Curs 9.

Analiza filogenetică. Predicția arborilor filogenetici

Biblio: cap. 10 din “An introduction to Bioinformatics algorithms”, N.Jones, P. Pevzner

Cap 7 din “Algorithms in Bioinformatics: A Practical Introduction”, W.K. Sung

Analiza filogenetica

- Scopul analizei
- Arbori filogenetici
- Metode de construire

Scopul analizei filogenetice

- Permite obținerea de informații privind procesele de evoluție

- Se bazează pe vizualizarea relațiilor de evoluție folosind structuri arborescente:
  - Relațiile de evoluție corespund unor ramuri în arbori
  - Lungimea ramurilor reflectă gradul de disimilaritate între entitățile analizate (de exemplu secvențe de aminoacizi)

- Construirea arborelui se bazează pe:
  - Construirea unei matrici de distanțe sau alinierea multiplă a secvențelor
  - Gruparea succesivă a secvențelor similare

Scopul analizei filogenetice

- Este utilă în urmărirea schimbărilor ce intervin în speciile care evoluează rapid (de exemplu, viruși). De exemplu în cazul virușilor ce induc gripa este de interes să se poată:
  - Studia schimbarea rapidă a genelor
  - Prezice structura viitoare a virusului
  - Produce un vaccin corespunzător

Scopul analizei filogenetice

- Etapele de evoluție sunt ilustrate printr-o structură arborescentă

- Nodurile frunză din structura arborescentă pot corespunde unor:
  - Organisme
  - Gene
  - Secvențe

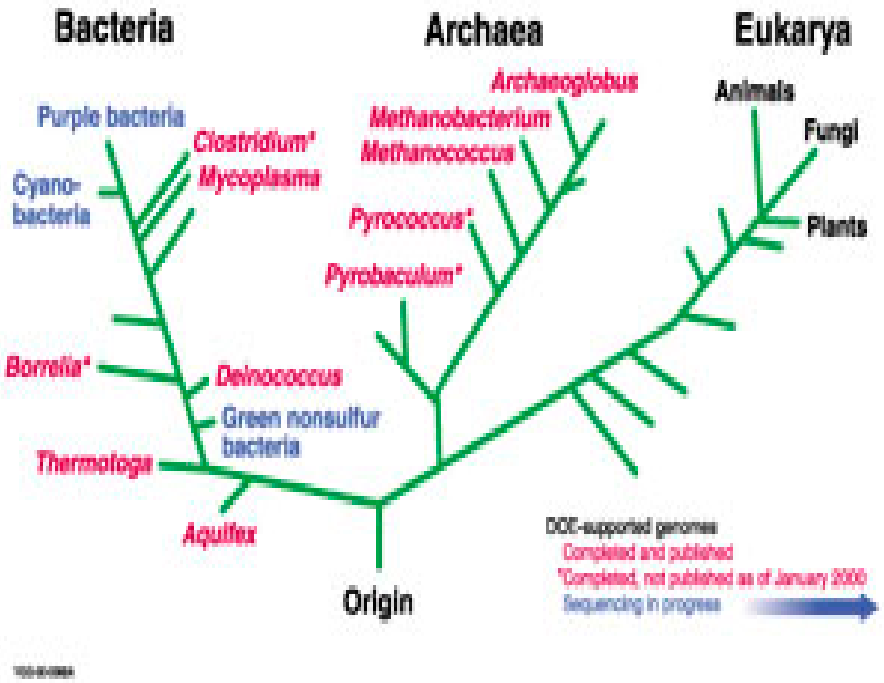


Image: http://microbialgenome.org/primer/tree.html

Arbori filogenetici

- Interpretarea unui arbore filogenetic:
  - Nodurile frunză reprezintă entitățile (speciile) analizate (denumite “taxa” sau “Operational Taxonomic Units” (OTU))
  - Nodurile interne reprezintă specii ancestrale (uneori ipotetice) iar ramurile din arbore exprimă relații între entitățile analizate
  - Entitățile care au un strămoș comun vor aparține aceluiași subarbore

Arbori filogenetici

- In interpretarea arborilor filogenetici poate conta nu doar topologia ci și lungimea ramurilor. In funcție de semnificația lungimilor ramurilor există diferite tipuri de arbori:

  - *Arbori scalați:* lungimile muchiilor sunt proporționale cu diferențele (numărul de mutații sau timpul estimat pentru evoluția dintr-o stare în alta) dintre entitățile corespunzătoare nodurilor vecine

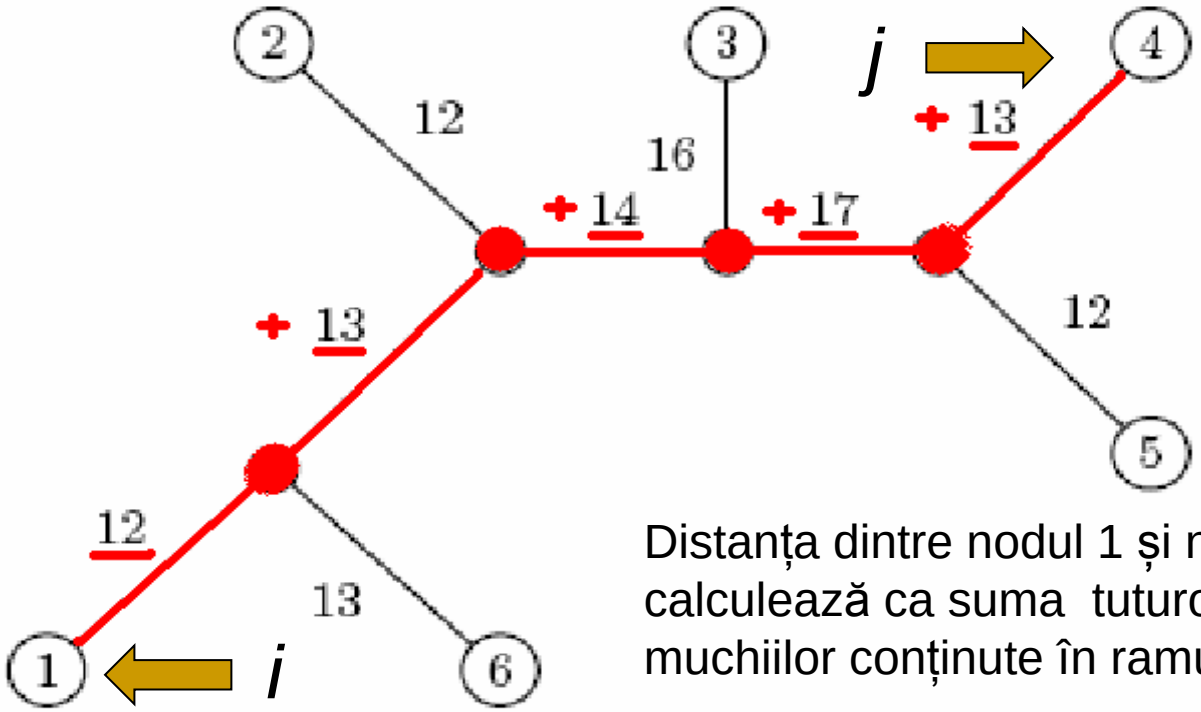
  - *Arbori aditivi:* lungimea totală a unei ramuri (suma lungimilor muchiilor) care leagă două noduri este corelată cu suma diferențelor cumulate de-a lungul etapelor de evoluție dntre entitățile corespunzătoare celor două noduri

  - *Arbori nescalați*: lungimile ramurilor nu poartă informație

Obs: în cazul în care entitățile din noduri sunt gene specificate prin secvențe ADN distanțele dintre ele se calculează pe baza distanței Hamming (secvențe aliniate) sau pe baza distanței de editare (secvențe nealiniate)

Arbori filogenetici

- Exemplu de arbore aditiv:

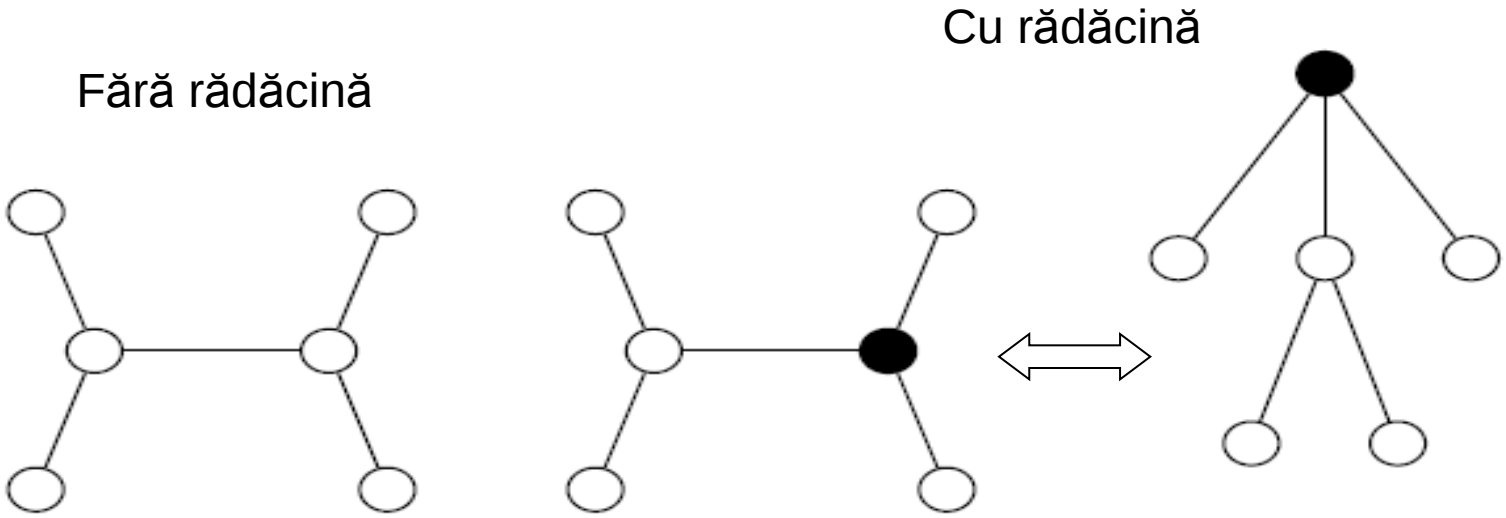


Distanța dintre nodul 1 și nodul 4 se calculează ca suma tuturor etichetelor muchiilor conținute în ramură

Arbori filogenetici

Tipuri de structuri arborescente:

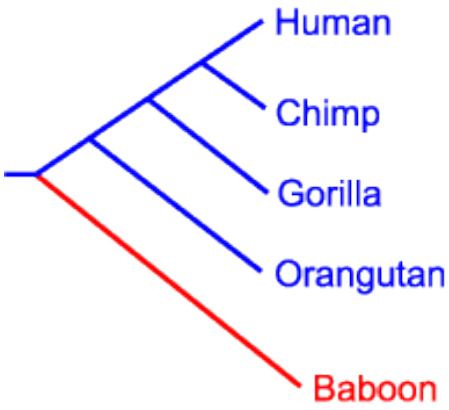
- Cu rădăcină (rooted tree): este pus în evidență un strămoș comun
- Fără rădăcină (unrooted tree): indică relația de evoluție între entități fără însă a marca un unic strămoș comun

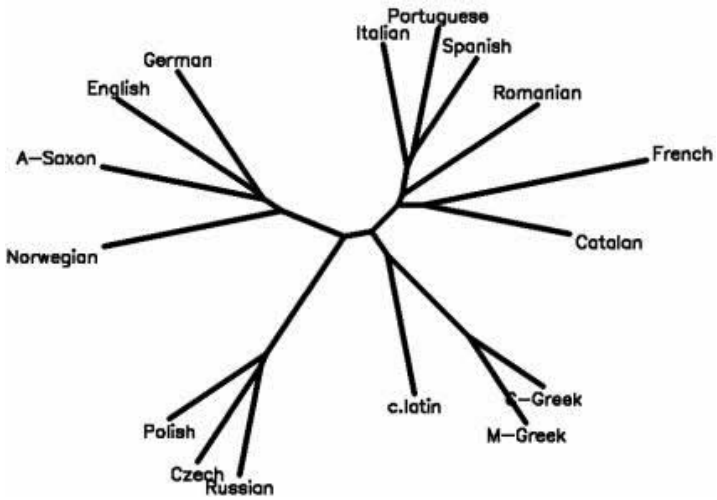


Arbori filogenetici

Exemple:

- Arbore cu rădăcină
- Arbore fără rădăcină



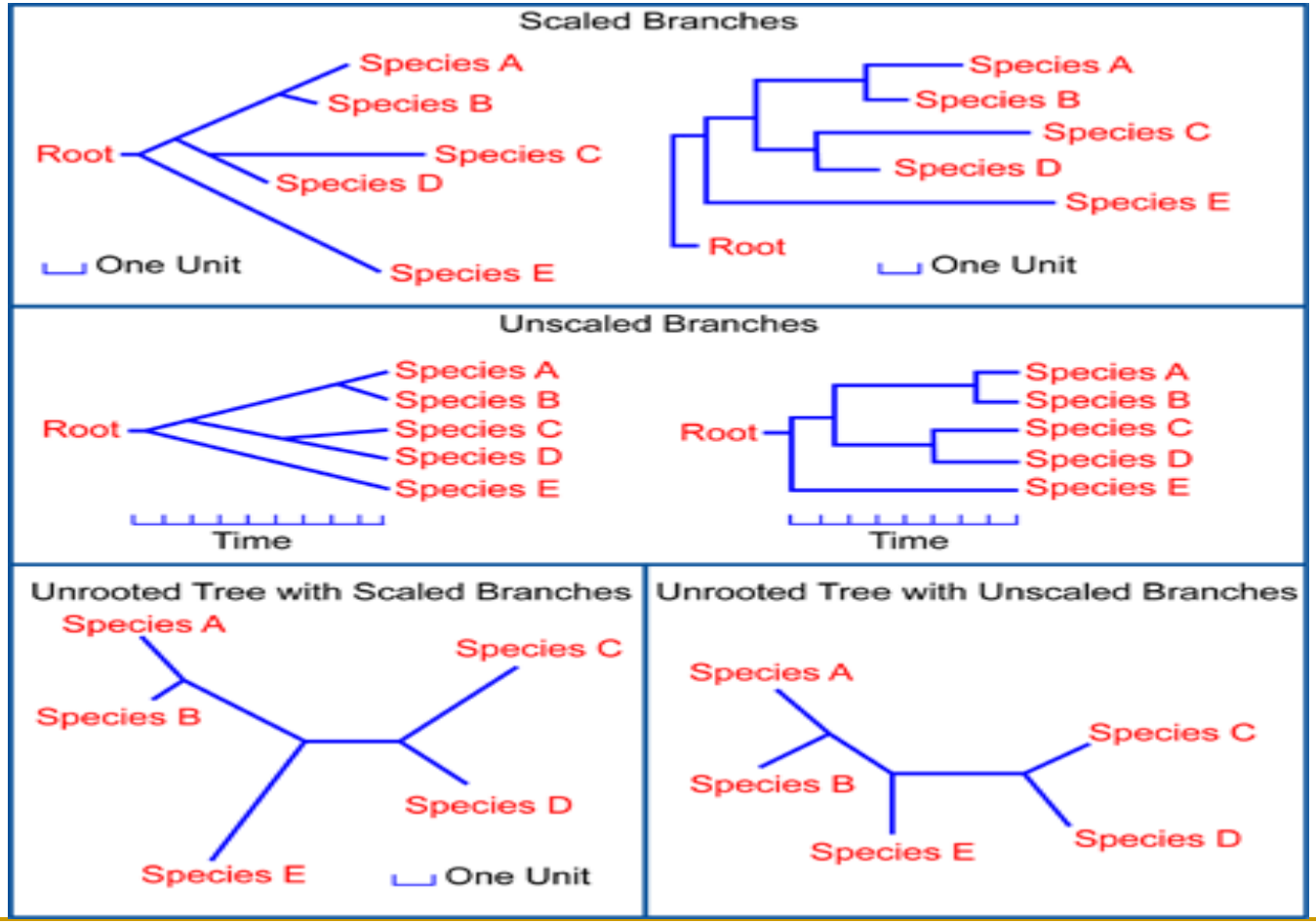


http://www.ncbi.nlm.nih.gov/About/primer/phylo.html

http://www.shef.ac.uk/english/language/quantling/images/quantling1.jpg

Arbori filogenetici

Variante de vizualizare:



Metode de construire a arborilor filogenetici

Construirea unui arbore filogenetic are ca scop identificarea structurii arborescente care reflectă procesul de evoluție pornind de la setul de specii (secvențe).

Tipuri de metode:

- Bazate pe distanțe (Distance based)
- Bazate pe minimizarea structurii (Maximum parsimony)
- Bazate pe metoda verosimilității maxime (Maximum likelihood)

Metode bazate pe distante

- Pornesc de la matricea de distanțe care conține pe linia i coloana j o măsură a disimilarității dintre secvențele s_i și s_j
- Matricea de distanțe fiind simetrică se reține de regulă doar partea superior (sau inferior) triunghiulară
- Proprietățile matricii de distanțe depind de tipul măsurii de disimilaritate folosite (aceasta nu este întotdeauna o metrică în sens matematic – de exemplu dacă se folosesc matrici de scor specifice secvențelor de aminoacizi măsura obținută nu satisface proprietatea triunghiului)

Metode bazate pe distante

- Variante de calcul a măsurii de disimilaritate:
  - Numărul de poziții diferite în aliniere (distanța Hamming) – se folosește în cazul secvențelor aliniate
  - Distanta Jukes-Cantor (varianta ajustată a distanței Hamming):

    $d(s_i,s_j)=-3/4 \ln(1-4/3*\text{nr poziții diferite/lungime secvențe})$

  - Numărul de operații necesare (înlocuire, eliminare, inserție) pentru a transforma o secvență în cealaltă secvență (distanța de editare) – se folosește în cazul secvențelor care nu sunt neapărat aliniate

Metode bazate pe distante

Problema construirii arborilor filogenetici:

Pornind de la matricea de distanțe dintre secvențe să se construiască un arbore în care topologia și lungimile ramurilor să exprime similaritățile dintre secvențe (construirea unui arbore care să se potrivească cât mai bine cu o matrice de distanțe)

Intrare: matrice de distanțe între secvențe ($D=D_{ij}$, i=1..n, j=1..n)

Ieșire: arbore T cu proprietatea că pentru orice pereche (i,j) are loc:

$$d_{ij}(T)=D_{ij}$$

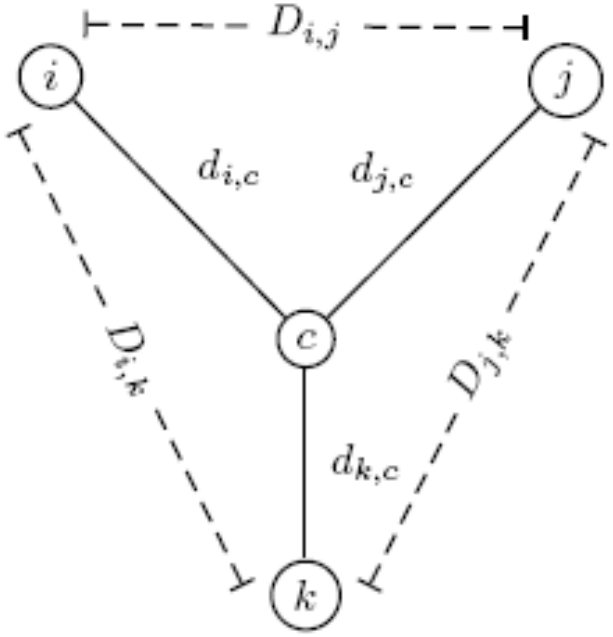
Distanța dintre noduri în arbore = suma etichetelor muchiilor care trebuie parcurse pentru a ajunge de la un nod la celălalt

Distanța dintre secvențe

Metode bazate pe distante

Caz particular: 3 secvențe

Se introduce un nod rădăcină, c, și 3 noduri frunză corespunzătoare celor 3 secvențe



Etichetele muchiilor trebuie să satisfacă

$d_{ic} + d_{jc} = D_{ij}$

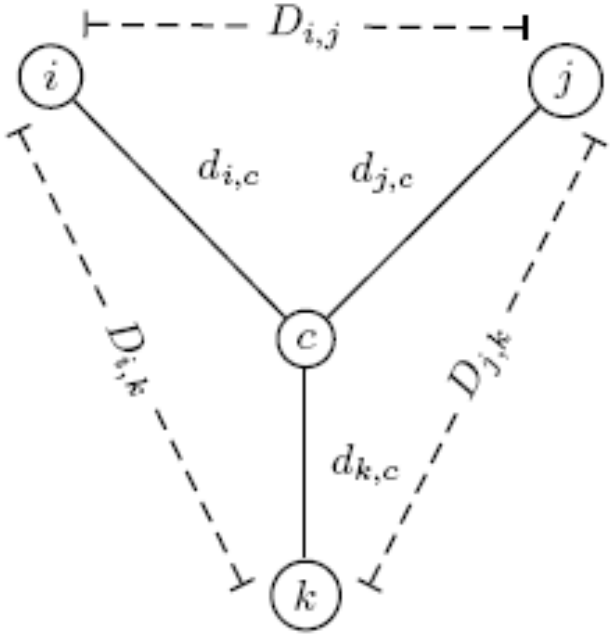
$d_{ic} + d_{kc} = D_{ik}$

$d_{jc} + d_{kc} = D_{jk}$

Metode bazate pe distante

Caz particular: 3 secvențe

Se introduce un nod rădăcină, c, și 3 noduri frunză corespunzătoare celor 3 secvențe



Etichetele muchiilor trebuie să satisfacă

$d_{ic} + d_{jc} = D_{ij}$

$d_{ic} + d_{kc} = D_{ik}$

$d_{jc} + d_{kc} = D_{jk}$

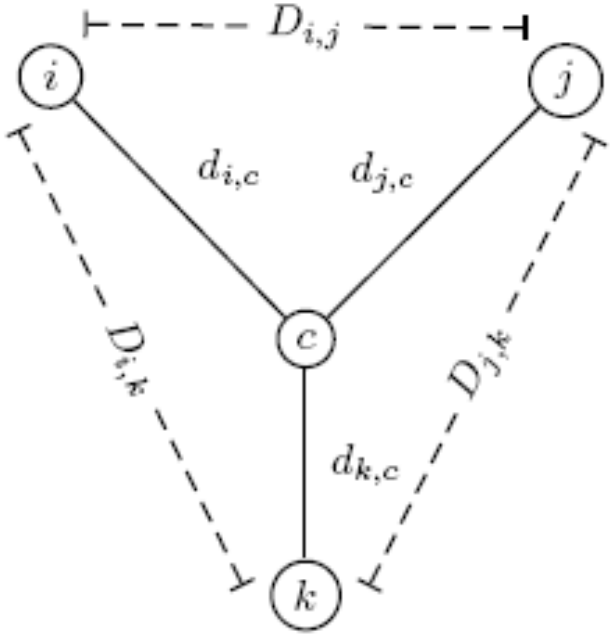
Prin rezolvarea sistemului:

$d_{ic} = (D_{ij} + D_{ik} – D_{jk})/2$
$d_{jc} = (D_{ij} + D_{jk} – D_{ik})/2$
$d_{kc} = (D_{ki} + D_{kj} – D_{ij})/2$

Metode bazate pe distante

Caz particular: 3 secvențe

Se introduce un nod rădăcină, c, și 3 noduri frunză corespunzătoare celor 3 secvențe



Cum se poate extinde în cazul a mai mult de 3 noduri frunză ?

Metode bazate pe distante

Exemplu. Considerăm cazul a 4 secvențe ADN și matricea distanțelor Hamming dintre ele:

S1=ATCC

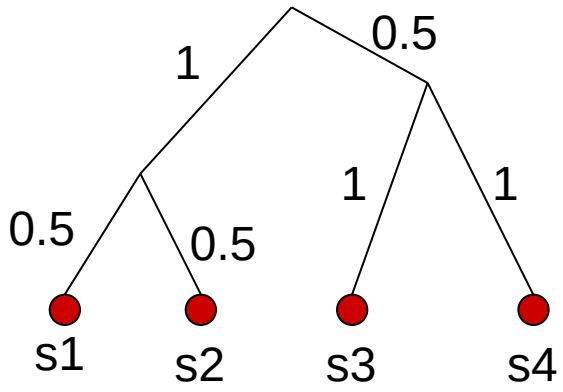
S2=ATGC

S3=TATC

S4=TCAC

	s1	s2	s3	s4
s1	0	1	3	3
s2		0	3	3
s3			0	2
s4				0

Arbore filogenetic



Distanțe în arbore:

d12=0.5+0.5=1

d13=d14=0.5+1+0.5+1=3

d23=d24=0.5+1+0.5+1=3

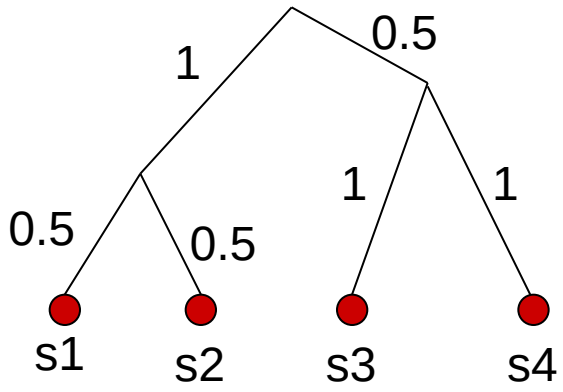
d34=1+1=2

Metode bazate pe distante

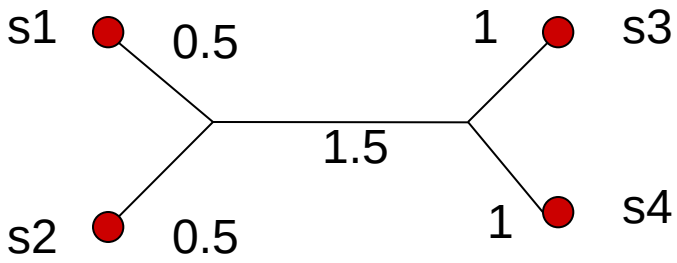
Observatie. Nu este întotdeauna posibil să se construiască un astfel de arbore. Daca însă matricea de distanțe este aditivă atunci este posibil.

O matrice de distanțe este aditivă dacă și numai dacă pentru orice 4 elemente (i,j,k și l) două dintre sumele distanțelor: $d_{ij}+d_{kl}$, $d_{ik}+d_{jl}$, $d_{il}+d_{jk}$ sunt egale între ele și mai mari sau egale cu a treia

Arbore filogenetic (cu rădăcină)



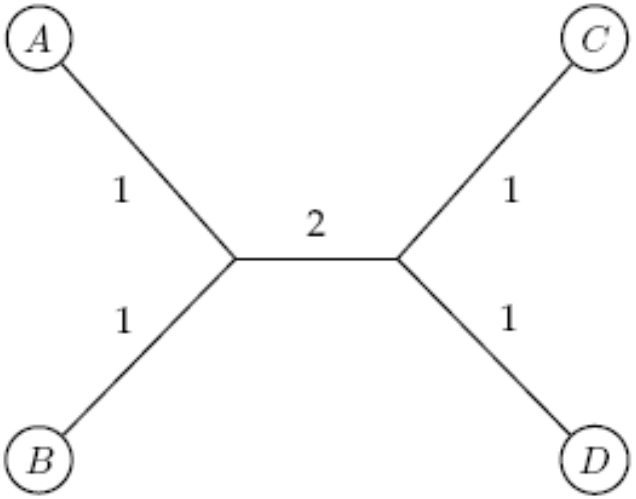
Arbore filogenetic (fără rădăcină)



Metode bazate pe distante

Matrice aditivă

	A	B	C	D
A	0	2	4	4
B	2	0	4	4
C	4	4	0	2
D	4	4	2	0



d(A,B)+d(C,D)=4, d(A,C)+d(B,D)=8, d(A,D)+d(B,C)=8

Matrice care nu este aditivă

	A	B	C	D
A	0	2	2	2
B	2	0	3	2
C	2	3	0	2
D	2	2	2	0

?

d(A,B)+d(C,D)=4, d(A,C)+d(B,D)=4, d(A,D)+d(B,C)=5

Metoda UPGMA

- UPGMA = Unweighted Pair Group Method with Arithmetic Mean
- Este cel mai simplu algoritm de construire a unui arbore filogenetic
- Este similar algoritmilor aglomerativi de clustering (vezi curs 10)
- Se caracterizează prin:
  - Calculează distanțele dintre clustere folosind media distanțelor (*average link*)
  - Asignează o înălțime (*height)* fiecărui nod din arbore; lungimea unei muchii va fi dată de diferența dintre înălțimile nodurilor care formează muchia

Obs:

- Distanța de la rădăcină la fiecare nod frunză este aceeași
- Se bazează pe ipoteza că în toate speciile corespunzătoare frunzelor, mutațiile se acumulează cu aceeași rată (ceas molecular constant) – este o ipoteză restrictivă

Metoda UPGMA

Algoritmul UPGMA (algoritm iterativ)

Inițializare:

Asignează fiecare x_i la clusterul său C_i (pentru fiecare cluster se definește un nod frunză); toate nodurile frunză au asignată înălțimea 0: $h(C_i)=0$

Repetă:

1. Determină cei mai apropiați clusteri C_i *și* C_j (cei pt care $d_{ij}=d(C_i,C_j)$ este minimă)
2. Reunește clusterii și construiește $C_k = C_i \cup C_j$
3. Adaugă un nod C_k care conectează pe C_i, C_j și îl plasează la înălțimea $h(C_k)= d_{ij}/2$; etichetează muchia (C_k,C_i) cu $h(C_k)$-$h(C_i)$ și muchia (C_k,C_j) cu $h(C_k)$-$h(C_j)$
4. Sterge C_i *și* C_j din lista de clusteri și îl adaugă pe C_k
5. Actualizează matricea de distanțe (va avea cu o linie și o coloană mai puțin)

Condiție de terminare:

Când se ajunge la un singur cluster

Metoda UPGMA

Distanța dintre doi clusteri se poate calcula în mai multe moduri:

$$d_A(C_i, C_j) = \frac{1}{card(C_i) \cdot card(C_j)} \sum_{x \in C_i, y \in C_j} d(x, y) \quad \text{(average link)}$$

$$d_C(C_i, C_j) = \max(d(x, y) \mid x \in C_i, y \in C_j) \text{ (complete link)}$$

$$d_S(C_i, C_j) = \min(d(x, y) \mid x \in C_i, y \in C_j) \text{ (single link)}$$

In actualizarea matricii de distanțe trebuie calculată distanța de la reuniunea a doi clusteri C_i si C_j la un alt cluster C_l cunoscându-se deja distanța corespunzătoare fiecărei perechi de clusteri. Se poate evita calculul tuturor distanțelor între elementele clusterilor folosind:

$$d_A(C_i \cup C_j, C_l) = \frac{d_A(C_i, C_l) card(C_i) + d_A(C_j, C_l) card(C_j)}{card(C_i) + card(C_j)}$$

$$d_C(C_i \cup C_j, C_l) = \max\{d_C(C_i, C_l), d_C(C_j, C_l)\}$$

$$d_S(C_i \cup C_j, C_l) = \min\{d_S(C_i, C_l), d_S(C_j, C_l)\}$$

Metoda UPGMA

Ordin complexitate:

- Varianta directă de implementare are ordinul $O(n^3)$ (în cazul a n specii/secvențe)
- Varianta optimizată: $O(n^2)$ (propusă în Gronau și Moran, 2006 - Optimal Implementations of UPGMA and Other Common Clustering Algorithms)
  - Idee: în loc să se determine la fiecare etapă perechea (C_i,C_j) aflată la distanța minimă (minimul global din matricea curentă de distanțe – a cărui determinare este de cost $O(n^2)$) se determină perechea care satisface un minim local (cost determinare: O(n))

Metoda UPGMA

Exemplu:

	A	B	C	D	E
A	–	4	2	8	6
B	–	–	4	10	8
C	–	–	–	4	4
D	–	–	–	–	2
E	–	–	–	–	–

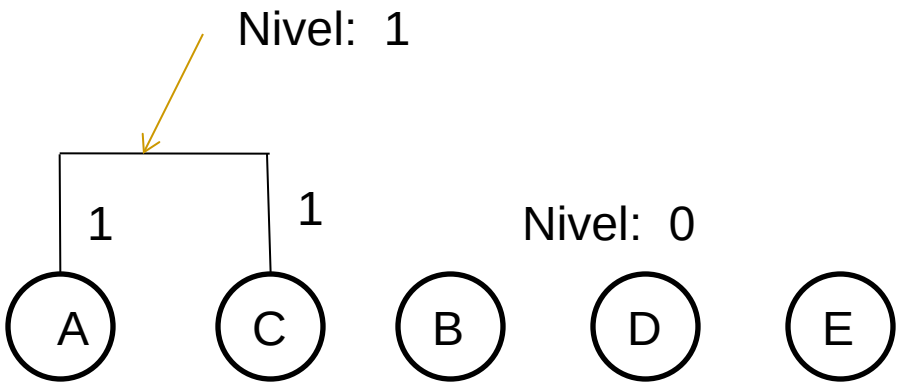
Metoda UPGMA

Exemplu:

	A	B	C	D	E
A	–	4	2	8	6
B	–	–	4	10	8
C	–	–	–	4	4
D	–	–	–	–	2
E	–	–	–	–	–

Etapa 1:

- determinarea unei perechi aflată la distanța cea mai mică: A și C
- Gruparea nodurilor selectate într-un cluster
- Asignarea de etichete muchiilor care unesc cele două noduri



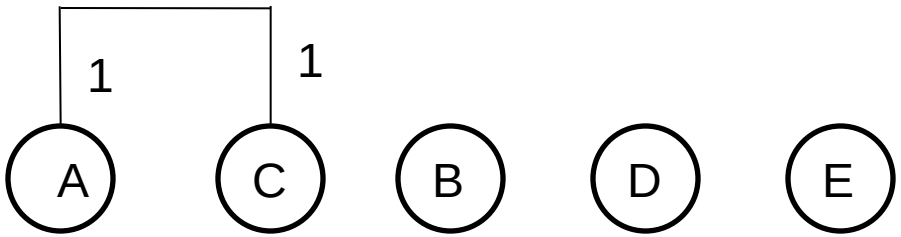
Metoda UPGMA

Exemplu:

	$\{A,C\}$	B	D	E
$\{A,C\}$	–	4	6	5
B	–	–	10	8
D	–	–	–	2
E	–	–	–	–

Etapa 1:

- determinarea unei perechi aflată la distanța cea mai mică: A și C
- Gruparea nodurilor selectate într-un cluster
- Asignarea de etichete muchiilor care unesc cele două noduri
- Modificarea matricii de distanțe



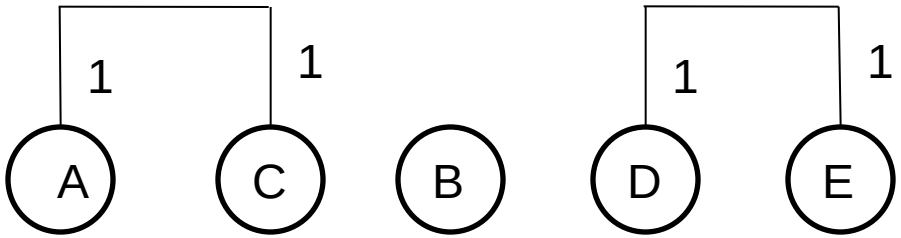
Metoda UPGMA

Exemplu:

	$\{A,C\}$	B	D	E
$\{A,C\}$	–	4	6	5
B	–	–	10	8
D	–	–	–	2
E	–	–	–	–

Etapa 2:

- determinarea unei perechi aflată la distanța cea mai mică: D și E
- Gruparea nodurilor selectate într-un cluster
- Asignarea de etichete muchiilor care unesc cele două noduri



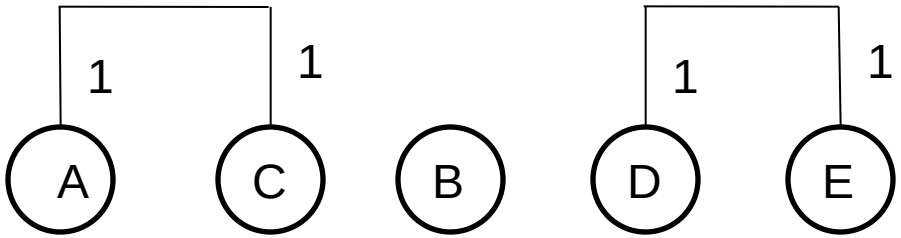
Metoda UPGMA

Exemplu:

	$\{A,C\}$	$\{D,E\}$	B
$\{A,C\}$	–	5.5	4
$\{D,E\}$	–	–	9
B	–	–	–

Etapa 2:

- determinarea unei perechi aflată la distanta cea mai mica: D și E
- Gruparea nodurilor selectate într-un cluster
- Asignarea de etichete muchiilor care unesc cele două noduri
- Modificarea matricii de distanțe



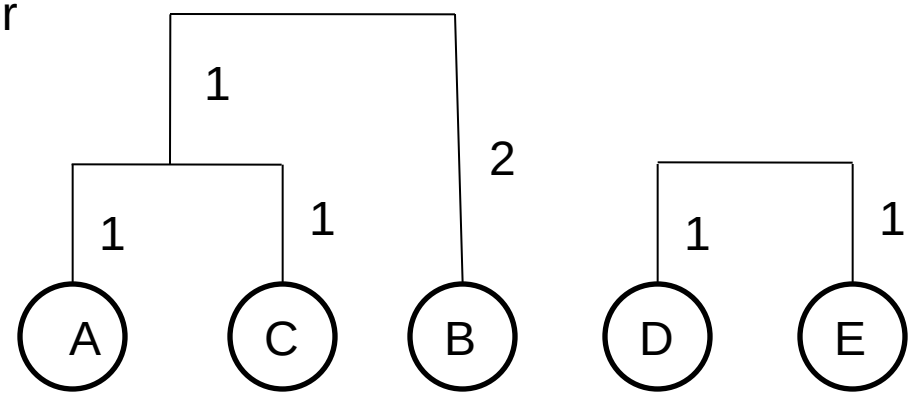
Metoda UPGMA

Exemplu:

	$\{A,C\}$	$\{D,E\}$	B
$\{A,C\}$	–	5.5	4
$\{D,E\}$	–	–	9
B	–	–	–

Etapa 3:

- determinarea unei perechi aflată la distanța cea mai mică: {A,C} si B
- Gruparea nodurilor selectate într-un cluster
- Asignarea de etichete muchiilor care unesc cele două noduri



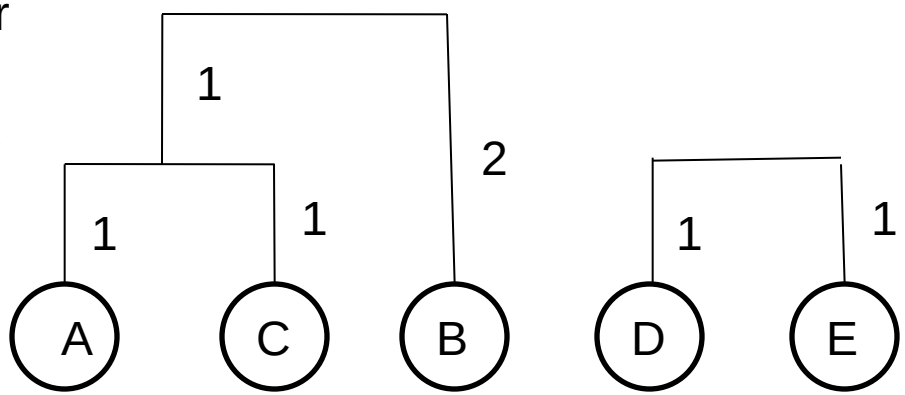
Metoda UPGMA

Exemplu:

	$\{A,C,B\}$	$\{D,E\}$
$\{A,C,B\}$	–	7.25
$\{D,E\}$	–	–

Etapa 3:

- determinarea unei perechi aflată la distanța cea mai mică: {A,C} și B
- Gruparea nodurilor selectate într-un cluster
- Asignarea de etichete muchiilor care unesc cele două noduri
- Modificarea matricii de distanțe



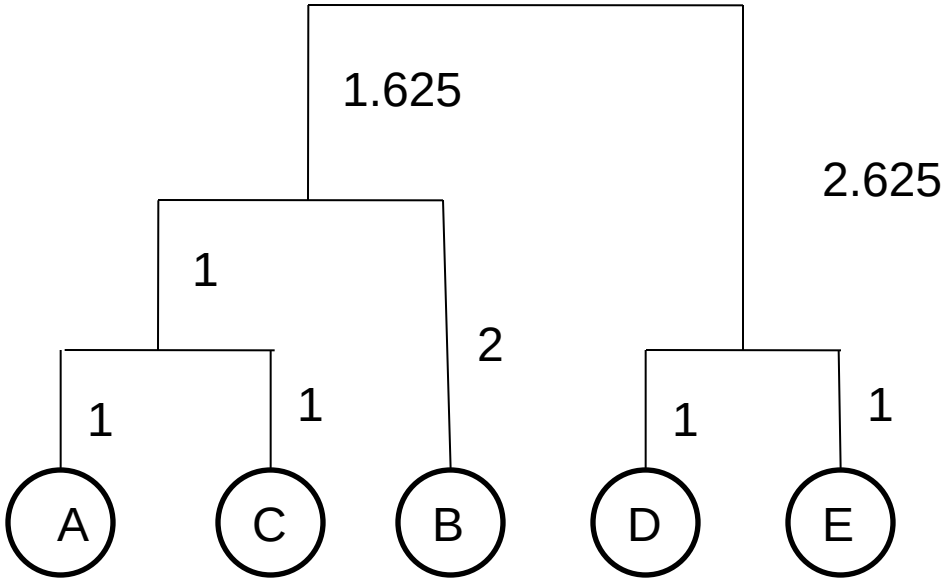
Metoda UPGMA

Exemplu:

	$\{A,C,B\}$	$\{D,E\}$
$\{A,C,B\}$	–	7.25
$\{D,E\}$	–	–

Etapa 4:

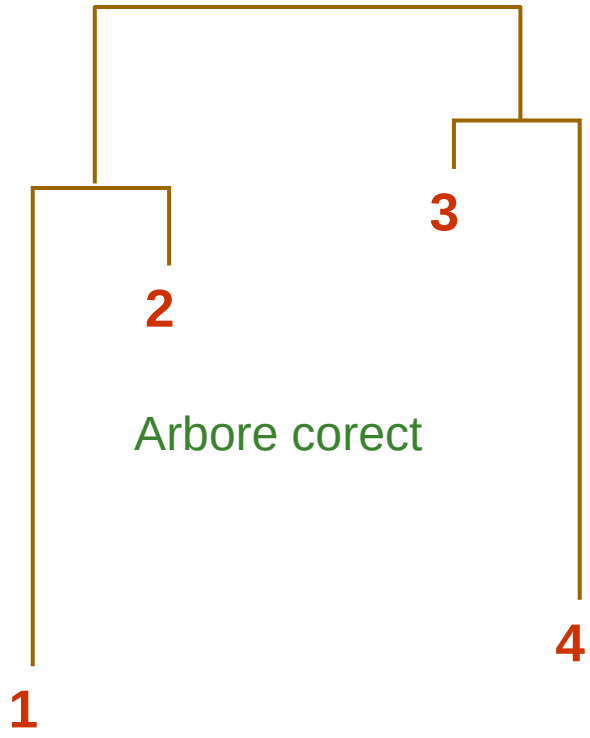
- Gruparea ultimelor două noduri rămase

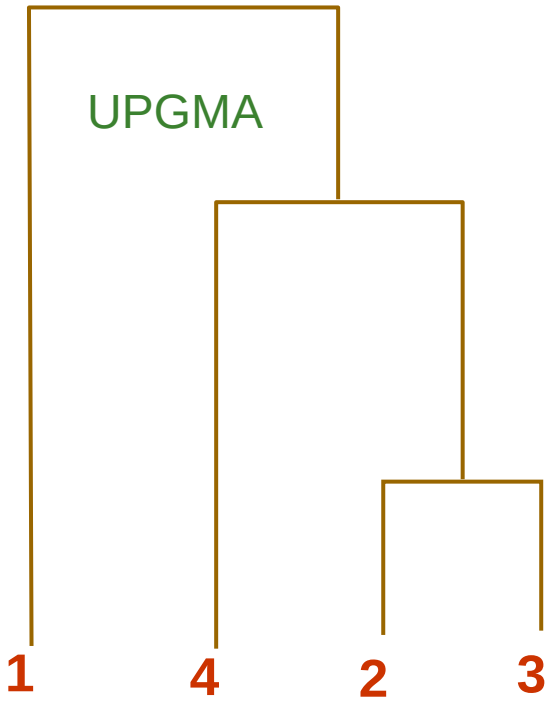


Metoda UPGMA

Obs. Dacă în procesul de evoluție mutațiile nu apar cu aceeași rată atunci algoritmul UPGMA nu conduce la arborele corect

Un algoritm care nu presupune că toate speciile evoluează cu aceeași rată este algoritmul Neighbor-Joining [Saitou, Nei – 1987]





Metoda neighbor-joining

Idee:

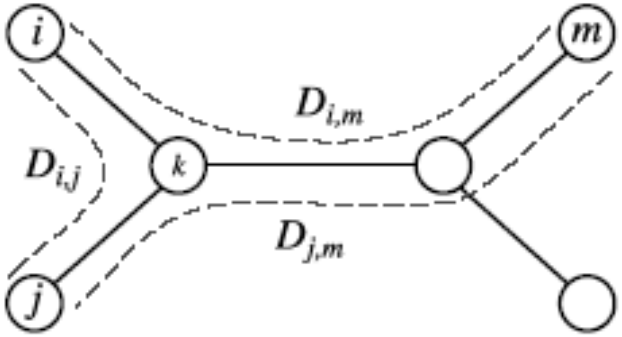
- Două noduri frunză sunt considerate vecine dacă în arborele filogenetic ar avea același părinte

- Se aleg două noduri frunză vecine i și j și se înlocuiesc în lista nodurilor frunză cu nodul k

- Distanța de la noul nod k la un nod frunză m se se calculează ca:

$$d_{km}=(d_{im}+d_{jm}-d_{ij})/2$$

- Procesul de “fuziune” al nodurilor frunză vecine continuă până când se ajunge la o singură perech de noduri vecine



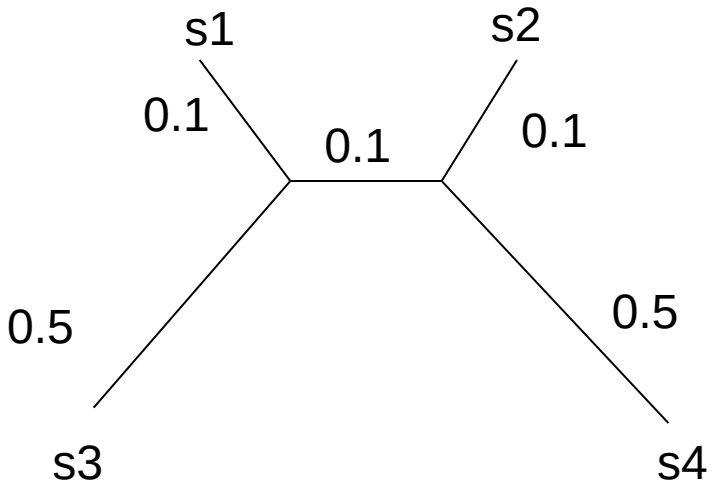
Metoda neighbor-joining

Determinarea nodurilor vecine nu este o problemă simplă:

- Nu este suficient să se aleagă cele mai apropiate două elemente (din punctul de vedere al distanței dintre ele)

- Trebuie alese elemente apropiate care sunt în același timp îndepărtate de celelalte (in raport cu numărul muchiilor care le separă)

Exemplu:



	$s1$	$s2$	$s3$	$s4$
$s1$	–	0.3	0.6	0.7
$s2$	–	–	0.7	0.6
$s3$	–	–	–	1.1
$s4$	–	–	–	–

Nodurile vecine sunt s1 și s3 respectiv s2 și s4 (au același părinte)

Metoda neighbor-joining

Cum se poate modifica matricea de distanțe astfel încât nodurile aflate la distanța minimă să fie noduri vecine în arbore ?

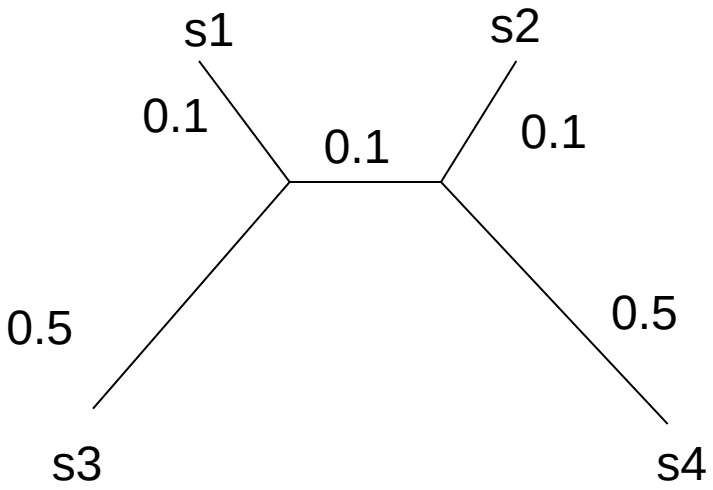
Idee:

- e necesar să se compenseze influența muchiilor lungi
- din fiecare element al matricii de distanțe se scade media distanțelor dintre nodurile corespunzătoare elementului din matrice și toate celelalte noduri

$$D_{ij} = d_{ij} - (r_i + r_j)$$

$$r_i = \frac{1}{N-2} \sum_{k \in L} d_{ik}$$

L este lista nodurilor frunză
N este numărul de noduri frunză=numărul de secvențe



Metoda neighbor-joining

Modificarea matricii de distanțe

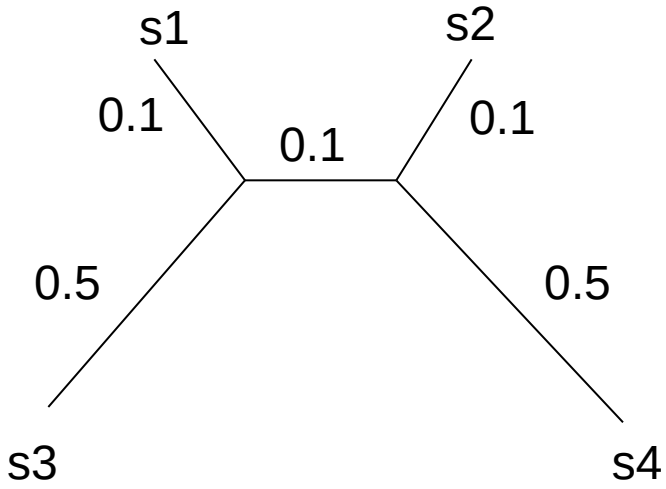
$$D_{ij} = d_{ij} - (r_i + r_j)$$

$$r_i = \frac{1}{N-2} \sum_{k \in L} d_{ik}$$

	s_1	s_2	s_3	s_4
s_1	–	0.3	0.6	0.7
s_2	0.3	–	0.7	0.6
s_3	0.6	0.7	–	1.1
s_4	0.7	0.6	1.1	–

Matrice Inițială

(Minim pt noduri ce nu sunt vecine)



$r_1 = 1.6/2 = 0.8$ $r_2 = 1.6/2 = 0.8$

$r_3 = 2.4/2 = 1.2$ $r_4 = 2.4/2 = 1.2$

	s_1	s_2	s_3	s_4
s_1	–	−1.3	−1.4	−1.3
s_2	−1.3	–	−1.3	−1.4
s_3	−1.4	−1.3	–	−1.3
s_4	−1.3	−1.4	−1.3	–

Matrice modificată

Pentru nodurile vecine valoarea este minimă

Metoda neighbor-joining

Inițializare:

- Se definește lista de noduri frunză, L, constituită din secvențele de prelucrat
- Se inițializează arborele T cu lista de noduri frunză
- Se calculează și se ajustează matricea de distanțe

Repetă:

- Se alege perechea (i,j) din L pentru care valoarea ajustată a distanței, D_{ij} este minimă
- Se definește un nou nod k și se calculeaza $d_{km}=(d_{im}+d_{jm}-d_{ij})/2$ pentru fiecare m din L
- Se adaugă nodul k la T cu muchiile către nodurile i și j având lungimile

$$(d_{ij}+r_i-r_j)/2 \quad \text{si} \quad (d_{ij}+r_j-r_i)/2$$

- Se elimină i și j din L și se adaugă k la L

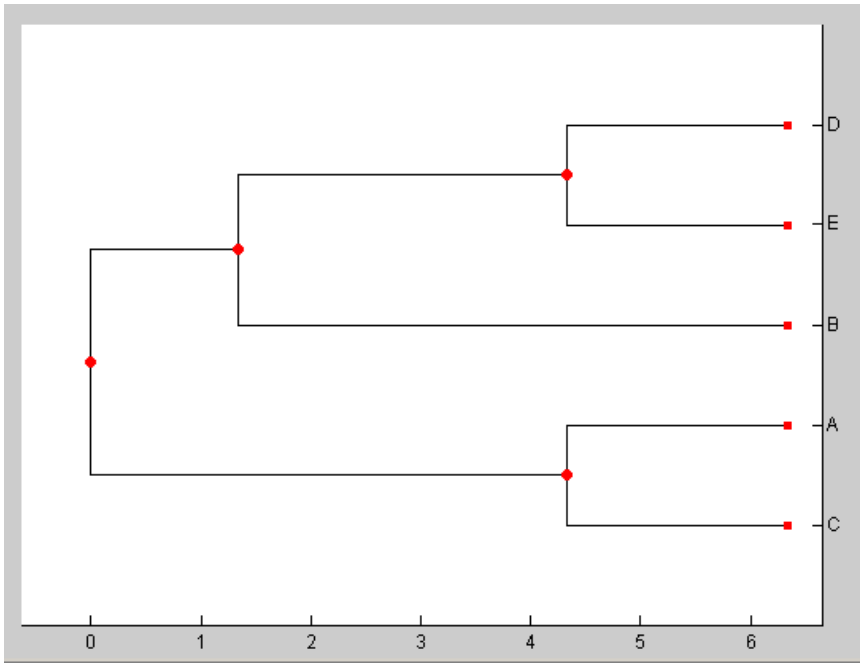
Conditie de terminare:

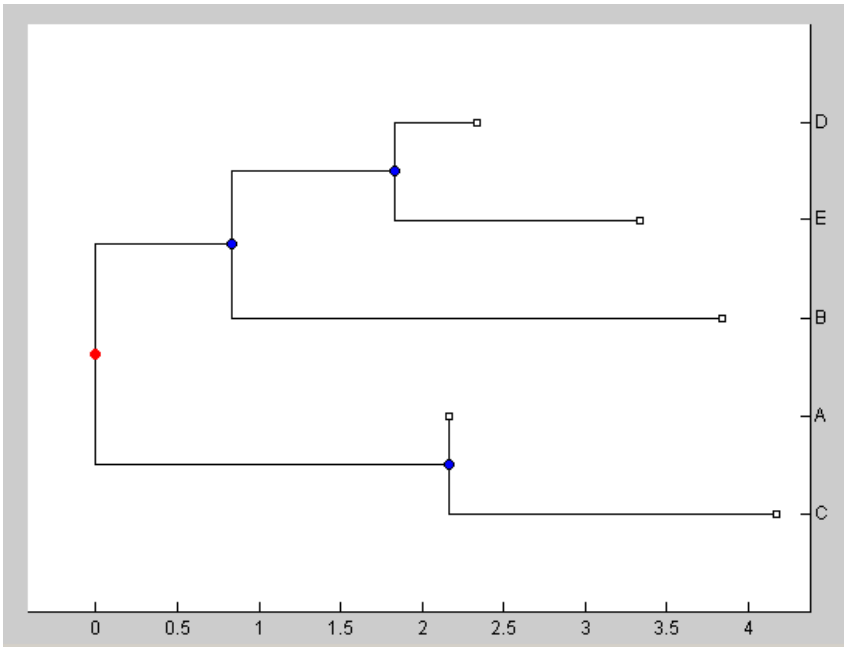
Când L conține doar două noduri (i si j) se adaugă muchia dintre ele având lungimea d_{ij}

Ordin complexitate $O(N^3)$

Metoda neighbor-joining

Diferența dintre un arbore construit cu UPGMA și unul construit cu Neighbor Joining (pentru aceeași matrice de distanțe – exemplul de la UPGMA, slide 25)





Matlab: seqlinkage, seqneighjoin

Neighbor-joining si ClustalW

Reminder: ClustalW este un algoritm de aliniere multiplă bazat pe o tehnică euristică de tip progresiv

Etapele de prelucrare in ClustalW

- **PairAlign:** Se realizează aliniere la nivelul perechilor de secvențe. Folosind scorurile corespunzătoare alinierilor se construiește matricea de distanțe
- **NJTree**: Se construiește un arbore de ghidare folosind algoritmul Neighbor-Join; arborele de ghidare indica ordinea în care sunt incluse secvențele în aliniere
- **Malign:** Se construiește alinierea multiplă care necesită:
  - Aliniere între două secvențe
  - Aliniere între o secvență și o altă aliniere (profil)
  - Aliniere între două alte alinieri (profile)

  Obs. In alinierea multiplă un gap o dată introdus rămâne permanent în aliniere.

Metode bazate pe minimizarea costului transformarilor

Specific:

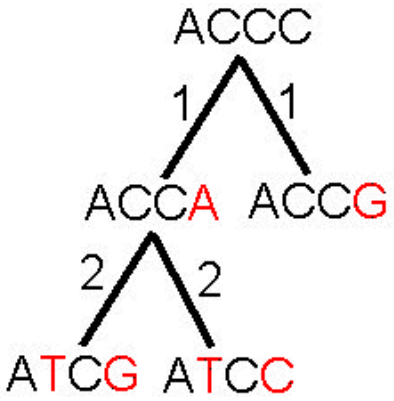
- Construiește arborele evolutiv prin minimizarea numărului de pași necesari pentru a genera variația observată în date
- Pentru fiecare poziție este identificat arborele ce necesită numărul cel mai mic de mutații pentru a se ajunge la variația observată în date
- Necesită în prealabil alinierea multiplă a secvențelor

Dezavantaje:

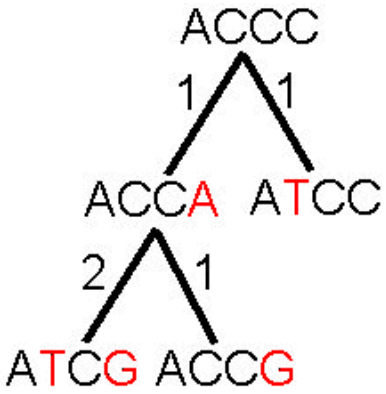
- Este un algoritm costisitor (obs: variantele exacte se bazează pe branch and bound; există și algoritmi aproximativi)
- Funcționează doar dacă există o similaritate semnificativă între secvențe

Metode bazate pe minimizarea costului transformarilor

Exemplu:



Scor=6



Scor=5 (arbore mai bun)

Instrumente software pentru analiza filogenetica

PHYLIP [http://evolution.genetics.washington.edu/phylip.html] - pachet free care include peste 30 de programe pentru analiza filogenetică

PAUP (Phylogenetic Analysis Using Parsimony) [http://paup.csit.fsu.edu/] – inițial a constat doar în algoritmi bazați pe minimizarea costului trasnformărilor; acum (PAUP*) conține mai multe metode de construire a arborilor filogenetici

BIONJ [http://www.atgc-montpellier.fr/bionj/binaries.php] – implementarea unei variante îmbunătățite a alg. Neighbor-Joining

CLUSTALW – program pentru aliniere multipla ce conține funcție pentru Neighbor Joining