BioInformatica

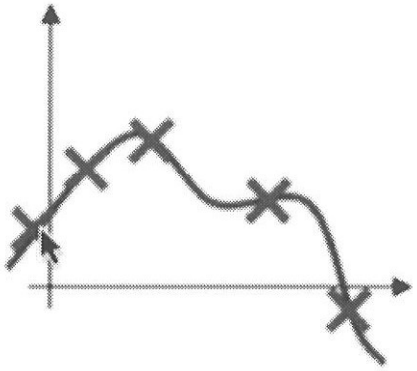
27/09/2021

È una materia a cavallo tra biologia e informatica, che sono discipline con 'metodi di indagine' diversi.
La Bioinformatica dovrebbe trovare un punto di incontro tra questi due approcci diversi.

Il BIOLOGO si occupa di *Life Science* ha un approccio esperimenti, ogni volta che si studia un fenomeno si cerca di raccogliere dati riguardo ad alcuni particolari.
Si descrive il fenomeno come una funzione del tipo:

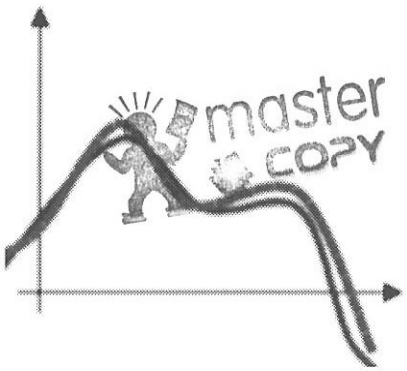
$$y = f(x)$$

Dove f è incognita.
Lo scopo della ricerca è quello di caratterizzare la f, dunque il biologo cerca di capire dove si collocano i punti significativi.



In pratica l'approccio è quello di raccogliere delle coppie di punto che in qualche modo la caratterizzano.
Si classificano queste coppie per capire se si possono raggruppare in qualche modo.

Gli informatici hanno un approccio un po' diverso, nella caratterizzazione degli f.

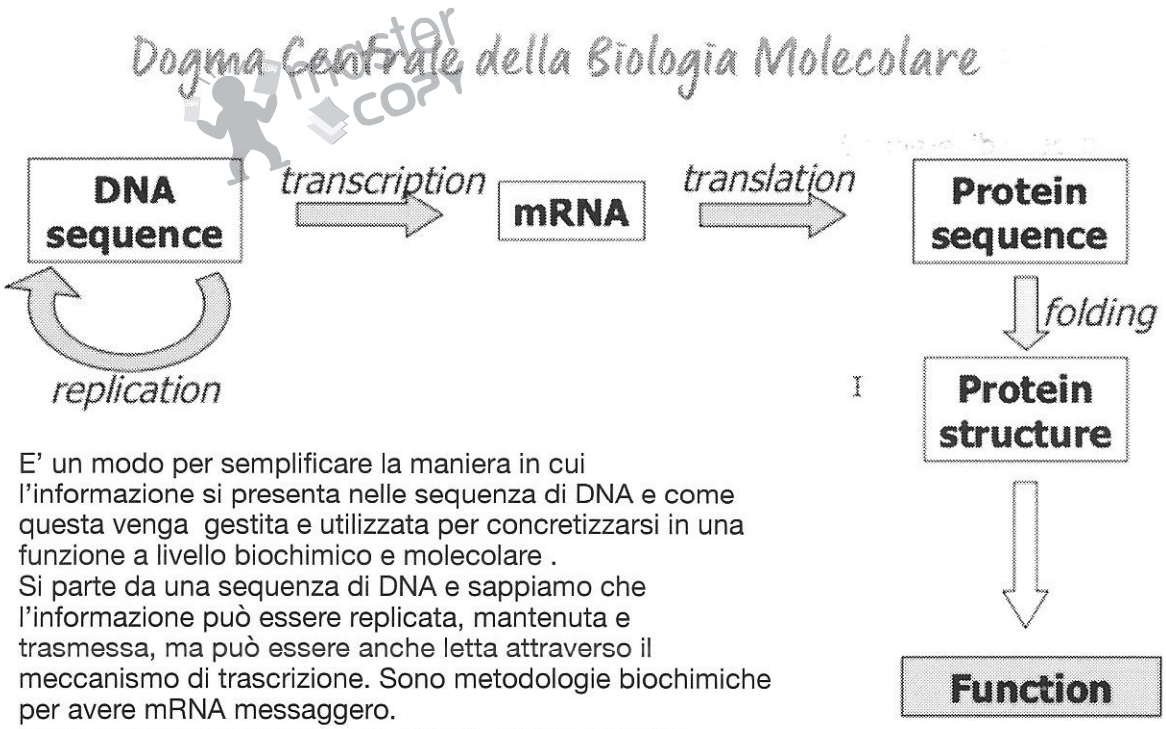


Loro cercano una f' che approssimi la funzione cercata al meglio. Cercano dei modelli che riescono a spiegare l'andamento di f.
Infatti trovare le coppie dei punti prevede una serie di lunghi esperimenti in laboratorio.
Dal punto di vista informatico è importante il modello che in qualche modo è in grado di spiegare l'andamento di f.

Questo perché per un informatico il singolo punto non è significativo, è importante il modello e lo scenario globale.

Effettivamente le problematiche che possono essere affrontate con metodologie computazionali ce ne sono tantissime.
Gli strumenti computazionali variano in base alla grandezza e alla tipologia di struttura biologica che vogliamo descrivere.

La Bioinformatica, dunque, serve a implementare approcci computazionali su problemi che non sono di facile risoluzione.
E' molto importante in campo di ricerca biomedica.

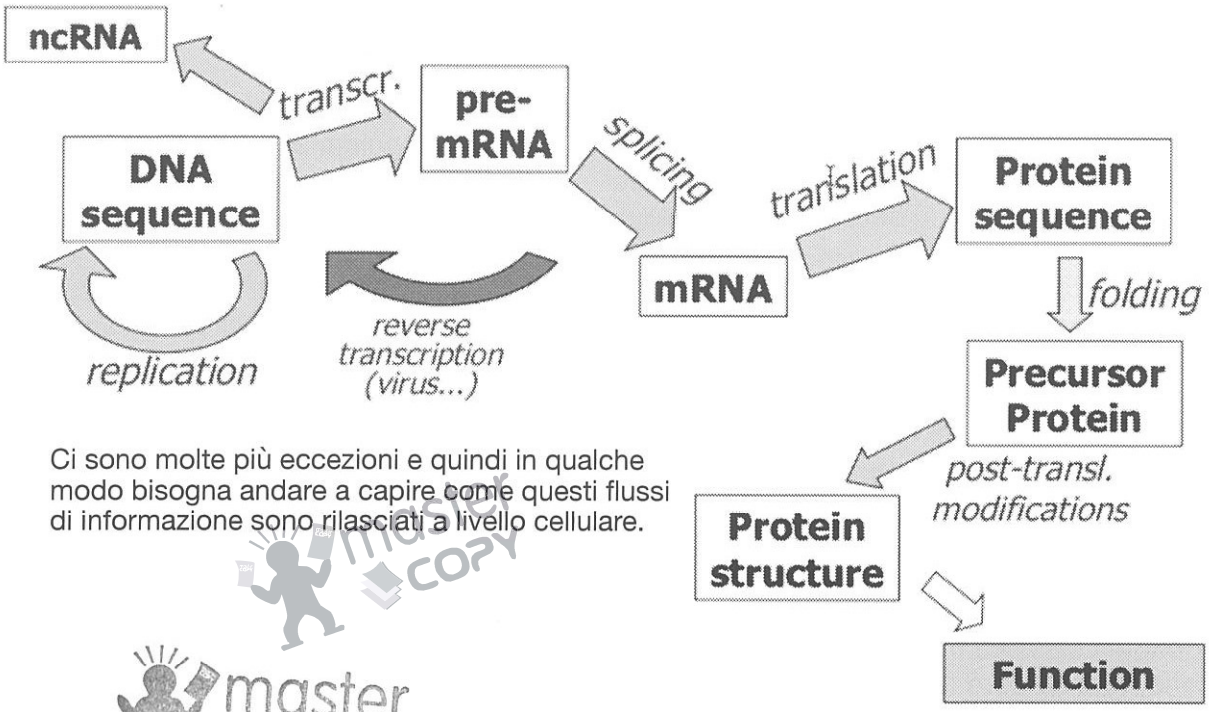


E' un modo per semplificare la maniera in cui l'informazione si presenta nelle sequenza di DNA e come questa venga gestita e utilizzata per concretizzarsi in una funzione a livello biochimico e molecolare .
Si parte da una sequenza di DNA e sappiamo che l'informazione può essere replicata, mantenuta e trasmessa, ma può essere anche letta attraverso il meccanismo di trascrizione. Sono metodologie biochimiche per avere mRNA messaggero.
Sono passaggi necessari per avere la sintesi proteica, posso procedere a assemblare le mie catene proteiche un aminoacido dopo l'altro.

Esiste una sorta di codifica tra sequenze di tre basi delle mRNA e uno specifico aminoacido, per cui sono in grado di trasformare l'informazione presente nell'mRNA sotto sequenze di coloni, triplette di basi, in una sequenza di aminoacidi, avendo sequenza proteica.

Nelle condizioni tipiche la catena di aminoacidi si ripiega su se stessa formando una biomolecola, che ha una determinata forma 3D e questa sarà quella che determinerà la sua funzione. È proprio dalla forma che la macromolecola assume la funzione di interagire con altre molecole.
Oltre il meccanismo di replicazione ho un flusso di informazioni che va da DNA alla funzione svolta a livello biologico.

Questa slide è soltanto un'approssimazione le cose in realtà sono molto più complicate.



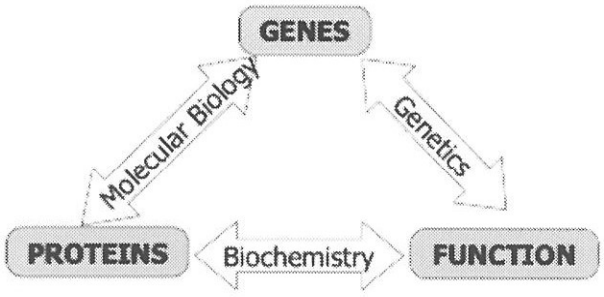
Ci sono molte più eccezioni e quindi in qualche modo bisogna andare a capire come questi flussi di informazione sono rilasciati a livello cellulare.

master copy

La prima grossa differenza è la presenza di un meccanismo per cui il passaggio dal DNA alla sequenza proteica non è mediato solo dall'mRNA, ma l'mRNA per poter essere utilizzato all'interno dei meccanismi che portano alla sintesi proteica deve essere il risultato di un pre-processing, ovvero un'elaborazione preventiva.
Il DNA viene trascritto e ottengo un precursore dell'RNA messaggero che verrà elaborato andando a togliere in maniera selettiva alcune porzioni per averne la forma effettiva e procedere nella traduzione in sequenza proteica.
Questo avviene negli eucarioti e quindi questi sono i processi che ci interessano di più.
Non è un processo che ha solo una direzione, non è nemmeno vero che il percorso sia unico, perché ci sono prodotti intermedi che hanno una loro funzione, ci sono sequenze di mRNA chiamati non codificanti che entrano a far parte di meccanismi di regolazione metabolica a livello cellulare.

Anche dopo l'operazione di folding, piegamento, spesso si hanno successive modifiche alla molecola e possiamo avere aggregazioni e macromolecole diverse. Non si arriva mai direttamente alla funzione finale.
La cosa dunque non è banale.
Per studiare lo scenario generale ci sono delle discipline diverse.
A livello biologico la genetica studia le relazioni tra i geni e le funzioni, la biochimica tra funzioni e proteine. Noi dobbiamo sapere un po' di tutto, perché la bioinformatica è un campo multidisciplinare, devo avere nozioni di genetica e biochimica.

master copy



I bioinformatici possono avere un background di partenza diversa.

E' un punto di incontro tra persone con background diversi.

Possiamo identificare un flusso di informazioni dalle sequenze alle funzioni.

Dalla sequenza di arriva a struttura della proteina e dunque di fatto alla funzione.
Le varie sequenze tra di loro hanno relazione di tipo evolutivo, quindi spesso uno strumento importante è l'analisi filo-genetica delle sequenze stesse. Le funzioni biologiche dipendono dai movimenti delle strutture biologiche e da come queste hanno interagito tra di loro.
Le funzioni posso anche dipendere da complessi di più di una molecola e questo percorso che va dalla sequenza alla funzione non è mai lineare, ma ci sono tanti fattori che possono contribuire.

L'analisi dei dati ad oggi viene fatta con strumenti di carattere informatico, sono molto interessanti , ma non coperte da questo corso.

Lo strumento pratico per lavorare sulle sequenze, sugli algoritmi etc è un linguaggio di scripting.

Il linguaggio classico usato si chiama *Pearl* che è particolarmente efficiente per elaborare stringhe, oramai però il linguaggio che si presta meglio Python che è un linguaggio ormai molto diffuso e si presta bene a essere usato in bioinformatica proprio per la sua adattabilità a elaborare stringhe.

Si usano stringhe perché le sequenze che analizzeremo saranno rappresentate come successione di caratteri e quindi stringhe.

Si usa Python perché è disponibile, relativamente efficiente e piò essere usato come collante per far funzionare insieme applicazioni diverse. Dispone di un numero di librerie incredibile.
L'IDE usato sarà Spyder.

PYTHON

Si dice che Python sia semplice perché ha una sintassi naturale, ha un modo per individuare le parti di codice molto preciso che rende i programmi chiari e facile da leggere.

Quando si scrive un programma in C++ e si deve identificare un blocco di codice si mettono le { } , ogni linea può iniziare in qualsiasi punto.
Si possono o meno lasciare spazi vuoti il programma non cambia, il programma si legge meglio se si allineano delle parti per colonna (formattazione).
I progettisti di Python hanno deciso di individuare i blocchi attraverso il fatto che un blocco di codice si individua a partire dalla sua distanza dal bordo sinistro. Quindi se viene spostato qualcosa di un carattere questo comporta un errore nello specificare il blocco del codice , quindi avremo un errore in sintassi.

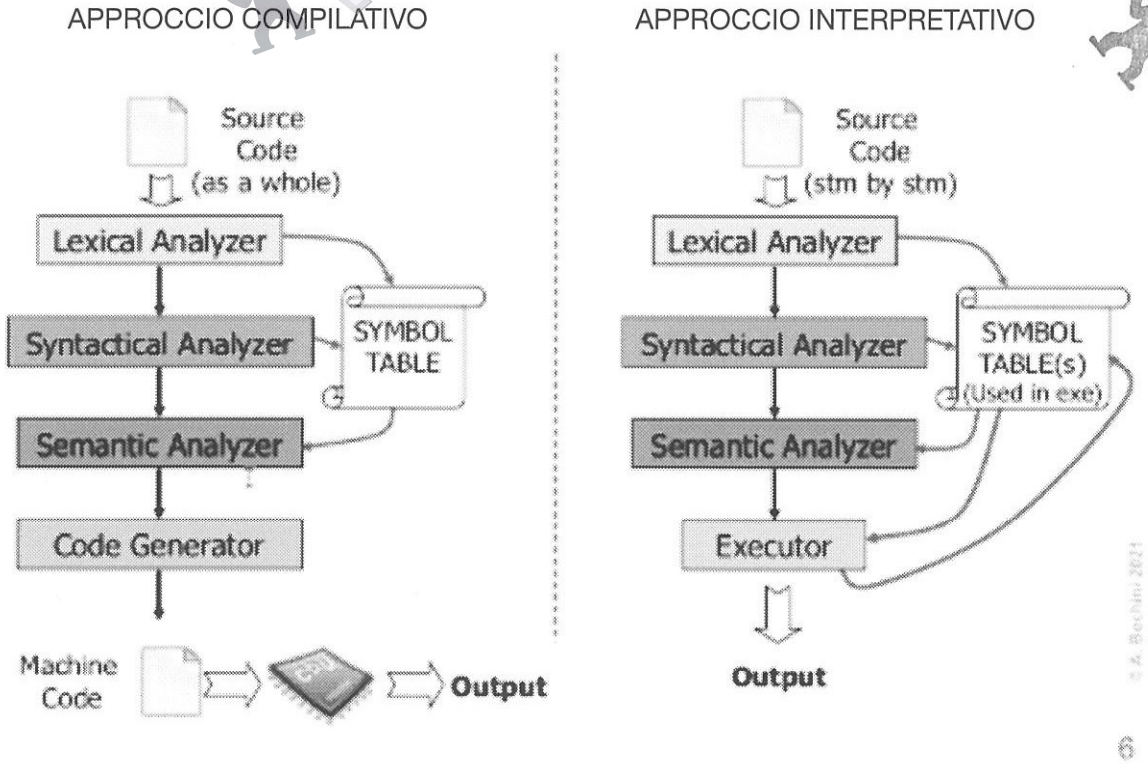
E' facilmente leggibile perché quindi se non lo scrivo per bene a livello di spazi il programma non funziona. Siamo forzati a scrivere il codice in maniera ordinata.

Riutilizzare pezzi di codice 'copiati' non è mai una brutta cosa in informatica.
E' una implementazione object Oriented.

Se si utilizzano le giuste librerie e si organizza bene il codice si avranno delle buone performance.
tipicamente linguaggi diversi seguono un approccio diverso. C++ segue l'approccio di compilazione, dopo aver scritto il programma si clicca il pulsante per la compilazione e genera programma eseguibile che genera il programma.

Altri seguono l'interpretazione , che vuol dire che esiste un altro programma interprete che prende come input il codice, analizza tutto il programma e lo esegue riga per riga.

Nel caso della compilazione il programma prende l'intera sorgente e lo esegue direttamente dalla CPU e quindi è naturalmente più efficiente. Nell'interprentazione ho un intermediario, un altro programma e quindi è più pesante.



Le varie fasi sono iudicate con blocchi di colore diverso.
Nella compilazione prendo tutto il codice sorgente e il compilatore cosa fa?
Prima c'è l'analizzatore lessicale che è il primo livello di analisi e tiene traccia di questa analisi in struttura dati interna che si chiama tabella dei simboli.
Ho poi l'analisi sintattica, si associa a ciascun commando quello che deve essere fatto, poi si analizza la sintassi che prende in ingresso anche le informazioni presenti nella tabella dei simboli. Deve capire come analizzare ogni simbolo per arrivare al passo finale che è la generazione del codice.
Questa struttura dati serve per dare in pasto alla struttura di codice tutto ciò che gli serve.
Il generatore di codice genera del codice macchina da far girare sulla CPU.

All'interno di un approccio interpretativo il discorso è quasi analogo, ma non analizza il codice tutto insieme ma statement by statement. Usa sempre la tabella dei simboli e tutti i blocchi analogamente all'approccio precedente però a valle del blocco troviamo