

LA SINTESI PROTEICA

PROCESSING - SCHEDA TECNICA



Il programma illustra il processo biochimico della **sintesi proteica**, nel quale l'informazione genetica contenuta nell'RNA viene convertita in **proteine**.

Si vuole dimostrare come la biologia si basi su un meccanismo di **codici, cifrature e decifrate**: come il codice binario è alla base del meccanismo informatico, i nostri processi vitali si fondano su un codice altrettanto semplice (e allo stesso tempo complesso) che è il nostro DNA.

Si divide in due parti principali, che richiamano le due fasi del processo biochimico, composto da:

1. Trascrizione

2. Traduzione

Organizzazione del programma:

Divisione in **3 sketch**:

- sketch principale denominato **sintesi_proteica**
 - sketch secondario denominato **Trascrizione** contenente le funzioni **void disegna_base** e **char Trascrizione**
 - sketch per la classe di oggetti denominata **tRNA**
-

0. PARTE INIZIALE

1. All'avvio del programma l'immagine di una cellula si ingrandisce, nella zona del **nucleo**, luogo in cui avviene la prima fase della sintesi proteica (*trascrizione*).
2. Tasto di avvio: la comparsa di un apposito tasto "**continua**", se cliccato, permette all'utente di visualizzare l'inizio del processo.
3. Scelta del tratto di codice genetico da decodificare nel programma: l'utente è chiamato a scegliere tra **3 sequenze di DNA**, la sequenza selezionata verrà tradotta in **proteina**.

1. TRASCRIZIONE

1. **Animazione iniziale:** il programma mostra all'utente come avviene il processo di **trascrizione**. L'immagine di una **doppia elica di Dna** scorre sullo schermo e, attraverso l'enzima **RNA-polimerasi** (figura in verde), compaiono gradualmente i tratti tradotti in RNA messaggero (**mRNA**).

2. Il processo della trascrizione viene visualizzato nel dettaglio, mostrando ciò che avviene base per base.

Prima cifratura: tramite l'utilizzo della specifica funzione di tipo **char Trascrizione**, viene associata ad ogni base di DNA, la corrisponde base nel linguaggio RNA.

La sequenza di basi di DNA da tradurre varia a seconda della scelta effettuata dall'utente (prima / seconda / terza immagine).

Entrambe le sequenze sono memorizzate all'interno di **array di caratteri**, e vengono disegnate sullo schermo tramite la funzione di tipo **void disegna_base** che, per ogni carattere dell'array, carica l'immagine della base corrispondente.

2. TRADUZIONE

1. **Ingrandimento graduale** del luogo della cellula in cui avviene la seconda fase del processo: la **traduzione**. L'attenzione viene ora spostata sul reticolo endoplasmatico (dove compare nuovamente il tasto "**continua**"), in cui ogni singola base dell'mRna verrà tradotta in amminoacido (**seconda cifratura**).
 2. Sullo schermo si presenta la sequenza di basi in linguaggio **mRNA** precedentemente tradotta (punto 2 della trascrizione).
 3. Il processo di traduzione può iniziare grazie alla molecola di **tRna** (RNA transfer) e ai **ribosomi** del reticolo. Poiché il suo funzionamento è piuttosto complesso, per gestirlo, è stata creata una **classe tRna** (class tRna).
-

ANALIZZANDO IL CODICE NEL DETTAGLIO...

LA CLASSE tRna

GLI ATTRIBUTI:

- Un **contatore** di tipo `int`
- Due variabili di **posizione** di tipo `float`
- Due variabili `PI` `Image`
- 4 array di tipo `char` di cui:
 - 3 **già riempiti** con le tre diverse sequenze di Dna.
 - 1 **vuoto** che, a seconda del DNA scelto, conterrà la corrispondente trascrizione in RNA.

I METODI:

- **void Anticodone**: chiamando la funzione `char Trascrizione`, questo metodo è responsabile del riempimento di un array `char anticod[]` che contiene 3 caratteri, ovvero 3 basi azotate trascritte in modo complementare all'array `char` di Rna. Il primo metodo inoltre carica l'immagine della **molecola tRNA** (a forma di T ribaltata) e disegna, chiamando la funzione `disegna_base`, le basi in linguaggio RNA contenute in `anticod[]`.
 - **void Traduzione**: la sequenza di mRNA viene letta di **tre basi in tre basi** (di 3 caratteri in 3) e viene riempita una stringa `cod` che ne conterrà i caratteri (esempio AUG). Rispettando quanto afferma la biologia, il primo amminoacido della catena ad essere prodotto sarà sempre la **metionina** perché corrisponde al codone identificato come "*sito d'inizio*".

Il metodo, tramite una struttura **switch-case**, associa ad ogni codone `cod` (in linguaggio mRNA) **un amminoacido** disegnando un piccolo cerchio colorato (è stato scelto un colore diverso per ogni amminoacido e nel cerchio è contenuta la sigla dello stesso). Il codone viene confrontato con tutte le sequenze degli amminoacidi presenti nel mondo dei viventi. Eccezione per l'ultimo codone, che si definisce "*codone di stop*", poiché non codifica per un amminoacido ma **termina il processo** di traduzione (in questo caso compare un piccolo rettangolo nero).
 - **boolean tRna_arrivato**: quest'ultimo metodo della classe restituisce in return un booleano che controlla quando il tRNA ha raggiunto una **posizione finale** fissata nello sketch principale.
-

La classe nello sketch principale

4. Nello sketch principale (sintesi_proteica) viene creato **un array di 6 oggetti** della classe tRna.
5. Questi compaiono **uno dopo l'altro** dall'alto della schermata grazie al metodo tRna_arrivato.
L'elemento di posizione corrispondente al tRna in questione di un array di variabili booleane sitoP[], assume lo stato **true**. L'array viene utilizzato per l'incremento di una variabile P, che attraverso un ciclo for permette l'arrivo graduale del tRna successivo quando il precedente è arrivato in **posizione finale**.
6. Rispettando quanto avviene biologicamente, gli oggetti della classe tRna si trovano all'interno di un **ribosoma** che scorre lungo il trascritto di **mRna**.
7. Il ribosoma presenta **tre siti** "A" "P" ed "E": il primo tRna arriva nel sito "**P**". Quando il successivo arriva nel sito "**A**", attraverso l'istruzione rect viene disegnato il **legame peptidico** che lega gli amminoacidi due alla volta.
8. Successivamente il legame **scorre in avanti** di un codone e il primo tRna, che ora si trova nel sito "**E**" (sito exit), esce gradualmente dallo schermo lasciandovi **solo l'aminoacido** (il piccolo cerchio) che trasporta.
9. La prima tripletta codifica sempre per l'aminoacido metionina ("MET") poiché il corrispondente codone "AUG" è il **codone di inizio** della traduzione.
10. L'ultima tripletta codifica sempre per il **codone di stop** determinando la fine del processo di traduzione.

3. CATENA POLIPEPTIDICA

1. La **catena di amminoacidi**, uniti tra di loro, da' origine alla **struttura primaria** di una proteina, che ha caratteristiche e funzioni differenti in base alla sequenza di amminoacidi presenti.
 2. Il polipeptide, a seconda della sua struttura, ha una **diversa destinazione** all'interno della cellula. Per questo a seconda del tratto di Dna che si era scelto di tradurre, all'inizio del programma, l'immagine della cellula viene **gradualmente ingrandita** sullo sfondo dello schermo, spostando l'attenzione su una sua precisa parte (nucleo, apparato del Golgi o mitocondri), simulando il viaggio della catena di amminoacidi all'interno di queste strutture.
-