

Introduction To Bioinformatics

2425-3-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 Python 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
- Shell di Unix
- 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 in Python in campo bioinformatico
- Costruire semplici pipeline che integrano Python e la shell Unix

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. Metodologie di sviluppo software open source per la bioinformatica
9. Cenni di shell
10. Introduzione a Python.
11. Pandas.
12. Biopython

Prerequisiti

Complessità in tempo e spazio.
Strutture dati fondamentali: liste, array, alberi di ricerca, dizionari
Algoritmi di ordinamento: radix sort, merge sort
Gerarchia delle memorie
Capacità di scrivere un breve programma in qualunque linguaggio di programmazione

Modalità didattica

Lezioni in aula e attività di laboratorio. Viene utilizzata la 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.

Tutte le attività sono tenute in presenza e non vengono registrate nè trasmesse in streaming. L'insegnamento è erogato in Italiano. Le attività previste sono:

- 14 lezioni da 2 ore svolte in modalità erogativa nella parte iniziale ed in modalità interattiva nella parte successiva
- 24 attività di laboratorio da 2 ore svolte in modalità erogativa nella parte iniziale ed in modalità interattiva nella parte successiva

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.

Per quanto riguarda Python fare riferimento a Pensare in Python di A. B. Downey (versione inglese Think Python).

Per la parte relativa a Pandas il libro di riferimento è Python Data Science Handbook di VanderPlas

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 prova scritta viene valutata in base alla correttezza e completezza delle risposte, e sulla capacità di individuare gli elementi essenziali di un argomento.

La parte progettuale consiste nella scrittura di un programma in Python e viene svolta individualmente, con successiva presentazione del lavoro fatto. Il testo del progetto è lo stesso per tutti gli studenti ed è valido per una sessione.

La discussione del progetto avviene dopo avere superato la prova scritta ed entro un anno da tale superamento.

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 progetto, ma entrambe le parti devono avere valutazione positiva.
Si ricorda che è necessario essere iscritti alle prove d'esame tramite segreterie online. Non ci saranno eccezioni al riguardo.

Non sono previste prove in itinere.

Orario di ricevimento

Il ricevimento è online e su appuntamento: https://www.unimib.it/gianluca-della-vedova

Sustainable Development Goals

SALUTE E BENESSERE