

# Elementi di Bioinformatica

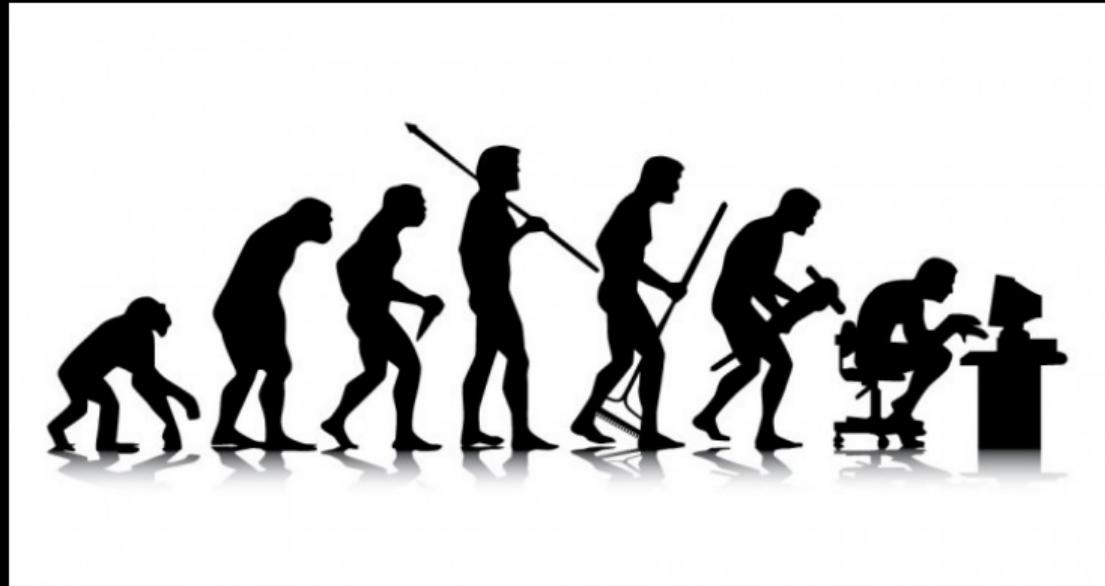
Gianluca Della Vedova

Univ. Milano–Bicocca  
<https://gianluca.dellavedova.org>

20 luglio 2025

Alberi evolutivi — Filogenesi

# Evoluzione



- Effetti visibili in generazioni
- Mutazioni casuali

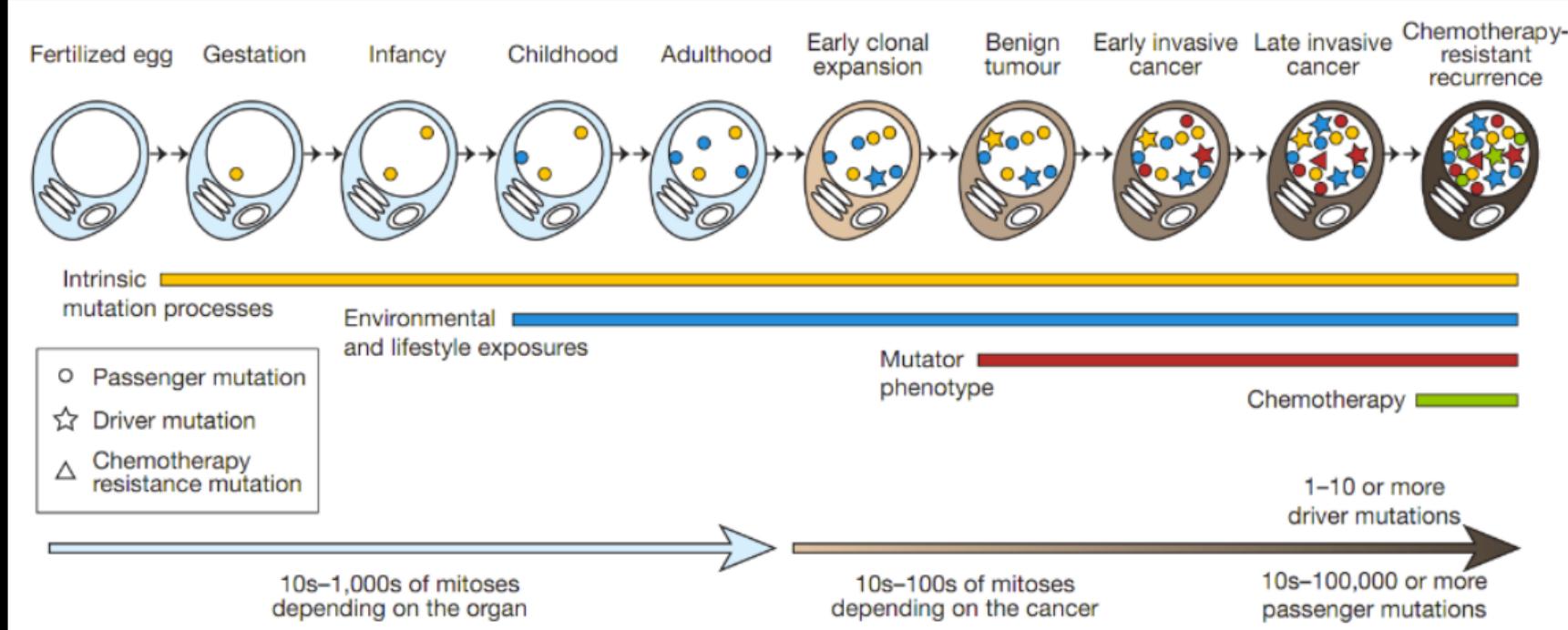
# Mutazioni reali



# Mutazione fantasiosa

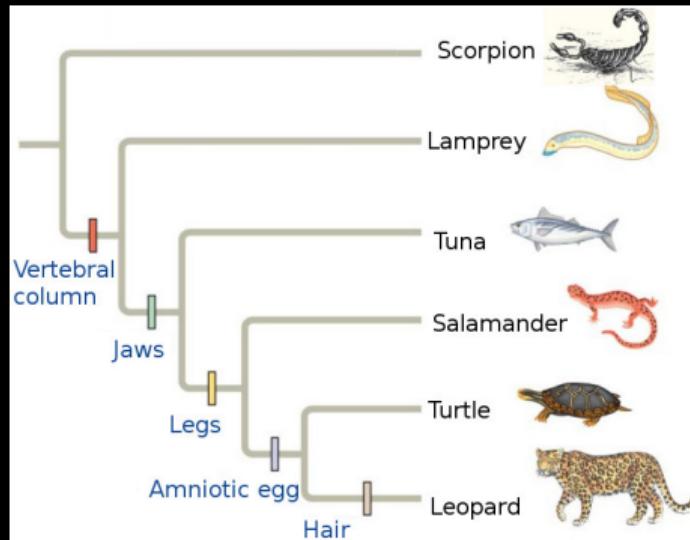


# Evoluzione in un individuo



- Cellule accumulano mutazioni durante la vita

# Evoluzione basata su caratteri



## Regola 1 (semplice)

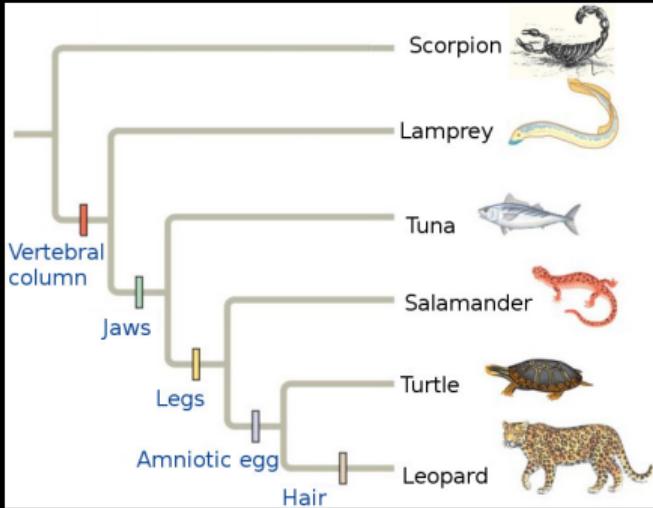
Ogni carattere è acquisito esattamente una volta nell'albero.

# Filogenesi perfetta

	A	J	H	L	V
Scorpione	0	0	0	0	0
Anguilla	0	0	0	0	1
Tonno	0	1	0	0	1
Salamandra	0	1	0	1	1
Tartaruga	1	1	0	1	1
Leopardo	1	1	1	1	1

## Problema

- Input: matrice binaria  $M$
- Output: un albero che spiega  $M$ , se esiste



## Algorithm di Gusfield — lineare

- 1 Radix Sort delle colonne, in ordine decrescente (anche del numero di 1)
- 2 Costruire l'albero, una specie alla volta

# Caratteri e stati

## Cambio di stato

- Un carattere  $c$  è **acquisito**  $\Rightarrow$  lo stato di  $c$  passa da 0 a 1 in un arco
- Un carattere  $c$  è **perso**  $\Rightarrow$  lo stato di  $c$  passa da 1 a 0 in un arco (**mutazione ricorrente**)

## Modelli di evoluzione

Ogni carattere  $c$  è acquisito **esattamente una volta** nell'albero.

- 1 Filogenesi perfetta: nessuna mutazione ricorrente, nessuna perdita
- 2 **Dollo**: mutazioni ricorrenti senza limiti, ma senza perdite
- 3 **Camin-Sokal**: Perdite senza limiti, ma senza mutazioni ricorrenti

# Approcci basati su parsimonia.

- Piccola (topologia nota) vs grande (topologia ignota)
- Algoritmo di Fitch
- Algoritmo di Sankoff
- Confronto

# Piccolo problema di parsimonia

## Istanza

- Matrice  $M$  con  $n$  specie e insieme di  $m$  caratteri  $C$ .
- Albero  $T$ , le cui foglie corrispondono alle specie di  $M$
- Per ogni carattere  $c \in C$ , un costo  $w_c$  fra ogni coppia di stati

## Soluzioni ammissibili

Per ogni carattere  $c \in C$ , una etichettatura  $\lambda_c$  che assegna ad ogni nodo uno degli stati possibili per  $C$

## Funzione obiettivo

$$\min \sum_{c \in C} \sum_{(x,y) \in E(T)} w_c(\lambda_c(x), \lambda_c(y)), \text{ dove } E(T) \text{ è l'insieme di lati di } T$$

# Algoritmo Sankoff

## Osservazione

Ogni carattere può essere gestito separatamente

## Programmazione dinamica

- $P[x, z]$ : soluzione ottimale del sottoalbero di  $T$  che ha radice  $x$ , sotto la condizione che  $x$  abbia etichetta  $z$
- $P[x, z] = 0$ , se  $x$  è una foglia con etichetta  $z$
- $P[x, z] = +\infty$ , se  $x$  è una foglia con etichetta diversa da  $z$
- $P[x, z] = \sum_{f \in F(x)} \min_s \{w(z, s) + P[f, s]\}$ , dove  $F(x)$  è l'insieme dei figli di  $x$  in  $T$ , se  $x$  è un nodo interno
- soluzione ottimale  $\min_s \{P[r, s]\}$ , dove  $r$  è la radice di  $T$

# Algoritmo Fitch

Solo per il caso non pesato, albero  $T$  binario

## Algoritmo

$S(x)$  è l'insieme di stati ottimali per il nodo  $x$ . Nessuna restizione sull'insieme degli stati.

- $S(x) = \lambda_c(x)$ , se  $x$  è una foglia
- $S(x) = S(f_l) \cap S(f_r)$ , dove  $f_l$  e  $f_r$  sono i figli di  $x$  in  $T$ , se  $S(f_l) \cap S(f_r) \neq \emptyset$
- $S(x) = S(f_l) \cup S(f_r)$ , dove  $f_l$  e  $f_r$  sono i figli di  $x$  in  $T$ , se  $S(f_l) \cap S(f_r) = \emptyset$

# Algoritmo Fitch

Solo per il caso non pesato, albero  $T$  binario

## Algoritmo

$S(x)$  è l'insieme di stati ottimali per il nodo  $x$ . Nessuna restizione sull'insieme degli stati.

- $S(x) = \lambda_c(x)$ , se  $x$  è una foglia
- $S(x) = S(f_l) \cap S(f_r)$ , dove  $f_l$  e  $f_r$  sono i figli di  $x$  in  $T$ , se  $S(f_l) \cap S(f_r) \neq \emptyset$
- $S(x) = S(f_l) \cup S(f_r)$ , dove  $f_l$  e  $f_r$  sono i figli di  $x$  in  $T$ , se  $S(f_l) \cap S(f_r) = \emptyset$

## Unificazione

$B(x)$ : insieme degli stati  $z$  tali che  $P[x, z]$  è minimo.

# Algoritmo Fitch

Solo per il caso non pesato, albero  $T$  binario

## Algoritmo

$S(x)$  è l'insieme di stati ottimali per il nodo  $x$ . Nessuna restizione sull'insieme degli stati.

- $S(x) = \lambda_c(x)$ , se  $x$  è una foglia
- $S(x) = S(f_l) \cap S(f_r)$ , dove  $f_l$  e  $f_r$  sono i figli di  $x$  in  $T$ , se  $S(f_l) \cap S(f_r) \neq \emptyset$
- $S(x) = S(f_l) \cup S(f_r)$ , dove  $f_l$  e  $f_r$  sono i figli di  $x$  in  $T$ , se  $S(f_l) \cap S(f_r) = \emptyset$

## Unificazione

$B(x)$ : insieme degli stati  $z$  tali che  $P[x, z]$  è minimo.  $B(x) = S(x)$

# Algoritmo Fitch

Solo per il caso non pesato, albero  $T$  binario

## Algoritmo

$S(x)$  è l'insieme di stati ottimali per il nodo  $x$ . Nessuna restizione sull'insieme degli stati.

- $S(x) = \lambda_c(x)$ , se  $x$  è una foglia
- $S(x) = S(f_l) \cap S(f_r)$ , dove  $f_l$  e  $f_r$  sono i figli di  $x$  in  $T$ , se  $S(f_l) \cap S(f_r) \neq \emptyset$
- $S(x) = S(f_l) \cup S(f_r)$ , dove  $f_l$  e  $f_r$  sono i figli di  $x$  in  $T$ , se  $S(f_l) \cap S(f_r) = \emptyset$

## Unificazione

$B(x)$ : insieme degli stati  $z$  tali che  $P[x, z]$  è minimo.  $B(x) = S(x)$

Come estendere Fitch ad albero generico (sempre caso non pesato)?

# Approcci basati su distanze.

## Distanza

$d : S \times S \mapsto \mathbb{R}^+$  tale che:

- 1  $d(a, b) = 0 \Leftrightarrow a = b, \forall a, b \in S$
- 2  $d(a, b) = d(b, a), \forall a, b \in S$  (simmetria)
- 3  $d(a, b) \leq d(a, c) + d(c, b), \forall a, b, c \in S$  (diseguaglianza triangolare)

# Approcci basati su distanze.

	Sc	A	To	Sa	Ta	L
Scorpione	-					
Anguilla	3	-				
Tonno	2	1	-			
Salamandra	1	2	1	-		
Tartaruga	3	1	4	1	-	
Leopardo	1	1	1	1	1	-

## Problema

- Input: matrice  $M$  di distanze stimate
- Output: un albero con le stesse distanze di  $M$ , se esiste

# UPGMA

- Unweighted Pair Group with Arithmetic Mean
- $D(C_1, C_2) \leftarrow \frac{1}{|C_1||C_2|} \sum_{i \in C_1} \sum_{j \in C_2} D(i, j)$
- All'inizio  $h = 0$  per ogni cluster/specie
- Fondi i due cluster  $C_1, C_2$  con minimo  $D(\cdot, \cdot)$ , ottenendo  $C$
- Per ogni cluster  $C^* \neq C$ ,  $D(C, C^*) = \frac{1}{|C||C^*|} \sum_{i \in C} \sum_{j \in C^*} D(i, j)$
- $h(C) \leftarrow \frac{1}{2}D(C_1, C_2)$
- $h(C) - h(C_1)$  etichetta  $(C, C_1)$ ;  $h(C) - h(C_2)$  etichetta  $(C, C_2)$
- UPGMA produce ultrametrica

# Ultrametrica

## definizione

$d : S \times S \mapsto \mathbb{R}^+$  tale che:

- 1**  $d(a, b) = 0 \Leftrightarrow a = b, \forall a, b \in S$
- 2**  $d(a, b) = d(b, a), \forall a, b \in S$  (simmetria)
- 3**  $d(a, b) \leq d(a, c) + d(c, b), \forall a, b, c \in S$  (disuguaglianza triangolare)
- 4**  $\max\{d(a, b), d(a, c), d(c, b)\}$  ottenuto da almeno 2 casi,  $\forall a, b, c \in S$

# Ultrametrica e orologio molecolare.

# Alberi e distanze additive.

## Proprietà

Sia  $T$  un albero binario senza radice e sia  $D$  la matrice delle distanze associata a  $T$ . Allora  $D$  soddisfa la condizione dei 4 punti.

## Condizione dei 4 punti

Si consideri:

- 1**  $D[v, w] + D[x, y]$
- 2**  $D[v, x] + D[w, y]$
- 3**  $D[v, y] + D[w, x]$

Il massimo dei tre valori è ottenuto da esattamente due dei 3 casi sopra

# Algoritmo per matrice di distanze additive.

## Tripla degenera

Siano  $x, y, z$  tre specie e sia  $D$  la matrice di distanza. Allora la tripla  $(x, y, z)$  è degenera se  $D[x, y] + D[x, z] = D[y, z]$ .

## Sbilancio

Siano  $x, y, z$  tre specie e sia  $D$  la matrice di distanza. Allora lo sbilancio di  $x$  rispetto a  $(y, z)$  è  $S(x, y, z) = D[x, y] + D[x, z] - D[y, z]$ .

# Neighbor Joining.

- $D(C_1, C_2) \leftarrow \frac{1}{|C_1||C_2|} \sum_{i \in C_1} \sum_{j \in C_2} D(i, j)$
- $u(C) \leftarrow \frac{1}{\text{num. cluster}-2} \sum_{C_3} D(C, C_3)$
- Fondi i due cluster  $C_1, C_2$  con minimo  $D(C_1, C_2) - u(C_1) - u(C_2)$ , ottenendo  $C$
- Per ogni cluster  $C^* \neq C$ ,  $D(C, C^*) = \frac{1}{|C||C^*|} \sum_{i \in C} \sum_{j \in C^*} D(i, j)$
- $\frac{1}{2} (D(C_1, C_2) + u(C_1) - u(C_2))$  etichetta  $(C, C_1)$
- $\frac{1}{2} (D(C_1, C_2) + u(C_2) - u(C_1))$  etichetta  $(C, C_2)$

# Modelli di evoluzione.

- Probabilità di transizione fra stati (A, C, G, T).
- dipende dal tempo trascorso fra i due eventi
- tasso istantaneo di mutazione
- probabilità di mutazione *in una generazione*: somma su ogni riga = 1

J. Felsenstein. Theoretical Evolutionary Genetics

# Modelli di evoluzione: Jukes-Cantor.

- ogni mutazione è equiprobabile
- $1 - \mu$ : nessuna mutazione
- $\mu/3$ : mutazione

# Modelli di evoluzione: Kimura 2 parametri

- Distinzione transizioni ( $A \leftrightarrow G, C \leftrightarrow T$ ), transversioni
- $1 - \mu$ : nessuna mutazione
- $\frac{R}{R+1}\mu$ : probabilità transizione
- $\frac{1}{2(R+1)}\mu$ : probabilità di trasversione  $A \leftrightarrow C$  o  $G \leftrightarrow T$
- $\frac{1}{2(R+1)}\mu$ : probabilità di trasversione  $A \leftrightarrow T$  o  $C \leftrightarrow G$
- $R = \frac{R}{R+1}\mu / \left(2 \frac{1}{2(R+1)}\mu\right)$ : rapporto probabilità di transizioni / probabilità trasversioni

# Modelli di evoluzione: General time-reversible

- matrice simmetrica
- conseguenza: alberi senza radice

# Massima verosimiglianza.

# Licenza d'uso

Quest'opera è soggetta alla licenza Creative Commons: Attribuzione-Condividi allo stesso modo 4.0. (<https://creativecommons.org/licenses/by-sa/4.0/>).

Sei libero di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire, recitare e modificare quest'opera alle seguenti condizioni:

- Attribuzione — Devi attribuire la paternità dell'opera nei modi indicati dall'autore o da chi ti ha dato l'opera in licenza e in modo tale da non suggerire che essi avallino te o il modo in cui tu usi l'opera.
- Condividi allo stesso modo — Se alteri o trasformi quest'opera, o se la usi per crearne un'altra, puoi distribuire l'opera risultante solo con una licenza identica o equivalente a questa.