Informazioni

- Lezione di domani 23/12
- Simulazione d'esame… venerdi' 9/1/26, 10-13:30 in aula Taliercio
  - Apriro' una lista di iscrizione su moodle
- Didattica Integrativa (dr. Lorenzo Saccaro)
  - Giovedi' 8/1, 14:30 -16:30, H11
  - Venerdi' 9/1, 14:30 – 16:30 H11
    - Correzione simulazione d'esame
  - Lunedi' 12/1 orario?
    - 10:30-12:30 oppure solito slot 14:30-16:30

Vettori o liste sequenziali

La classe ArrayList

- La classe **ArrayList della libreria standard** (vettore, o lista sequenziale) consente di memorizzare una raccolta di oggetti
- Il comportamento è simile a quello di un array ma ci sono due vantaggi:
  - la dimensione di un vettore puo' variare
  - nella classe ArrayList esistono metodi per svolgere le tipiche operazioni su una collezione di oggetti (inserimento, rimozione, etc.)

La classe ArrayList

- Per creare un vettore di conti bancari è necessario creare un oggetto di tipo ArrayList

```
ArrayList<BankAccount> accounts = new ArrayList<BankAccount>
```

- La dimensione di un oggetto di tipo ArrayList è **inizialmente nulla**
- Il metodo **size()** restituisce la dimensione corrente del vettore

```
int dim = accounts.size(); // dim vale 0
```

Classi generiche

- La classe ArrayList è una ***classe generica***
  - ArrayList<T> contiene oggetti di tipo T
  - T è un tipo parametrico
  - inserendo al posto di T il nome di una classe si ottiene un vettore di oggetti della classe specificata

- Il tipo ArrayList<BankAccount> specifica un vettore di conti bancari

Il metodo add

- Per inserire un oggetto ***alla fine*** della sequenza di oggetti contenuti nel vettore si usa il metodo **add**
- La dimensione viene incrementata di 1 ad ogni invocazione di add

```
accounts.add(new BankAccount(1000));
accounts.add(new BankAccount());
dim = accounts.size(); //ora vale 2
```

Il metodo add

- E' anche possibile inserire un oggetto in una posizione ***intermedia*** con l'invocazione di **accounts.add(i,c)** che aggiunge l'oggetto **c** in posizione **i** spostando tutti gli ***elementi successivi*** ad i di una posizione

```
accounts.add(new BankAccount(500),1);
dim = accounts.size(); //ora vale 3
```

Il metodo remove

- Per eliminare l'oggetto che si trova ***in posizione i*** si usa il metodo **remove**
- La dimensione viene ***diminuita di 1*** ad ogni invocazione di remove

```
accounts.remove(1);     //viene eliminato l'elemento
                        //in posizione 1
dim = accounts.size(); //ora vale 2
```

- Tutti gli elementi che si trovano dopo l'elemento rimosso ***vengono spostati*** di una posizione all'indietro

Il metodo get

- Per ispezionare gli oggetti contenuti nel vettore si usa il metodo **get** che richiede come parametro esplicito l'indice del vettore

```
BankAccount acct = accounts.get(3);
```

restituisce l'elemento del vettore di posizione 3

- Accedere ad un elemento non esistente e' un errore frequente

```
BankAccount acct = accounts.get(3);
```

- L'indice valido di valore massimo e' **accounts.size()-1**

Il metodo set

- Per assegnare un nuovo valore ad un elemento di un vettore si usa il metodo **set**

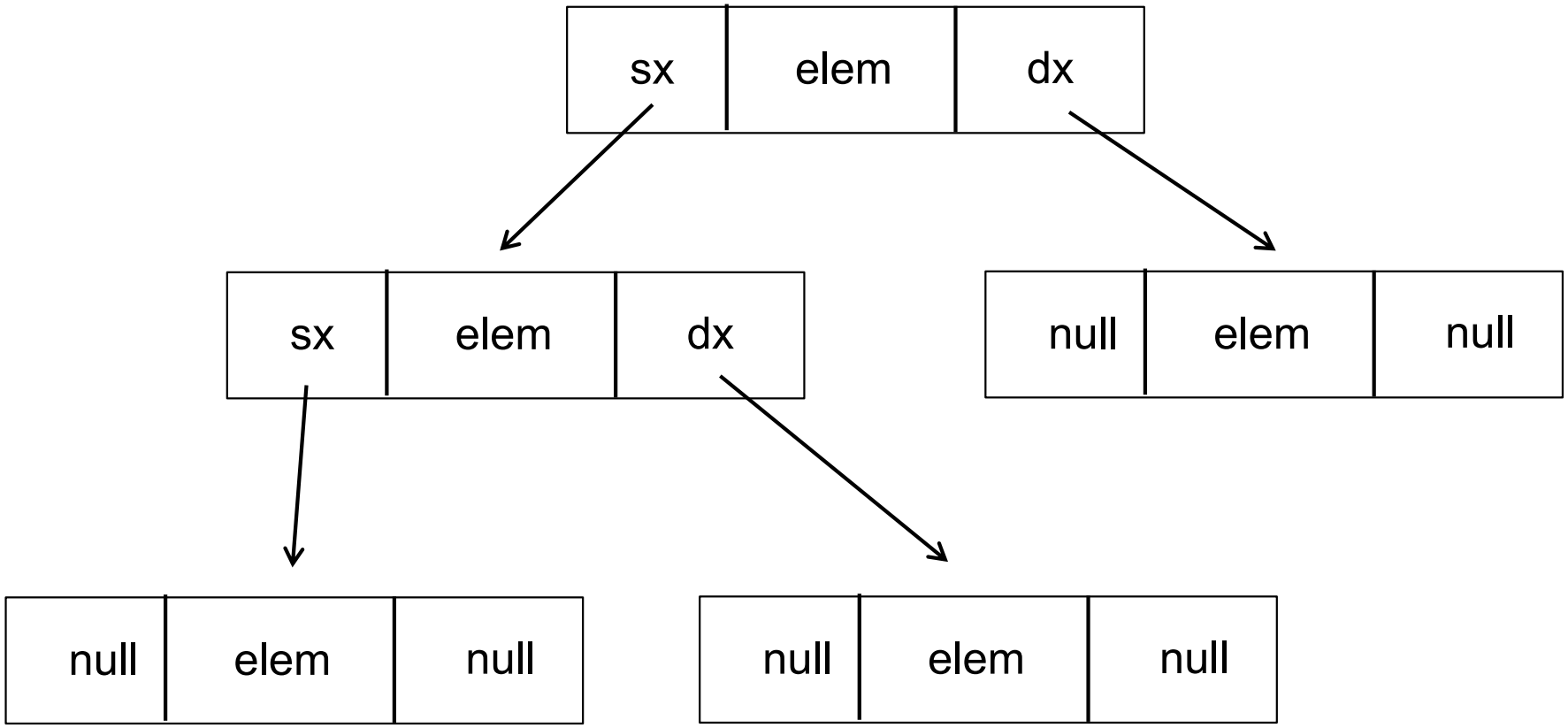
```
BankAccount acct = new BankAccount(2000);
accounts.set(2, acct);
```

inserisce nella posizione 2 del vettore accounts un riferimento al conto acct, ***sovrascrivendo*** l'eventuale valore memorizzato precedentemente

- Il metodo set puo' solo sovrascrivere ***elementi gia' esistenti*** nel vettore

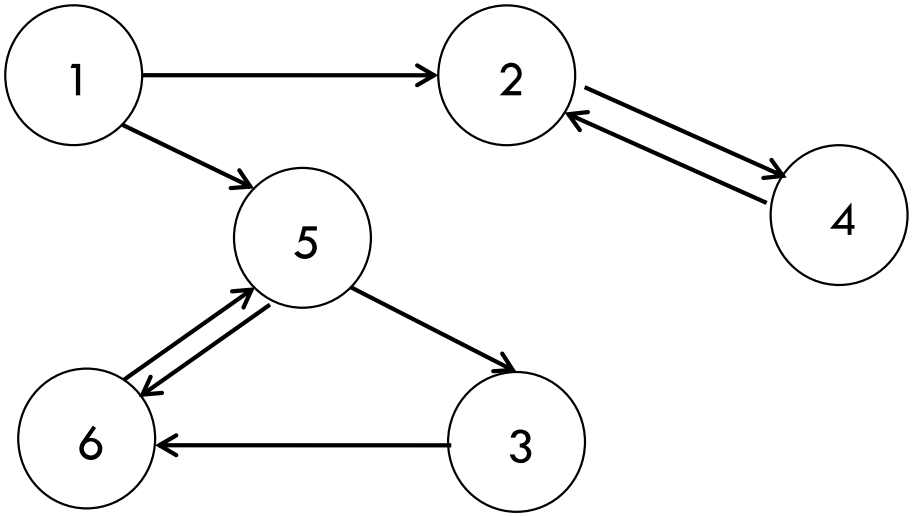
In futuro…

- Eliminerete il vincolo di un solo successore…
  - Qui abbiamo il vincolo di al piu' due successori: albero binario. Inserendo i dati in modo particolare si ottengono alberi binari di ricerca (molto utilizzati nei DB)

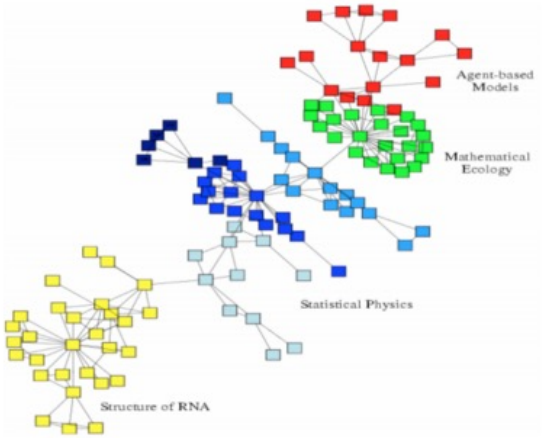


In futuro…

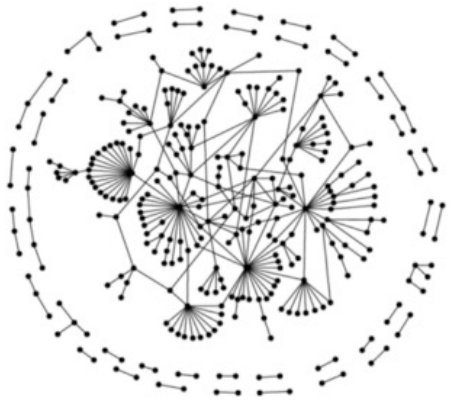
- E poi eliminerete anche il vincolo di avere un solo predecessore…
  - Grafo! Diretto/indiretto, semplice/multigrafo, connesso/non connesso, pesato/non pesato…



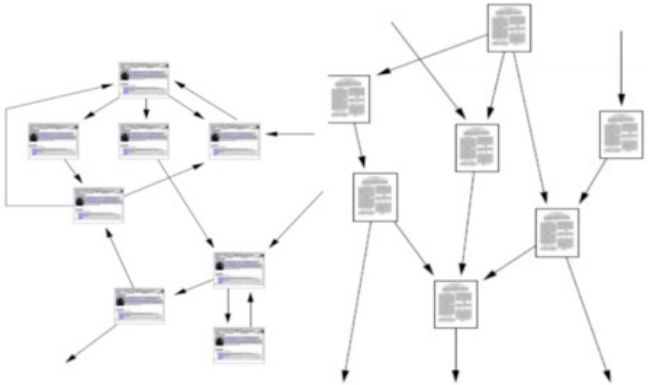
Social networks



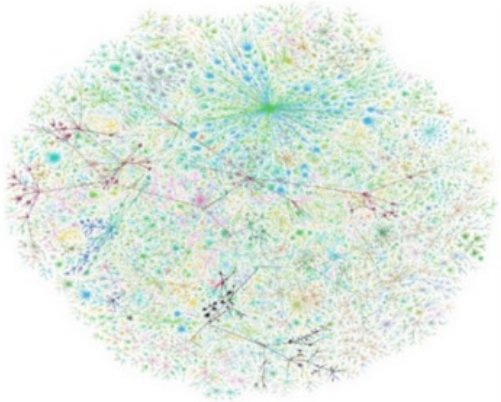
Economic networks



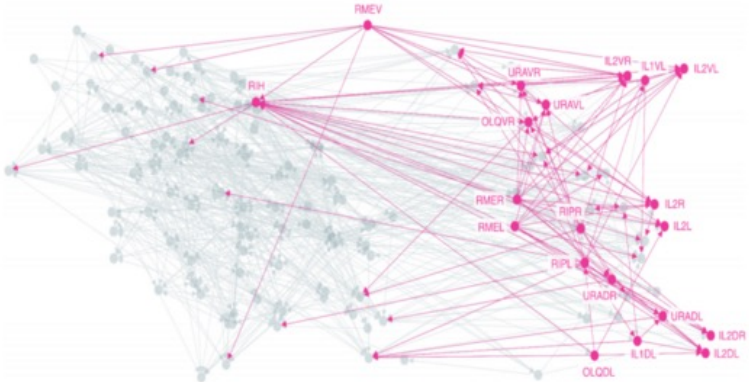
Biomedical networks



Information networks:
Web & citations



Internet



Networks of neurons

Immagine: William Hamilton [COMP551: Graph Representation Learning]

Riassunto Strutture Dati

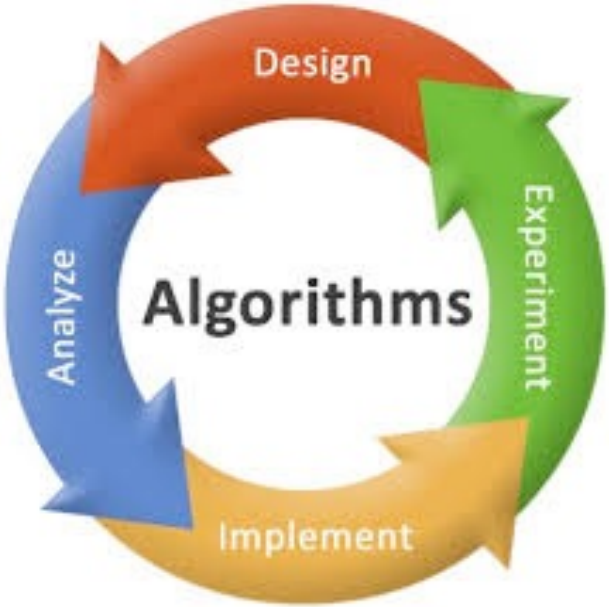
- Abstract Data Type (ADT)
  - Pile – Stack
  - Code – Queue
  - Coda Doppia – Deque
  - Mappa – Map
  - Multimappa – MultiMap (Dictionary)
  - Tabella – Table
  - Tabella Hash – Hashtable
  - Insiemi – Set
- Strutture dati lineari
  - Array
  - Liste concatenate (semplici e doppie) – LinkedList
  - ArrayList (NON per il corso)

Quiz autovalutazione 4 e 5 correzione

Importanza delle scelte progettuali

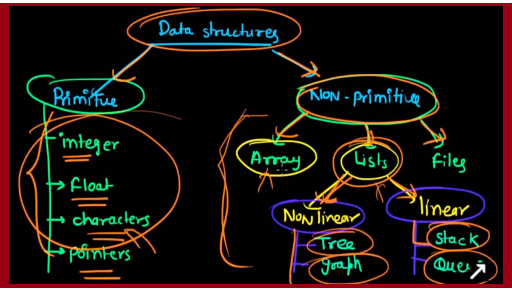
Le scelte progettuali

- Dato un problema
  - Progettazione e analisi di un algoritmo per la sua soluzione
  - Scelta del linguaggio di programmazione e implementazione dell'algoritmo
  - Verifica sperimentale della correttezza e delle prestazioni



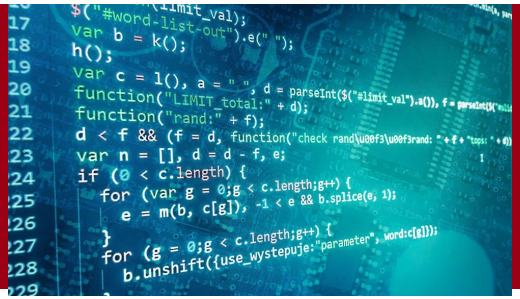
L'idea algoritmica

- Per uno stesso problema possono esistere molteplici approcci alla sua soluzione
- La soluzione che proponiamo ha un peso decisivo sulle performance del software che poi andremo a sviluppare
- La scelta di una particolare soluzione può dipendere dal contesto applicativo
  - Real time? Applicativi aziendali? Simulazioni scientifiche? Big data analysis? Sviluppo di librerie?



La struttura dati

- In generale un algoritmo può essere implementato utilizzando diverse strutture dati
- La scelta della struttura dati può influenzare pesantemente
  - le prestazioni dell'algoritmo
    - Utilizzo di memoria (primaria o secondaria)
    - Tempi di esecuzione
  - I tempi di sviluppo
    - Complessità nella progettazione, nell'implementazione o nell'utilizzo



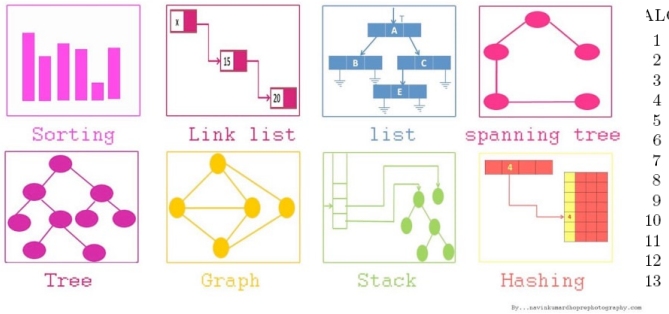
L'implementazione

- Scegliere il linguaggio di programmazione adatto alle nostre esigenze
  - Valutare flessibilità (portabilità/modularità), efficienza (compilato vs interpretato vs ibrido), librerie a disposizione
- Progettare e scrivere il codice in modo che sia
  - Corretto
  - Comprensibile
  - Efficiente
    - Un algoritmo implementato male ha performance al di sotto di quanto si possa effettivamente ottenere

Qualche esempio di applicazione

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

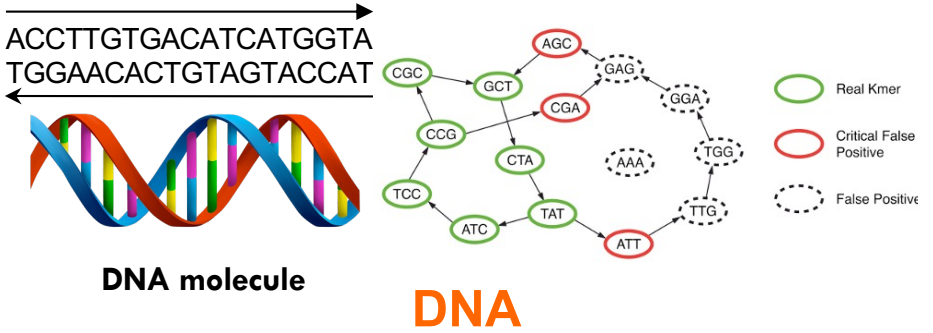
Bioinformatica



Data Structures

```
ALGORITHM_N2()
1  read input text string x, and string length m
2  compute row 1 with classic k-mismatch algorithm
3  copy row 0 in column 0 (the matrix is symmetric)
4  for i ← 1 to n − m − 1
5       do
6          for j ← i + 1 to n − m − 1
7               do
8                  M_{i,j} ← M_{i−1,j−1}
9                  if x_{i+m−1} ≠ x_{j+m−1}
10                    then M_{i,j} ← M_{i,j} + 1
11                 if x_{i−1} ≠ x_{j−1}
12                    then M_{i,j} ← M_{i,j} − 1
13                 M_{j,i} ← M_{i,j}
```

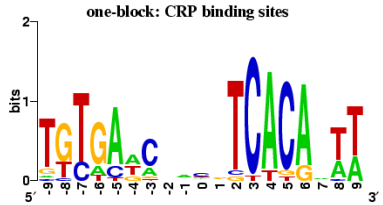
Algorithms



DNA molecule

DNA

Stringhe su A,C,G,T

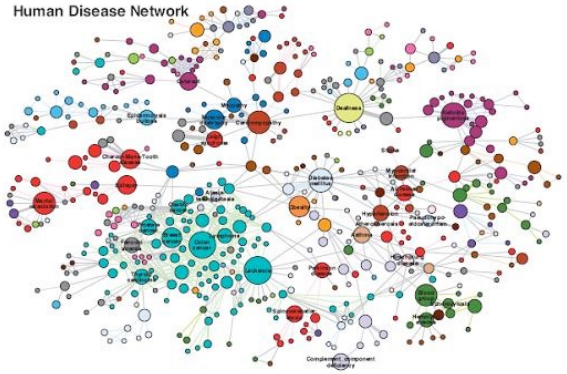


Models

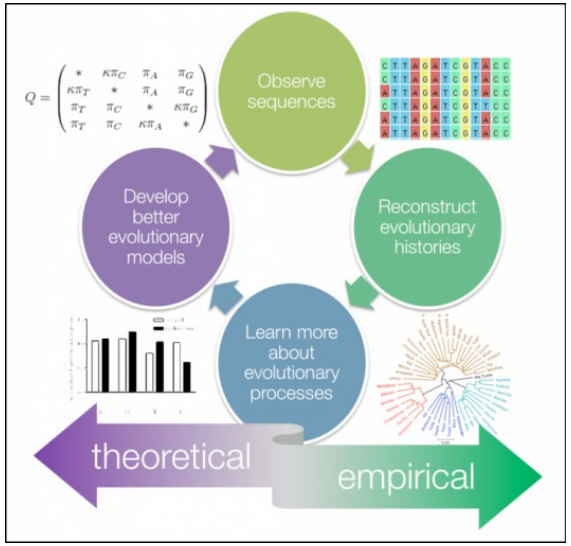
+

```
----PKKAVQLVLQMRD----AEKIANGLLNEARAMR---
----SSDVVSYVRDQLR----VQLACESLAQVALDRR---
----RQDVVRIVGEYLT----DQNAATHLIRHAVGNN---
----NEEIASLVIRWMD----DKNVATHLIRNALS-----
----DQEVVDLITSTVN----LKKATPQFVAEETIKF---
---VEQYWQDDFLPVLQ----VAEAVPRLIELANQVN---
-----KELFFEIITNSK----LKQAVFNLYRKSIENA---
LQLLSQKFVDNVVSLSK----PSVFAQEISKLTGKNY---
----YETVCDISRENKS----PMSAAEKMKDYAISYG---
----VDTVCDIARENST----PLRAAAELKDHAMAYG---
----YQTAVDIARTQRN----PMIAAQKLRDFAISYG---
----PELIVDVARECRS----LMKASQKLRDLAIAYG---
----KRTVIDVVRANRH----PLLASTKLRDYAIAYG---
----IDDLNSTFHNNRS----PIVVAKKIQDQLQSYD---
---HVENYEMWMKRKIG----PQEIADLIMEEVIRTK---
```

Pattern discovery
(indexing, data mining, string algorithms)



Interaction network
(machine learning, graph algorithms)

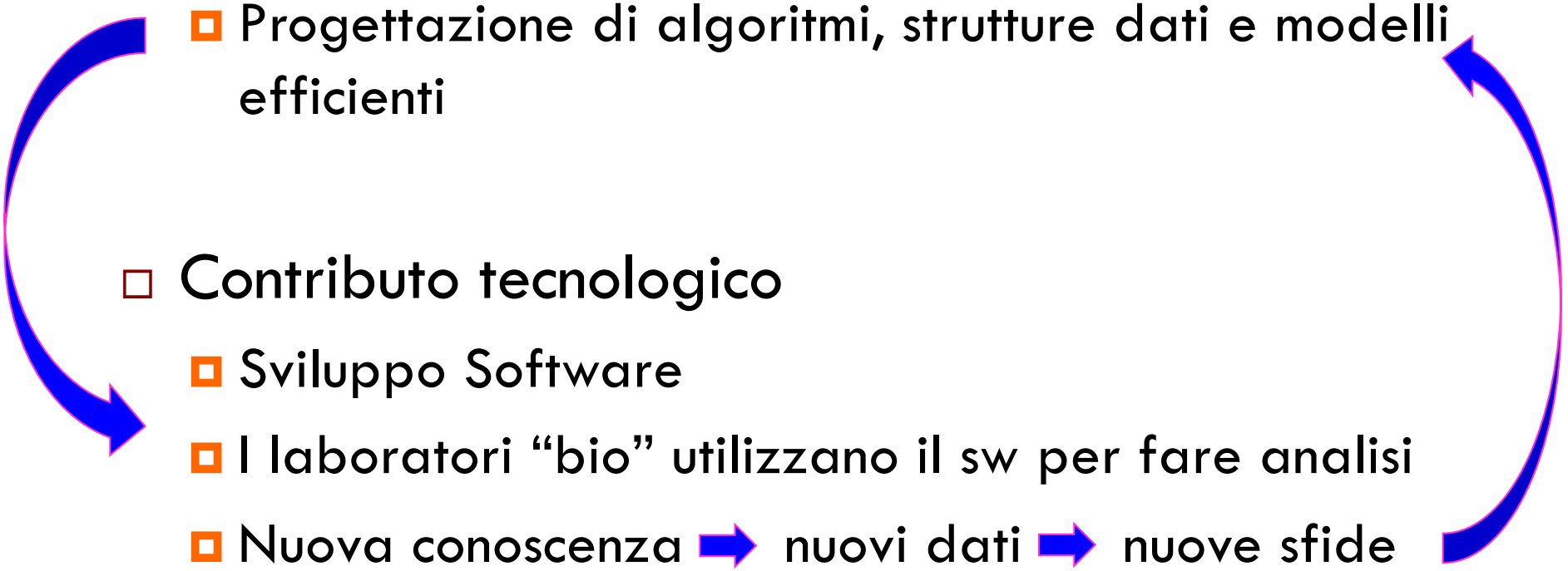


Phylogenomics
(sequence similarity, algorithms on trees)

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Dalla teoria all'applicazione e viceversa

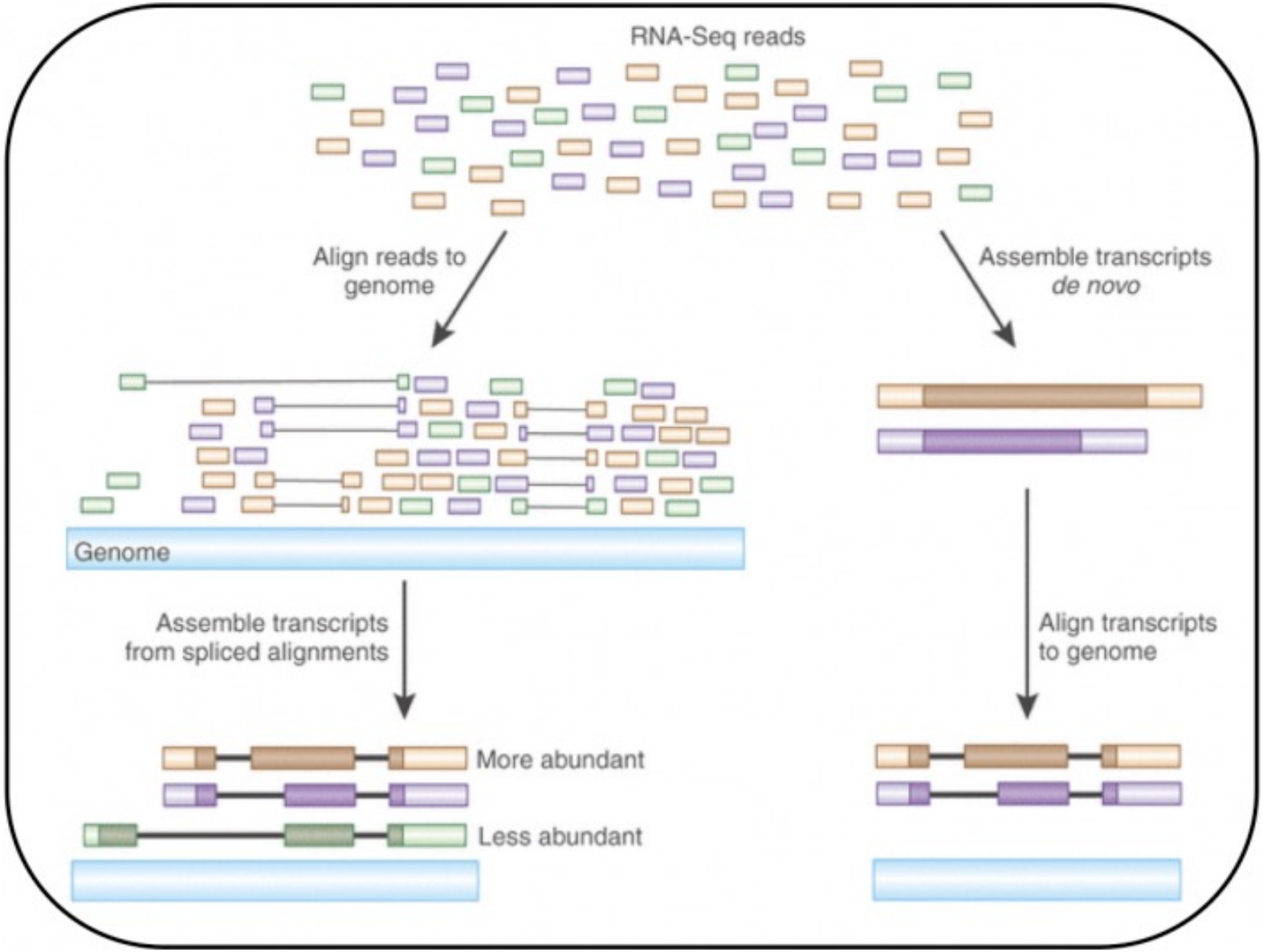
- Necessità di elaborare grandi quantità di dati e scarsa conoscenza dei contenuti => Sfide computazionali
  - Progettazione di algoritmi, strutture dati e modelli efficienti
- Contributo tecnologico
  - Sviluppo Software
  - I laboratori "bio" utilizzano il sw per fare analisi
  - Nuova conoscenza ➡ nuovi dati ➡ nuove sfide



Esempio 1: read mapping

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Read mapping



DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Burrows-Wheeler Transform & FM index

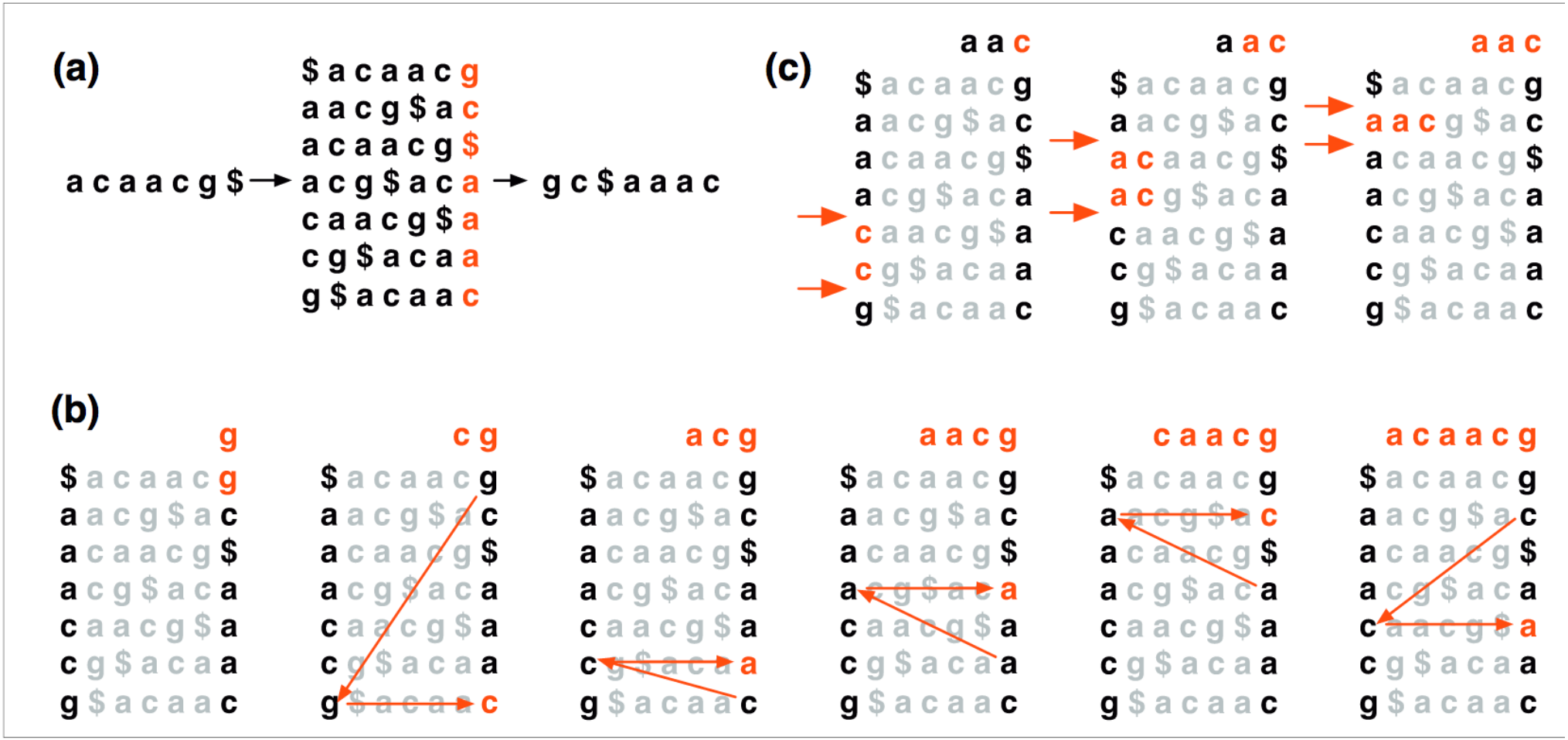


Figure 1
Burrows-Wheeler transform. **(a)** The Burrows-Wheeler matrix and transformation for 'acaacg'. **(b)** Steps taken by EXACTMATCH to identify the range of rows, and thus the set of reference suffixes, prefixed by 'aac'. **(c)** UNPERMUTE repeatedly applies the last first (LF) mapping to recover the original text (in red on the top line) from the Burrows-Wheeler transform (in black in the rightmost column).

Utilizzando strutture dati compresse…

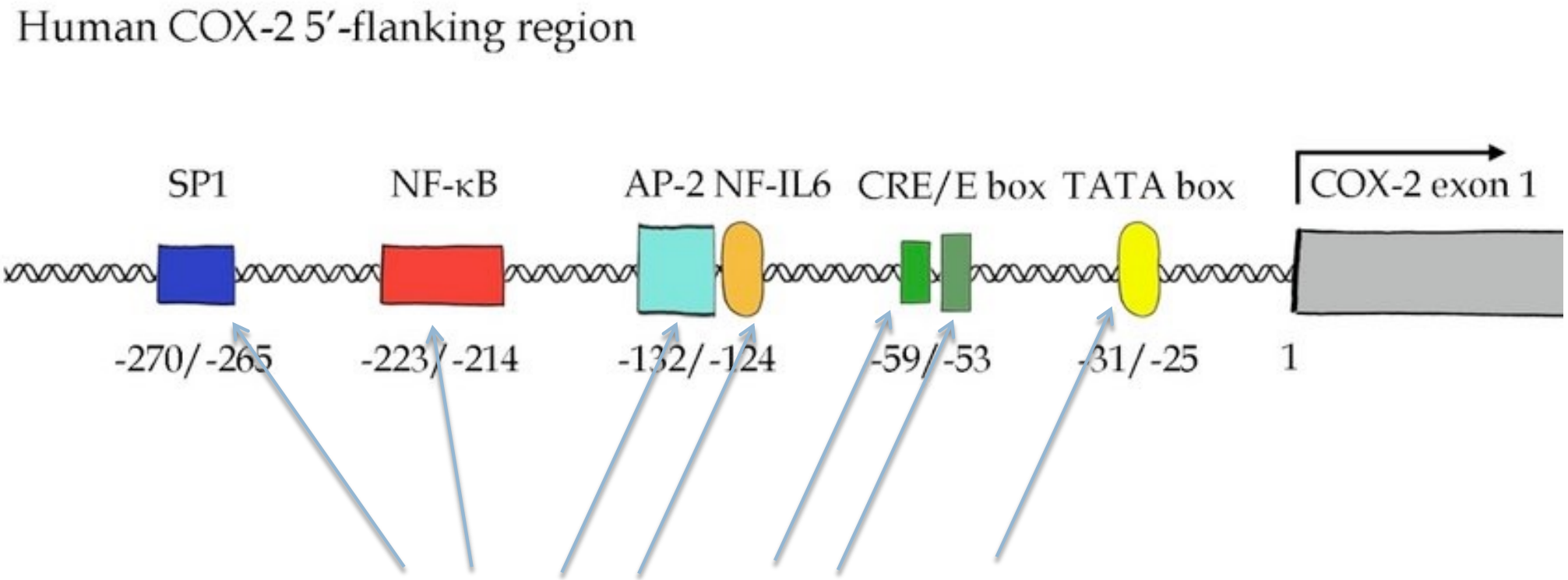
Bowtie alignment performance versus SOAP and Maq

	Platform	CPU time	Wall clock time	Reads mapped per hour (millions)
Bowtie -v 2	Server	15 m 7 s	15 m 41 s	33.8
SOAP		91 h 57 m 35 s	91 h 47 m 46 s	0.10
Bowtie	PC	16 m 41 s	17 m 57 s	29.5
Maq		17 h 46 m 35 s	17 h 53 m 7 s	0.49
Bowtie	Server	17 m 58 s	18 m 26 s	28.8
Maq		32 h 56 m 53 s	32 h 58 m 39 s	0.27

Esempio 2: weighted matching

CODICE
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Applicazione: regolazione genica (gli "interruttori" dei geni)



I segnali sono sottostringhe della regione che precede il gene!

Score di segmento

CIASCUN SEGNALE E' RAPPRESENTATO DA UNA MATRICE

	1	2	3	4	5	6
A	0.3	0.0	0.1	0.2	1.0	0.3
C	0.1	0.8	0.5	0.2	0.0	0.4
G	0.2	0.0	0.4	0.3	0.0	0.0
T	0.4	0.2	0.0	0.3	0.0	0.3

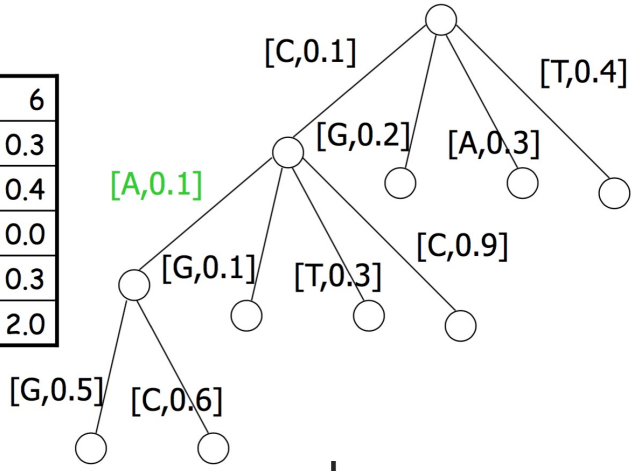
A T C C G T A C A C T G C

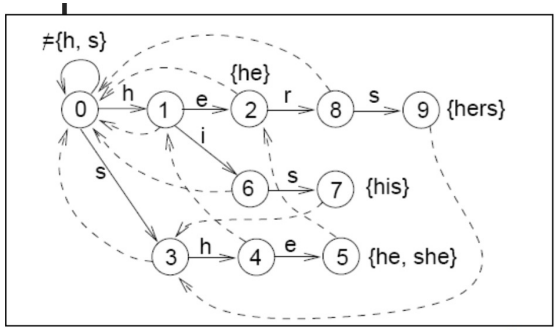
Score = 2.1

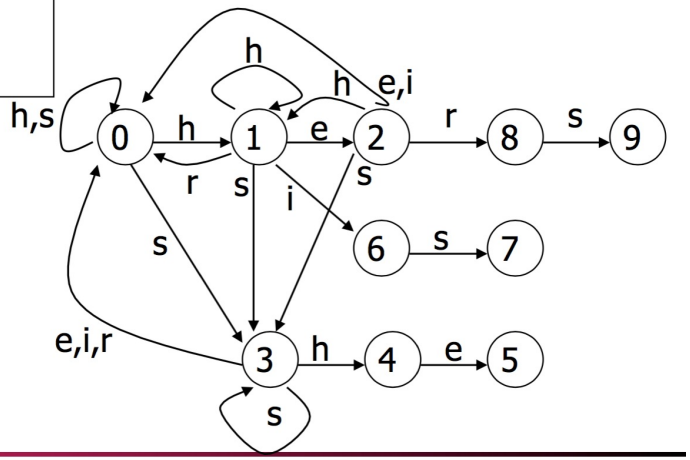
Moods: multi-pattern matching & look-ahead filtration &q-mer filtering

- Third position

	1	2	3	4	5	6
A	0.3	0.0	0.1	0.2	1.0	0.3
C	0.1	0.8	0.5	0.2	0.0	0.4
G	0.2	0.0	0.4	0.3	0.0	0.0
T	0.4	0.2	0.0	0.3	0.0	0.3
Pth	-1.0	-0.2	0.3	0.6	1.6	2.0







Time O(n)
Space depends on D and Σ

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Moods: non solo velocita'...

ALGODAN
Algorithmic Data Analysis

MOODS: Motif Occurrence Detection Suite

MOODS is a suite of algorithms for matching position weight matrices (PWM) against DNA sequences. It features advanced matrix matching algorithms implemented in C++ that can be used to scan hundreds of matrices against chromosome-sized sequences in few seconds.

MOODS can be used as standalone analysis tool or as a component in larger programs. It contains Python and Perl/BioPerl interfaces, and can thus be easily called from C++, Python and Perl programs.

MOODS is dual-licenced under GPL version 3 license and Biopython license.

	600k		Chr20	
p-value	10^{-6}	10^{-4}	10^{-6}	10^{-4}
MOODS:				
- Naive algorithm	6.5 s	7.3 s	689 s	782 s
- Permutated lookahead	3.8 s	6.3 s	405 s	677 s
- MLF	0.4 s	1.1 s	16.0 s	117 s
TFBS	20.4 s	53.1 s	–	–
Motility	103 s	103 s	180 min	181 min
Biopython	42 min	41 min	–	–
matches	952	$7.3 \cdot 10^4$	$1.1 \cdot 10^5$	$6.7 \cdot 10^6$

Citato in:
Nature, Cell, Nature Biotechnology, BMC Genomics, BMC Cancer, Nature Protocols, Genome Biology, Microbiology, etc.

L'analisi per input piu' grandi (62Mb) non e' possibile!

C.Pizzi, P.Rastas, E.Ukkonen
Fast Search Algorithms for Position Specific Scoring Matrices
In *Proc. of the 1st Conference on Bioinformatics Research and Development (BIRD 07), Berlin, Germany, March 2007, LNCS/LCBI 4414 pp 239--250*

J. Korhonen, P. Martinmäki, C. Pizzi, P. Rastas and E. Ukkonen. MOODS: fast search for position weight matrix matches in DNA sequences. Bioinformatics 25(23), pages 3181-3182. (2009)

C. Pizzi, P. Rastas and E. Ukkonen: Finding Significant Matches of Position Weight Matrices in Linear Time. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 8(1), pages 69 - 79. (2011)

Take home message

Gli algoritmi e le strutture dati contano!

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Messaggio promozionale!

Tra un paio d'anni…

- Al terzo anno avete la possibilita' di frequentare alcuni corsi a vostra scelta

- **Elementi di Bioinformatica**
  - Bellissimo corso introduttivo alla disciplina!
  - Non e' un corso di biologia, e' un corso di informatica!
    - Possibilita' di vedere "in azione" in un ambito specifico quanto imparato in altri corsi
    - Conoscere problemi computazionali dominio-specifici e le loro soluzioni
  - Ottimo feedback da parte degli studenti

Tra un paio d'anni…

- Tesi triennali
  - disponibili su argomenti relative a strutture dati e algoritmi efficienti per l'analisi di sequenze
  - Studio bibliografico o implementazione sw o benchmarking di tool per vari tipi di analisi
  - Applicazione in ambito Bioinformatico (ma non solo)
- Disponibilita' a essere referente per tirocini per chi fa l'indirizzo Applicativo

Tra qualche ulteriore anno…

- Tesi magistrali
  - Progettazione e sviluppo di soluzioni efficienti per problemi computazionali riguardanti analisi di sequenza con applicazione prevalentemente alla bioinformatica
    - Tecniche di combinatorial pattern matching
    - Tecniche di machine e deep learning
- Disponibilita' a essere referente per tirocini e per Research Training

CODICE
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Tra qualche ulteriore anno…
Learning from Sequences

- **Learning from Sequences** è un insegnamento di laurea magistrale dedicato all'analisi e all'apprendimento da grandi quantità di dati sequenziali. Il corso copre temi chiave come **compressione dei dati, strutture dati compresse, tecniche di sketching, encoding efficienti, analisi di serie temporali** e **metodi di machine e deep learning per sequenze**. L'obiettivo è fornire gli strumenti per **scalare l'analisi a miliardi di simboli** di sequenze testuali o numeriche lunghe e rumorose, bilanciando accuratezza, memoria e tempo di esecuzione.

- I casi di studio sono rivolti principalmente a dati genomici e di ambito health, ma **non sono richieste conoscenze di biologia**: il focus è sull'**ingegneria degli algoritmi**, sull'uso consapevole delle risorse computazionali e sulla progettazione di soluzioni applicabili a grandi sistemi reali. Un corso ideale per chi vuole rafforzare le proprie competenze in **algoritmi, data structures e AI**, applicandole a uno dei contesti più sfidanti dei big data moderni.

Esami

Iscrizione

- **Iscrizione su uniweb Obbligatoria**
  - C'è una scadenza: siamo molto rigorosi per motivi organizzativi (e non solo)
  - Attenzione: La data/ora indicata nella lista non e' necessariamente il giorno/ora dell'esame: abbiamo a disposizione 3 giorni, a liste chiuse stabiliamo i turni e ve li comunichiamo
  - Ci si iscrive solo alla lista del proprio canale

A cosa serve l'esame

- L'esame è un momento di verifica della propria preparazione

- Le competenze che il corso prevede vengano acquisite vi serviranno
  - Per affrontare con i giusti prerequisiti i prossimi corsi
  - Per affrontare con competenza i progetti delle tesi di laurea
  - Nel mondo del lavoro

Quanto grave e' copiare?

- Tanto. Copiatura e sostituzione di persona sono atti gravi secondo la legge
- Ripercussioni civili e penali
- Ripercussioni disciplinari

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Alcune Regole d'esame (nel dubbio, chiedete sempre!):

1. E' vietato tenere con sé al banco ed utilizzare strumenti atti a comunicare (cellulari, tablet, videocamere, microfoni o quant'altro). Tali strumenti vanni SPENTI, e RIPOSTI nello zaino/borsa, che va tenuto negli appositi spazi
2. E' vietato consultare: libri, appunti etc.
3. E' vietato l'utilizzo di strumenti di AI
4. E' vietato consultare altri studenti
5. Al banco si può tenere solamente: documento d'identità, penne, foglio consegnato dai docenti (si può usare il retro per fare conti durante il quiz), NO CALCOLATRICE
6. Il foglio va riconsegnati alla fine della prova
7. **La docente si riserva la facoltà di convocare studenti ad un esame orale nel caso di dubbi sull'integrità dell'esame scritto**

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Comportamenti illeciti:

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Coloro che imbrogliano agli esami danneggiano in primo luogo i compagni/compagne che si comportano correttamente.

La **violazione** delle regole d'esame comporta **l'esclusione immediata dalla prova ed il suo annullamento.**

Nei **casi più gravi** la violazione può portare al **deferimento alla Commissione disciplinare** di Ateneo di cui all'art. 29 del "Regolamento delle carriere degli studenti".

L'accertamento da parte della Commissione della violazione contestata può portare alla **sospensione da tutte le attività didattiche** (esami, lezioni, laboratori, tesi, tirocini etc.) fino ad un massimo di 3 anni.

Purtroppo nel 2025, 3 studenti DEI sono stati sospesi per 5 mesi, 1 studente per 10 mesi a causa di comportamenti illeciti durante le prove d'esame.

La sospensione può anche comportare la **perdita dei benefici** associati allo status di studente (borse di studio, permessi di soggiorno etc.).

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Modalita' d'esame

- L'esame consiste in due prove da sostenere **nello stesso appello**
  - prova di teoria: quiz a risposta multipla
    - N domande, ~N minuti
  - prova di programmazione
    - Alcuni esercizi di programmazione
  - E' possibile che all'interno di una delle due prove si inseriscano una o due domande a risposta aperta su concetti difficilmente mappabili in domande a risposta multipla

Modalita' d'esame

- Entrambe le parti devono essere sufficienti
  - Si è ammessi alla parte di programmazione se il questionario è sufficiente
- Il candidato può sostenere l'esame in ciascun appello, ma
  - Tutte le prove devono essere sostenute nello stesso appello

CODICE
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Quiz domande a risposta multipla

- Come i quiz di autovalutazione
- Su tutto il programma
- Una trentina di domande
- Risposta esatta 1 punto

Programmazione

- Un esercizio di realizzazione di una classe eseguibile che manipoli array o stringhe o numeri
  - Possibile che sia esplicitamente richiesto di realizzare un metodo statico ricorsivo
  - Possibile che sia richiesto di leggere e/o scrivere dati e risultati a terminale o su file
  - Possibile che sia richiesto di gestire eventuali eccezioni

Programmazione

- Realizzazione di un ADT
  - Pila, coda, coda doppia, mappa, multimappa, insieme, tabella, set etc
  - Combinazioni di ADT
  - Variazioni (sottoclassi) secondo specifiche
- Scrivere un tester che acquisisca un input di prova da file e utilizzando la struttura dati implementata riporti in uscita il risultato delle elaborazioni richieste

Programmazione

- L'obiettivo principale della prova di programmazione è quello di realizzare un programma
  - **che rispetti le specifiche** (senza fare cose in più...)
  - **che compili**
  - **che funzioni**

Programmazione - consigli

- concentrarsi prima sulla realizzazione di un ***prototipo funzionante nel rispetto delle specifiche***
  - di solito, la cosa migliore è seguire le specifiche passo per passo
  - un programma funzionante che non rispetti le specifiche non serve a niente...
  - **Compilare spesso per identificare subito eventuali errori**

Programmazione - consigli

- verificare accuratamente il corretto funzionamento del programma realizzato, soprattutto in ***eventuali situazioni limite***
  - per velocizzare il collaudo, si consiglia di preparare alcuni file di input da usare con la redirezione, prevedendo l'output corretto (di solito io fornisco un tester proprio a questo scopo)

Programmazione

- Il candidato deve produrre un elaborato che:
  1. **rispetti le specifiche**
  2. **non generi errori in compilazione**

Obiettivi minimi

- che, *possibilmente*,
  3. *non generi errori in esecuzione*
  4. *esegua correttamente, secondo le specifiche*
- In subordine, se rimane tempo, che
  5. sia ottimo dal punto di vista della complessità temporale dei metodi realizzati
  6. sia propriamente commentato

Modalita' d'esame

- **Per sostenere la prova lo studente dovrà usare le proprie credenziali di accesso ai servizi di Ateneo (Single Sign-On, come per la posta elettronica o per il sistema di E-Learning del DEI). La mancanza delle credenziali SSO impedirà di sostenere l'esame.**
- Le credenziali sono personali.

Pubblicazione dei risultati

- Dopo la correzione della prova, i risultati dell'esame saranno pubblicati su Uniweb
- Ciascuno studente riceverà un avviso nella posta elettronica
- Insieme ai risultati verrà indicata la data in cui sarà possibile prendere visione del proprio elaborato previa prenotazione

CODICE
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Pubblicazione dei risultati

- Dal momento della pubblicazione dei risultati in UniWeb, ciascuno studente ha 7 giorni di tempo per **rifiutare** il voto. Trascorsi i 7 giorni, i voti che non sono stati rifiutati verranno verbalizzati online dal docente. Non vengono più scritti voti nel libretto cartaceo. I voti rifiutati vengono eliminati.
- **Chi avesse motivate esigenze di verbalizzare il voto prima che trascorrano i 7 giorni DEVE avvertire il docente durante la prova d'esame, in modo che possa attivare una "procedura d'urgenza" ed eventualmente correggere il compito prima degli altri.**

Take home message

- Studiate e venite all'esame preparati!
  - Quiz di autovalutazione
  - Esercizi dei laboratori
  - Studiare slide e codice messi a disposizione
  - Riferimenti sul libro di quanto fatto
  - Programma svolto settimana per settimana
  - Forum del corso

Concludendo….

Obiettivi del corso

- Introdurre concetti fondamentali dell'informatica
- Proporre un approccio **ingegneristico e progettuale** all'attività di programmazione
- Utilizzare **concretamente** il linguaggio Java nella sua formulazione base per implementare soluzioni algoritmiche

Programma del corso

- Concetti fondamentali di informatica
  - Modello di Von Neumann
  - Rappresentazione dell'informazione
  - Concetto di algoritmo
  - Algoritmi fondamentali di ricerca e di ordinamento
  - Ricorsione
  - Complessità computazionale
  - Strutture dati e strutture dati astratte
    - Array, liste, pile, code, insiemi, mappe, multimappe, tabelle, hashtable

Programma del corso

- Concetti fondamentali di programmazione
  - Sintassi di base del linguaggio Java
    - classi, metodi, strutture di controllo, operatori
  - Caratteristiche proprie della programmazione ad oggetti
    - polimorfismo, ereditarietà
- Realizzazione di semplici progetti di programmazione nei laboratori didattici


The End

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Wooclap!!!

UNIVERSITÀ DEGLI STUDI DI PADOVA