

Elementi di Bioinformatica

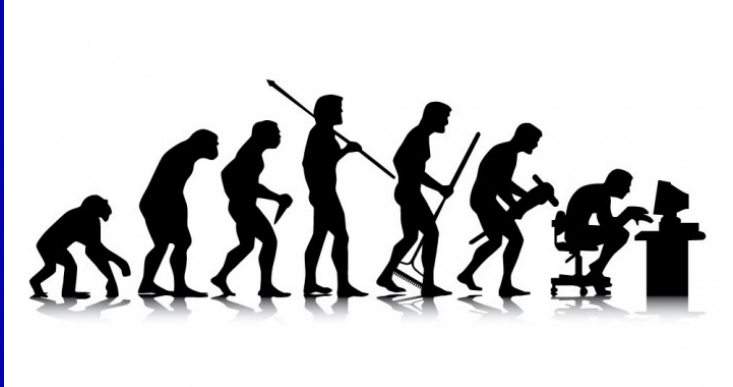
Gianluca Della Vedova

Univ. Milano-Bicocca
<http://gianluca.dellavedova.org>

5 dicembre 2018

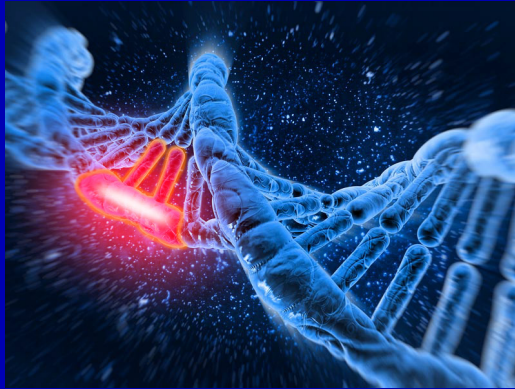
Alberi evolutivi

Evolution



- Change over generations
- Random mutations

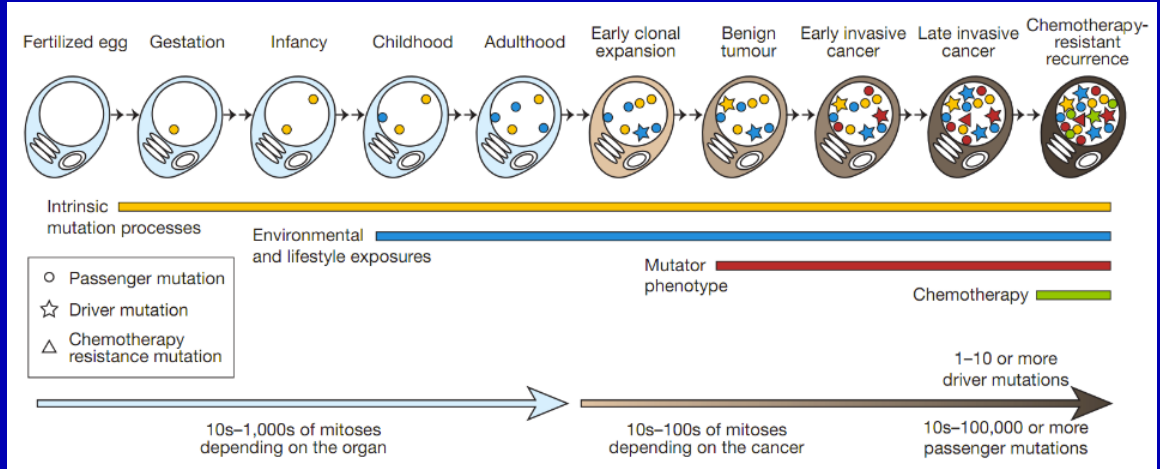
Actual Mutation



Hollywood Mutation

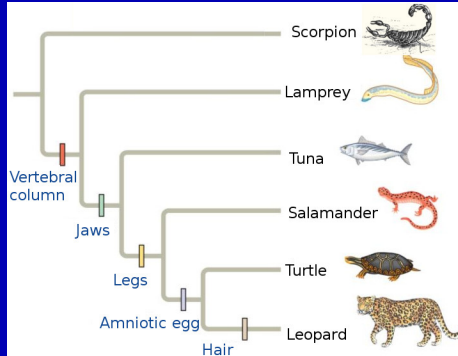


Individual Evolution



■ Cells **accumulate** mutations throughout the entire life

Character-based evolution



A possible rule

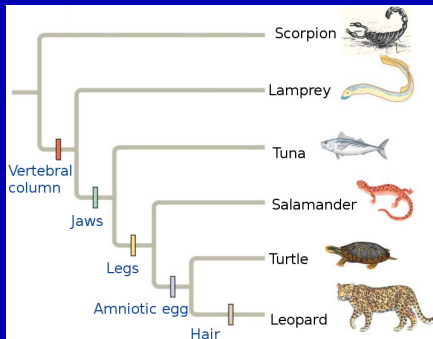
Each character is gained **exactly once** in the tree.

Perfect Phylogeny Problem

	A	J	H	L	V
Scorpion	0	0	0	0	0
Lamprey	0	0	0	0	1
Tuna	0	1	0	0	1
Salamander	0	1	0	1	1
Turtle	1	1	0	1	1
Leopard	1	1	1	1	1

Problem

- Input: a binary matrix M
- Output: a tree **explaining** M , if it exists



Linear time algorithm (Gusfield, Networks 1991)

- 1 Radix Sort the columns by decreasing number of 1s
- 2 Build the tree, inserting the species one at a time

Characters and States

Change of state

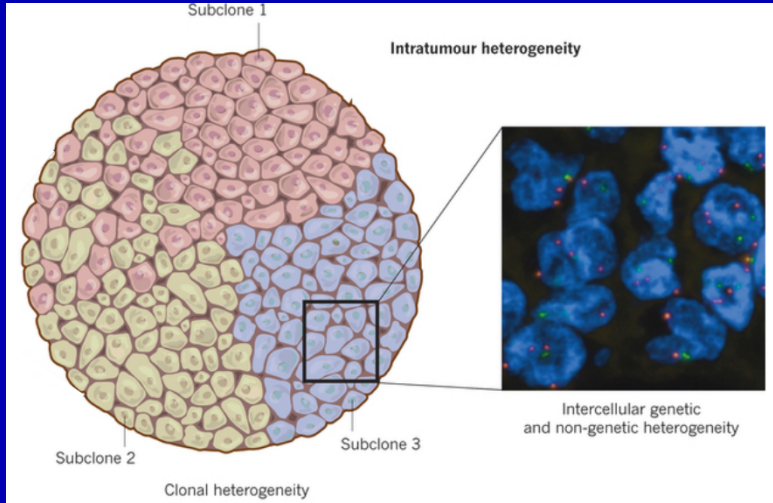
- A character c is **gained** \Rightarrow the state of c changes from 0 to 1 in an edge
- A character c is **lost** \Rightarrow the state of c changes from 1 to 0 in an edge (**backmutation**)

Models of Evolution

Each character c is gained **exactly once** in the tree.

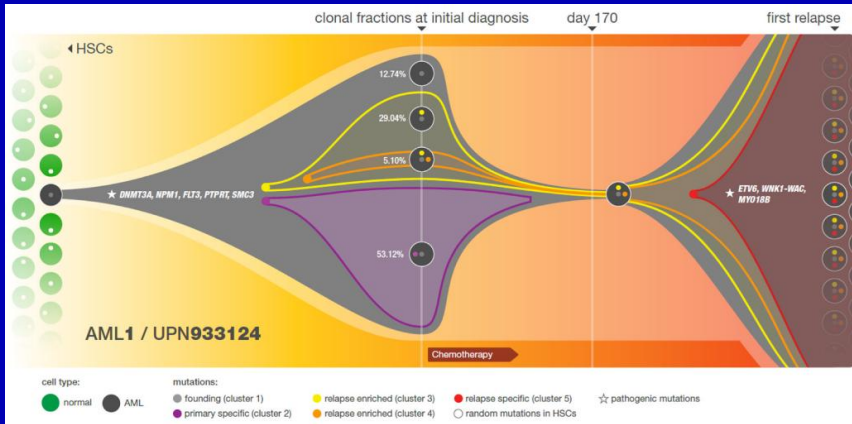
- 1 Perfect Phylogeny: No backmutations
- 2 Persistent Phylogeny: Each character can be lost at most once in the tree. **012 model**
- 3 **Dollo** parsimony: Unlimited backmutations

Tumors



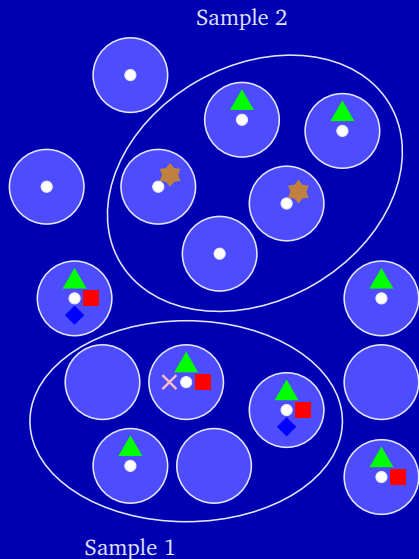
- A **tumor** is a mixture of healthy and cancer cells
- A **tumor** is a mixture of cancer clones

Tumor Evolution



- Different clones make different fractions of the tumor

Tumor Evolution



- A **sample** is a mixture of clones
- For each sample, we have the **frequency** of each mutation
- frequency matrix F



S_1	0.2	0.6	0.6	0.4	0.2	0.0
S_2	0.0	0.4	1.0	0.0	0.0	0.4

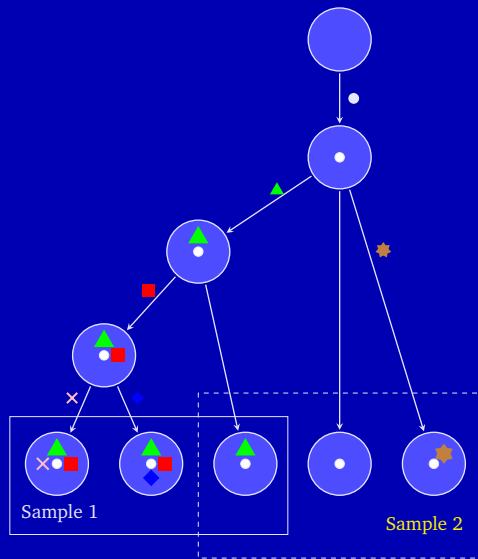
Tumor Evolution: Compute

Matrix B representing tree T

◆	▲	●	■	×	★
0	0	1	0	0	1
0	1	1	1	1	0
0	1	1	0	0	0
0	0	1	0	0	0
1	1	1	1	0	0

Usage matrix U

	Species				
0	0.2	0.2	0	0.2	
0.4	0	0.4	0.2	0	



Approcci basati su parsimonia.

- Piccola vs grande parsimonia
- Algoritmo di Fitch
- Algoritmo di Sankoff
- Confronto

Piccola parsimonia

Istanza

- Matrice binaria M con n specie e insieme di 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 λ_c che assegna ad ogni nodo uno degli stati possibili per C

Funzione obiettivo

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

Algoritmo Sankoff

Osservazione

Ogni carattere può essere gestito separatamente

Programmazione dinamica

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

Algoritmo Fitch

Solo per il caso non pesato, albero T binario

Algoritmo

- $S(x) = \lambda_c(x)$, 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 $M[x, z]$ è minimo.

Algoritmo Fitch

Solo per il caso non pesato, albero T binario

Algoritmo

- $S(x) = \lambda_c(x)$, 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 $M[x, z]$ è minimo. **$B(x) = S(x)$**

Algoritmo Fitch

Solo per il caso non pesato, albero T binario

Algoritmo

- $S(x) = \lambda_c(x)$, 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 $M[x, z]$ è minimo. $B(x) = S(x)$

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

Approcci basati su distanze.

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.

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

Neighbor Joining.

- $D(C_1, C_2) \leftarrow \frac{1}{|C_1||C_2|} \sum_{i \in C_1} \sum_{j \in C_2} D(i, j)$
- $u(C) \leftarrow \frac{1}{\text{num. cluster}-2} \sum_{C_3} D(C, C_3)$
- All'inizio $h = 0$ per ogni cluster/specie
- 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 ($A \leftrightarrow G, C \leftrightarrow T$), trasversioni
- $1 - \mu$: 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.