Riconoscimento e recupero dell'informazione per bioinformatica

Filogenesi

Manuele Bicego

Corso di Laurea in Bioinformatica
Dipartimento di Informatica - Università di Verona

Sommario

- Introduzione alla filogenesi
- I tipi di alberi
- I passi della filogenesi
  - (creazione del dataset)
  - allineamento
  - albero (scelta del modello evolutivo e del metodo di clustering)
  - validazione

2

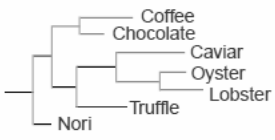
Introduzione

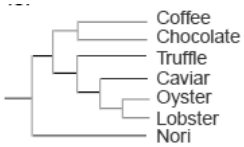
- ⇨ Filogenesi: la scienza che studia l'evoluzione e le relazioni evolutive degli organismi attraverso l'analisi e il confronto di sequenze proteiche o nucleotidiche
  - ⇨ Lo scopo è inferire le relazioni genealogiche tra gli organismi
- ⇨ Il risultato di un'analisi filogenetica è un albero, detto albero filogenetico
- ⇨ Gli organismi analizzati si chiamano in generale taxa (l'analisi comparativa di taxa è detta taxonomy)

3

L'albero filogenetico

- ⇨ Il risultato di un'analisi filogenetica
  - ⇨ Le foglie sono le sequenze da cui l'albero è stato derivato
  - ⇨ Un nodo interno rappresenta "l'antenato comune" di tutte le specie che stanno nel relativo sottoalbero
  - ⇨ (In generale) la lunghezza di un ramo misura la divergenza evolutiva (o la quantità di evoluzione) tra i due nodi che connette
    - ⇨ Più lungo è il ramo maggiore è l'evoluzione intercorsa tra l'antenato e le foglie



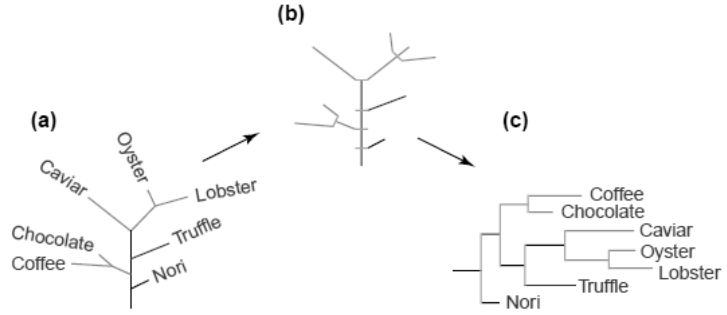


Eccezione: il cladogramma: tutti i nodi allineati

4

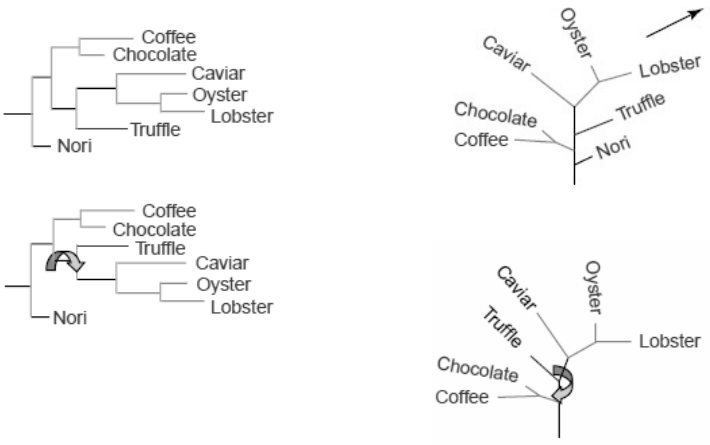
Tipi di albero

⇨ Due rappresentazioni equivalenti



5

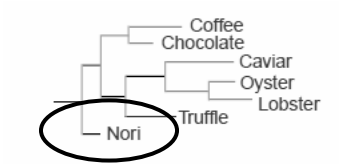
⇨ Nota: rotazioni sui nodi non cambiano l'albero



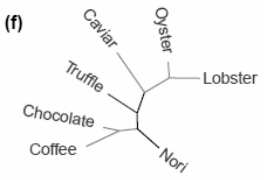
6

Tipi di albero

- Alberi con o senza radice:
  - La radice è l'elemento più "lontano" evolutivamente, difficile da scegliere
  - Di solito si sceglie un elemento che non c'entra nulla con le specie analizzate (outgroup)



  - Oppure si lascia l'albero "unrooted"



7

Passi di un'analisi filogenetica

1. Costruire il dataset
   - Ottenere le sequenze geniche o proteiche degli organismi in esame
2. Allineamento multiplo di sequenze
   - Registrare le sequenze tra di loro (inserire gap)
3. Clustering e costruzione dell'albero
   - Derivare l'albero filogenetico
4. Validazione
   - Determinare la robustezza dell'albero

8

Step 1: Costruire il dataset

⇨ Problema complesso ma non rilevante per questo corso

Una sola considerazione:

⇨ Occorre scegliere se analizzare le sequenze geniche o le sequenze proteiche

- ⇨ Sequenze geniche: informazione dettagliata ma rumorosa: adatte a trovare relazioni tra organismi "evolutivamente" vicini
- ⇨ Sequenze proteiche: informazione grezza ma pulita: adatte a trovare relazioni generali

- ⇨ Si può anche pensare di fondere le due fonti di informazione

9

Step 2: Allineamento

⇨ Allineamento tra due sequenze: trovare un sistema di riferimento comune, inserendo gap (-)

```
CAGATCTTGACGATCCCAAGTGGTTCATTGGCTTTAGATGAAG
TACCGATCTTGACGATCCCAAGTCATTGGCTTTAGATGAAG
CAGATCTTGACGATCCCAAGTGGTTCATTGGCTTTAGATGAAG
CAGATCTTCACGATCCCAAGTGGTTCATTGGCTTTAGAT

CA--GATCTTGACGATCCCAAGTGGTTCATTGGCTTTAGATGAAG
TACCGATCTTGACGATCCCAAG----TCATTGGCTTTAGATGAAG
CA--GATCTTGACGATCCCAAGTGGTTCATTGGCTTTAGATGAAG
CA--GATCTTCACGATCCCAAGTGGTTCATTGGCTTTAGAT----
```

⇨ Problema: occorre allineare simultaneamente molte sequenze

⇨ Soluzione: "Progressive sequence alignment"

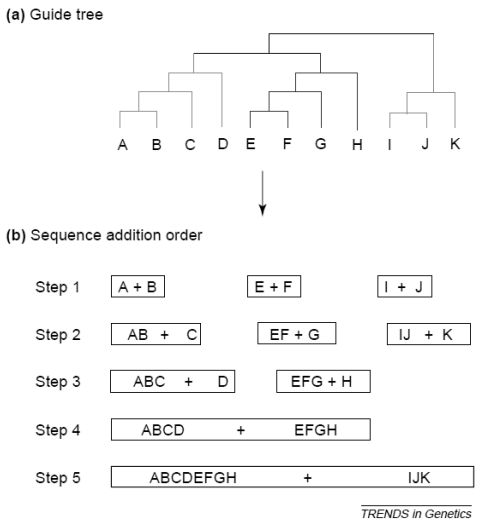
10

Step 2: Allineamento

Progressive Sequence Alignment

IDEA:

- ⇨ Allineare le sequenze passo dopo passo, aggiungendo ogni volta all'allineamento una sola sequenza
- ⇨ La sequenza considerata ad ogni iterazione è la più simile alle sequenze già considerate
  - ⇨ Necessità di un albero "grezzo" che faccia da guida (che determini l'ordine in cui le sequenze vengono aggiunte all'allineamento)



11

Step 2: Allineamento

- ⇨ Regola principale della tecnica Progressive Sequence Alignment: "Once a gap always a gap"
  - ⇨ I gap possono solo essere aggiunti o allargati, ma mai rimossi
- ⇨ Motivazioni
  - ⇨ Biologica) Si basa sull'assunzione che la miglior informazione su dove mettere i gap si può trovare solo tra le sequenze più simili
  - ⇨ Computazionale) il metodo risulta essere enormemente più veloce
- ⇨ Problema: a volte ad allineamento finito si riesce a vedere meglio dove piazzare i gap
  - ⇨ Soluzione: programmi come BioEdit che permettono di editare l'allineamento

12

Step 3: costruzione dell'albero

Due classi generali di approcci utilizzati:

⇨ Metodi "tree-searching"

- ⇨ Vanno a cercare l'albero migliore nel "tree space" (lo spazio di tutti gli alberi), in modo da ottimizzare un particolare criterio

⇨ Metodi distance-based

- ⇨ Partono da una matrice di distanza, e sono basati su metodi di clustering

13

Step 3: costruzione dell'albero

Metodi tree-searching: due approcci principali:

⇨ Maximum Parsimony:

- ⇨ trova l'albero che minimizza il numero di eventi evolutivi (mutazioni) dall'organismo ancestrale
- ⇨ PRO: può gestire facilmente inserzioni e cancellazioni
- ⇨ PRO: in alcune condizioni è molto efficiente
- ⇨ CONTRO: se l'albero vero ha un particolare tipo di struttura, la tecnica MP può fallire

⇨ Maximum Likelihood:

- ⇨ Dato un modello evolutivo, questo metodo seleziona l'ipotesi (l'albero) che meglio spiega i dati osservati
- ⇨ PRO: produce risultati molto accurati
- ⇨ CONTRO: lento (ricerca nel tree-space)
- ⇨ Possibili estensioni con modelli Bayesiani (MCMC)

14

Step 3: costruzione dell'albero

Metodi distance-based

⇨ Viene calcolata una distanza "evolutiva" tra tutte le coppie di sequenze.

  ⇨ La distanza si chiama evolutiva perché tiene conto di un modello evolutivo

⇨ L'albero filogenetico viene costruito a partire da questa matrice di distanza

  ⇨ Utilizzo di metodi di clustering generici (UPGMA)

  ⇨ Utilizzo di metodi nati ad hoc per la filogenesi (Neighbor Joining)

15

Step 3: costruzione dell'albero

Calcolo delle distanze e modelli evolutivi

⇨ Il calcolo della distanza si basa tipicamente sul numero di sostituzioni che ci sono tra le due sequenze (numero di caratteri diversi)

Diversi modelli evolutivi a seconda di:

⇨ Considerare il numero totale di siti analizzati

⇨ Considerare diversamente transizioni e transversioni (mutazioni tra strutture chimiche simili o diverse: purine (A,G) e pirimidine (C,T))

⇨ Considera o meno la frequenza con cui i nucleotidi appaiono

⇨ Considera o meno la frequenza con cui abbiamo transversioni rispetto a transizioni

16

Step 3: costruzione dell'albero

Distanze semplici:

⇨ Number of differences:

⇨ numero di siti dove le due sequenze differiscono

⇨ P-distance

⇨ Percentuale di siti nucleotidici dove le due sequenze sono differenti

⇨ Nessuna assunzione, solo normalizza sulla lunghezza

17

Step 3: costruzione dell'albero

Distanze più complesse: assumono un modello di sostituzione (cioè un modello che mi dice quanto pesare una sostituzione)

⇨ Jukes-Cantor

	A	T	C	G
A	-	α	α	α
T	α	-	α	α
C	α	α	-	α
G	α	α	α	-

⇨ 1 parametro, quanto peso dare ad una sostituzione

18

Step 3: costruzione dell'albero

⇨ Distanza Tajima-Nei

⇨ Pesa in modo diverso le sostituzioni tenendo conto della frequenza che i nucleotidi hanno all'interno delle sequenze

	A	T	C	G
A	-	αg_T	αg_C	αg_G
T	αg_A	-	αg_C	αg_G
C	αg_A	αg_T	-	αg_G
G	αg_A	αg_T	αg_C	-

⇨ Distanza di Kimura

⇨ Pesa in modo diverso le transversioni dalle transizioni

	A	T	C	G
A	-	β	β	α
T	β	-	α	β
C	β	α	-	β
G	α	β	β	-

19

Step 3: costruzione dell'albero

⇨ Ci sono distanze simili per il confronto di due sequenze proteiche

NOTA

⇨ Cambiando la distanza cambiano i risultati

⇨ La scelta deve avvenire considerando anche le informazioni a priori

⇨ Ci sono molte altre distanze, proposte negli ultimi anni, che tengono conto di altri fattori

⇨ ESEMPIO: Contenuto in GC

20

Step 3: costruzione dell'albero

- Costruzione dell'albero: clustering a partire dalla matrice delle distanze
  - Molte diverse tecniche
- Approccio di base: UPGMA (Unweighted pair group method using arithmetic averages)
  - Classico algoritmo agglomerativo gerarchico
  - La distanza tra cluster è definita come la media delle distanze di tutte le possibili coppie formate da un punto del primo e un punto del secondo
  - Da un punto di vista biologico è piuttosto povero: assume un rate di evoluzione costante

21

Step 3: costruzione dell'albero

- Neighbor Joining: approccio largamente utilizzato
- IDEA: funziona similarmente al metodo UPGMA
  - Trova i clusters C1 e C2 che minimizzano una funzione f(C1,C2)
  - Unisce i due cluster C1 e C2 in un nuovo cluster C
  - Aggiunge un nodo nell'albero corrispondente a C
  - Assegna le distanze ai nuovi rami
- Le differenze rispetto all'UPGMA sono relative a:
  - La scelta della funzione f(C1, C2)
  - Come assegnare le distanze ai rami

22

Step 3: costruzione dell'albero

Criterio da ottimizzare

⇨ Invece di scegliere i cluster Ci e Cj più vicini tra di loro, il neighbor joining allo stesso tempo:

- ⇨ minimizza la distanza tra i cluster Ci e Cj
- ⇨ e massimizza la separazione di Ci e Cj (entrambi) dagli altri cluster

Distanze dei rami:

⇨ Calcola esplicitamente la lunghezza dei rami

23

Step 3: costruzione dell'albero

L'algoritmo: Iterativamente

0. Si parte dalla matrice delle distanze d
1. Ripetere i seguenti passi:

⇨ Data la matrice di distanze *d* corrente (su *r* taxa / clusters) viene calcolata la matrice Q:

$$Q(i,j) = (n-2)d_{ij} - \sum_{C_k, k \neq j} d_{ik} - \sum_{C_k, k \neq i} d_{jk}$$

(distanza tra cluster Ci e Cj meno la distanza tra Ci e Cj e tutto il resto)

24

Step 3: costruzione dell'albero

- Viene trovata la coppia di taxa / cluster che minimizza la funzione Q
  - Viene creato un nodo dell'albero che unisce questi due cluster
- Viene calcolata la distanza tra i due cluster della coppia e il nuovo nodo
  - Siano Cf e Cg i cluster uniti, u il nuovo nodo che li contiene

$$BL_{f,u} = \frac{1}{2} d_{fg} + \frac{1}{2(n-2)} \left[\sum_{C_k, k \neq g} d_{fk} - \sum_{C_k, k \neq f} d_{gk} \right]$$

  - Questa distanza rappresenta la lunghezza dei rami che uniscono il nuovo nodo e i due cluster uniti

25

Step 3: costruzione dell'albero

- La matrice delle distanze viene ridotta:
  - Vengono eliminate le colonne e le righe relative ai cluster Cf e Cg
  - Viene calcolata la distanza tra il nuovo cluster e tutti gli altri (distanza tra Cf e Ck meno la distanza tra Cf e il nodo u)

$$d_{u,k} = \frac{1}{2} \left[d_{fk} - BL_{fu} \right] + \frac{1}{2} \left[d_{gk} - BL_{gu} \right]$$

- Si prosegue fino alla fine del clustering

26

Step 3: costruzione dell'albero

- ⇨ Vantaggi del Neighbor Joining
  - ⇨ Basato sul criterio di "evoluzione minima": ad ogni iterazione viene scelta la topologia che produce il ramo più corto (l'evoluzione minore)
  - ⇨ Molto veloce (complessità polinomiale), applicabile anche a dataset molto grandi
  - ⇨ Non assume un rate di evoluzione costante (diversamente dall'UPGMA)
  - ⇨ Statisticamente consistente con molti modelli evolutivi
- ⇨ Svantaggi:
  - ⇨ È un algoritmo greedy (sub ottimale)
  - ⇨ Esistono tecniche molto più complesse che posso superare le prestazioni del neighbor joining

27

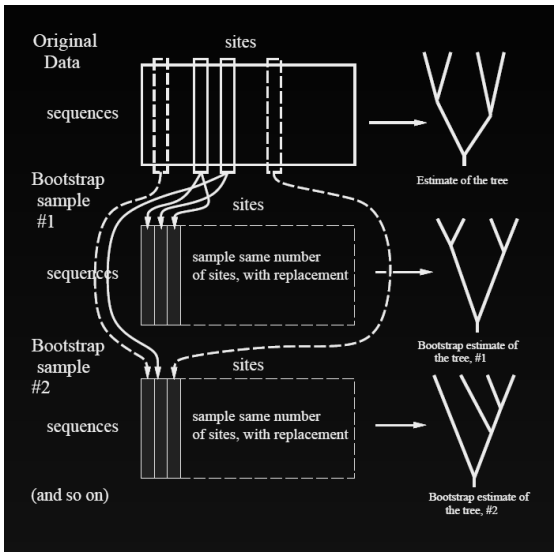
Step 4: validazione

PROBLEMA: Come validare un albero filogenetico: è affidabile?

- ⇨ Soluzione più utilizzata: il bootstrap
  - ⇨ Essenzialmente testa se il dataset supporta l'albero trovato o se esso non è altro che vincitore marginale rispetto ad alternative più o meno equivalenti
- ⇨ IDEA del bootstrap
  - ⇨ Vengono creati M nuovi data set campionando casualmente N colonne (con rimpiazzo)
    - ⇨ in questo modo in ogni dataset generato contiene lo stesso insieme di specie, con alcuni dei nucleotidi duplicati e con altri rimossi
    - ⇨ Si assume che ogni sito sia evoluto indipendentemente dato l'albero
  - ⇨ Per ogni data set viene costruito l'albero filogenetico

28

Step 4: validazione

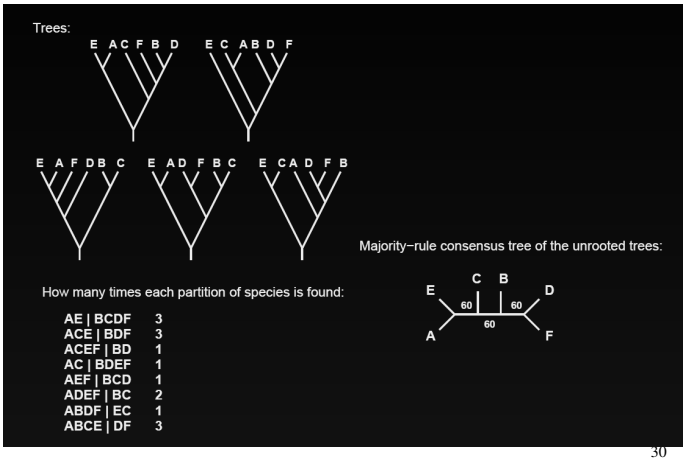


29

Step 4: validazione

⇨ Alla fine della procedura

⇨ Viene calcolata la frequenza con cui ogni sottogruppo dell'albero viene ripetuta



30

Questa indica la robustezza di un raggruppamento (>95% è affidabile)