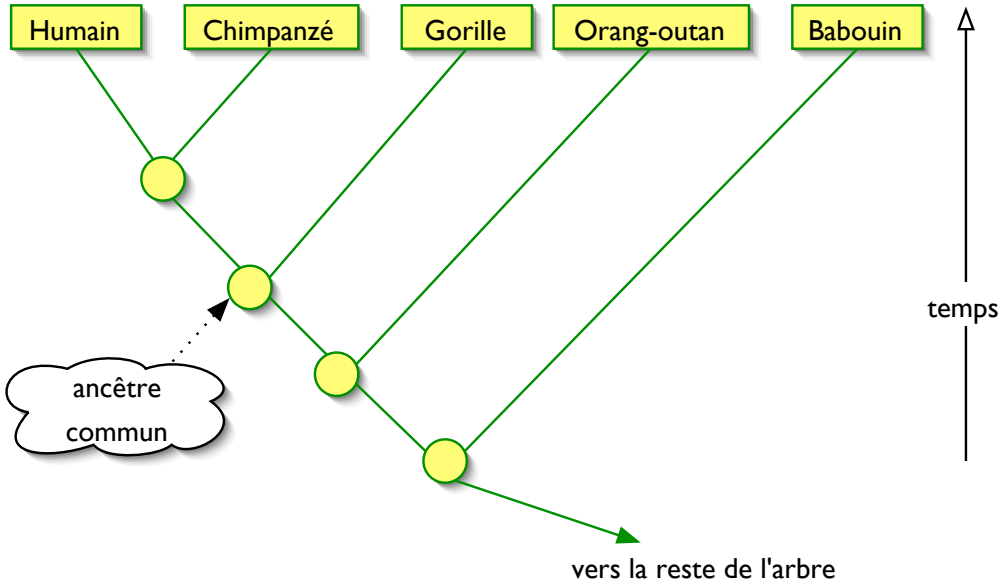
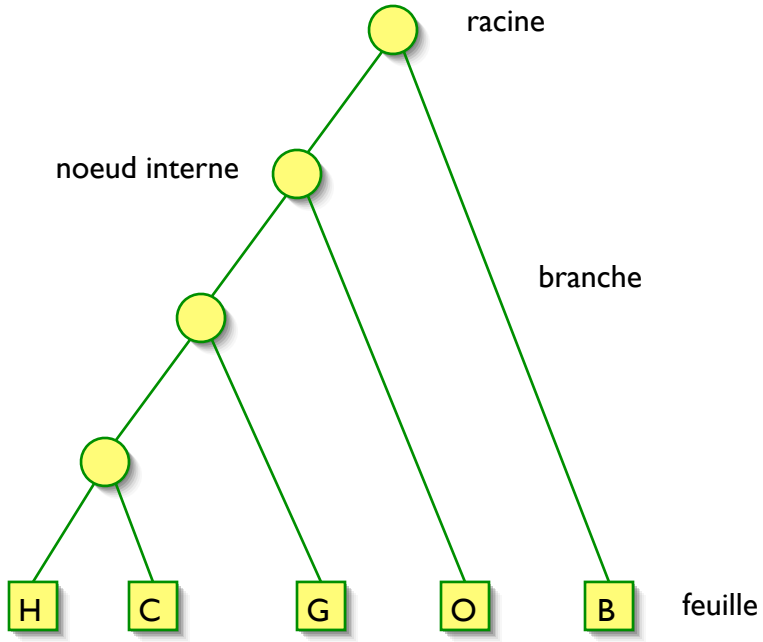
Phylogénies

Phylogénies

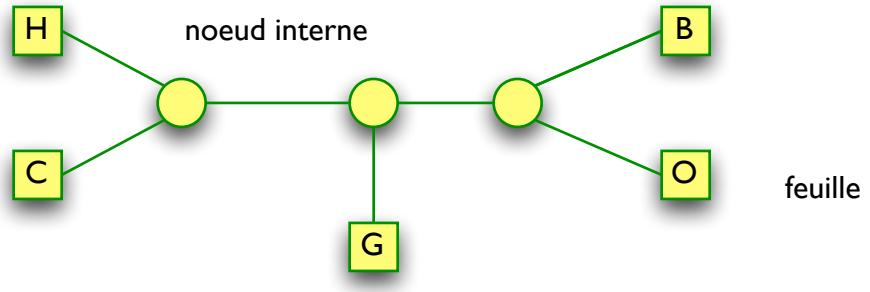
phylogénie ou arbre évolutif



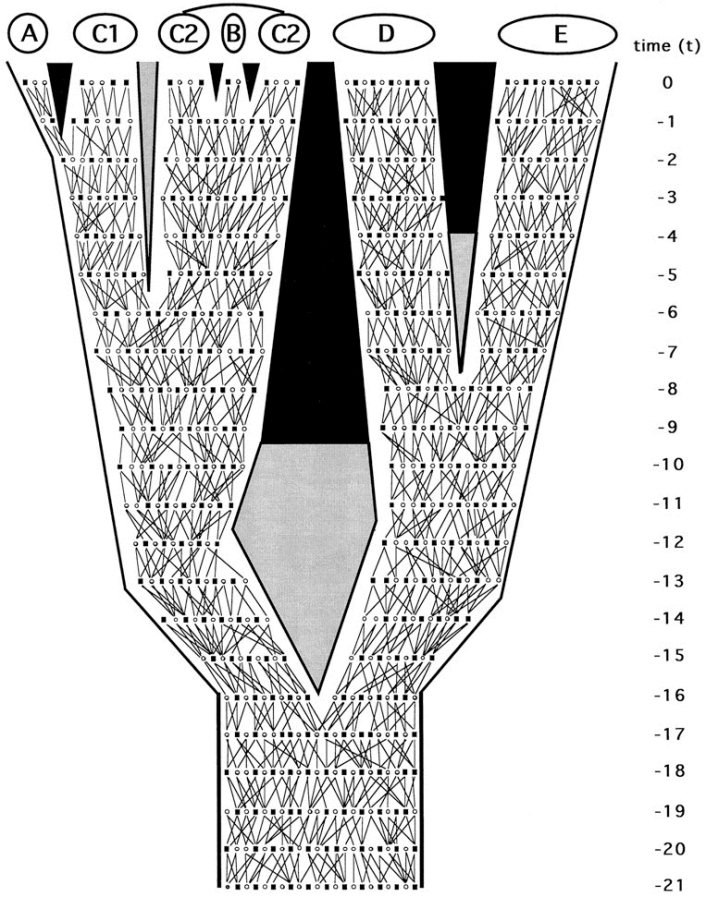
Arbre raciné



Arbre non-raciné



D’où vient l’arbre ?



Évolution de caractères/séquences

Séquence assosiée à chaque noeud : peut être des séquences moléculaires ou d'autres «caractères» (possède des ailes, présence de colonne vertébrale)

Évolution comme transmission d'une séquence ancestrale à partir de la racine vers les feuilles

Mutations introduites sur les branches

$\Rightarrow$ arbre sur les séquences (pas nécessairement le même que la phylogénie des espèces)

Mutations

Substitutions : $O(10^{-9})$ par site par an dans ADN de mammifères, cca. 10 fois plus grand en ADN mitochondrial)

- transitions : $A \leftrightarrow G$ ou $T \leftrightarrow C$ et transversions
- peut changer le codon ou non

Recombinations

Insertions et suppressions

Inversions

Maximum de parcimonie (MP)

minimiser le nombre total de changements de caractères sur les branches (valeur de parcimonie)

Problème 1 : calculer la valuer de parcimonie pour un arbre donné — PD

Problème 2 : trouver l'arbre qui minimise la parcimonie — NP-difficile

Nombre d'arbres non-racinés avec n feuilles : $1 \cdot 3 \cdot 5 \cdots (2n-5)$

Parcimonie 2

Problème 1 (la petite parcimonie)

Séquences [alignées] : $s_1, \ldots, s_n$, de longueur ℓ

Arbre : chaque s_i assignée à une feuille ; déterminer les séquences ancestrales s_j : $j > n$ assignées aux noeuds internes

Valeur de parcimonie sur une branche ij :
$\mathsf{VP}(ij) = \sum_{k=1}^{\ell} I\{s_i[k] \neq s_j[k]\}$ où $I\{A\}$ est l'indicateur de A (égal à 1 ssi A est vrai)

On veut minimiser $\sum_{ij} \mathsf{VP}(ij)$.

Parcimonie 3

Observation : $\sum_{ij} \mathrm{VP}(ij) = \sum_{k=1}^{\ell} \sum_{ij} I\{s_i[k] \neq s_j[k]$ donc il suffit de trouver un algorithme pour des séquences de longueur 1 est l'utiliser ℓ fois.

Programmation dynamique : calculer pour chaque noeud u
- $V(u)$: VP pour le sous-arbre raciné à u
- $V(u, a)$: VP pour le sous-arbre si u est assigné caractère a

Feuilles : $V(u) = 0$; $V(u, a) = 0$ si u est assigné a, $V(u, a) = \infty$ sinon

Récurrences pour noeud interne : $V(u, a) = \sum_v$ est un déscéndant $\min\{V(v) + 1, V(v, a)\}$; $V(u) = \min\{V(u, a)\}$

Distances

Modèle d'évolution de séquences sur un arbre :
relai d'un message de la racine vers les feuilles
transmission avec du «bruit» sur chaque branche
bruit = mutations

modèle probabiliste pour les mutations
les séquences correspond à un «échantillon» aléatoire

Cavender-Farris

modèle de Cavender-Farris : séquences binaires, mutations symmetriques

Exemple : noeud x est le parent de y, transmission d'un bit $X \to Y$.
Probabilité de mutation sur la branche : p_{xy}.
Alors $\mathbb{P}\{X \neq Y\} = p_{xy}$.

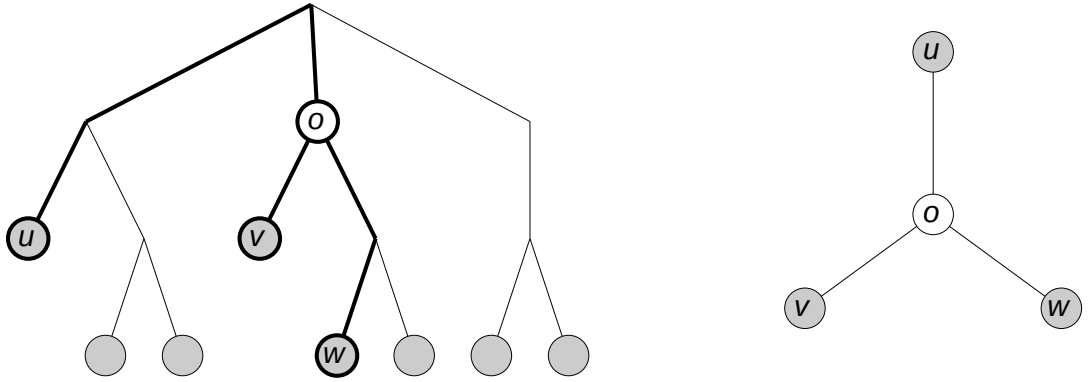
Similarité de deux noeuds : $S(x, y) = \mathbb{P}\{X = Y\} - \mathbb{P}\{X \neq Y\}$.

Thm. $S(x, y)$ se multiplie sur chaque chemin.

Donc $D(x, y) = -\log S(x, y)$ est une distance additive sur l'arbre.

Distances - 2

Algorithme pour la construction d'un arbre à partir des distances : triples



$$D(u,o) = \Big(D(u,v) + D(u,w) - D(v,w) \Big) / 2.$$

Algorithme pour construire l'arbre

[Esquissé. . .]

Feuilles : $1, \ldots, n$; matrice de distances entre feuilles de taille $n \times n$

1. Commencer avec le triple $1, 2, 3$ et son centre
2. Pour $i = 4, \ldots, n$:
2.1 choisir* un triple j_1, j_2, i avec $j_1, j_2 < i$;
2.2 insérer son centre c sur le chemin entre j_1 et j_2 dans l'arbre ;
2.3 joindre i à c et calculer les nouvelles longueurs de branches

* le centre doit être différent d'autres vertices déjà dans l'arbre

Jukes-Cantor

Jukes-Cantor-Neyman : évolution de séquences sur un alphabet $\Sigma = \{a_1, \ldots, a_m\}$

mutations de caractères indépendantes en chaque position des séquences
mutation sur une branche XY : symétrique, spécifiée par la probabilité de mutation p

$$\mathbb{P}\Big\{Y = a' \;\Big|\; X = a\Big\} = \begin{cases} 1-p & \text{si } a = a'; \\ p/(m-1) & \text{si } a \neq a'. \end{cases}$$

Distance : $D(X,Y) = -log(1 - \frac{m}{m-1}\mathbb{P}\{X \neq Y\})$

Modèle Markovien

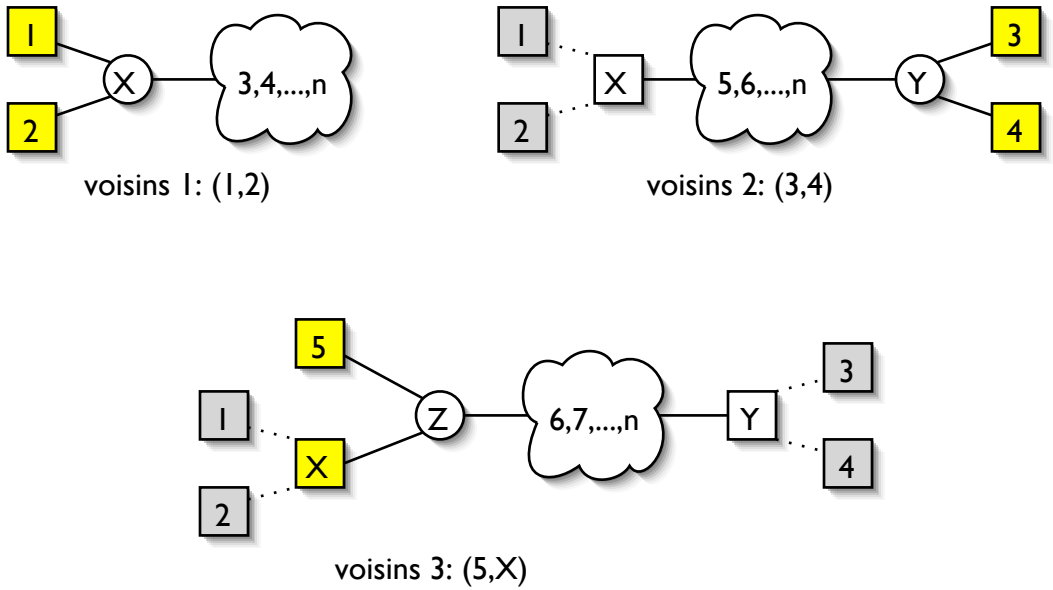
Mutations indépendantes ; probabilités de substitutions spécifiées par une matrice de transition

$$\mathbf{M}_{X \to Y}[a, a'] = \mathbb{P}\Big\{ Y = a' \;\Big|\; X = a \Big\}$$

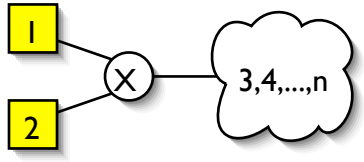
On a que $\mathbf{M}_{X \to Z} = \mathbf{M}_{X \to Y} \mathbf{M}_{Y \to Z}$.

Donc $D(X, Y) = -\log \left(\det \mathbf{M}_{X \to Y} \det \mathbf{M}_{Y \to X} \right)$ est une distance sur l'arbre (additive et symétrique) — appellée distance paralinéaire

Neighbor joining



Neighbor joining - 2



calcul de longueurs de branches : prenons la moyenne

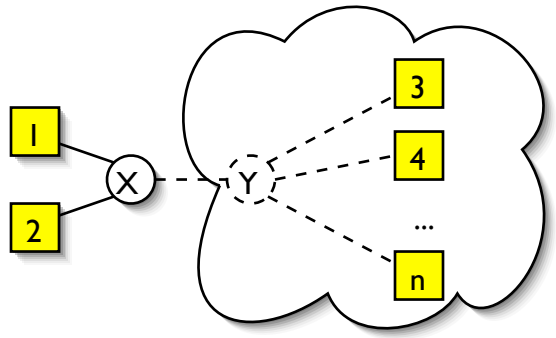
$$L_{1X} = \frac{1}{n-2} \sum_{i=3}^{n} (d_{1i} + d_{12} - d_{2i})/2.$$

L_{2X} même chose

$L_{iX} = (d_{1i} + d_{2i} - d_{12})/2.$

Neighbor joining - 3

choix des voisins : minimiser le somme total de longueurs de branches



$$S_{12} = L_{1X} + L_{2X} + L_{XY} + \sum_{i=3}^{n} L_{Yi}.$$

où $L_{XY} = (\sum_{i \geq 3} L_{Xi} - \sum_{i \geq 3} L_{Yi})/(n-2)$.

Neighbor joining - 4

On a $\sum_{i=3}^{n} L_{Yi} = \frac{1}{n-3} \sum_{3 \leq i < j \leq n} d_{ij}$. Après un peu d'arithmétique :

$$S_{12} = \frac{1}{2} d_{12} + \frac{1}{n-2} \sum_{i<j} d_{ij} - \frac{1}{n-2} \left(\frac{\sum_{i=1}^{n} d_{1i} + \sum_{i=1}^{n} d_{2i}}{2} \right)$$

Calculer en avance : $R_k = \sum_{i=1}^{n} d_{ki}$ pour chaque feuille k,
choisir la paire qui minimise $(n-2)d_{kk'} - R_k - R_{k'}$: elle minimise $S_{kk'}$ aussi

Neighbor joining - 5

algorithme en $O(n^3)$ (pas itératif) :
(1) trouver 1,2 en minimisant S_{12} (temps $O(n^2)$) ;
(2) calculer L_{1X}, L_{2X}, et d_{iX} pour $i \geq 3$ (temps $O(n)$) ;
(3) remplacer $1, 2$ par X dans la matrice de distances.

Thm. Ça donne l'arbre correct pour des distances additives.

Preuve. Idée seulement : le calcul des longueurs de branches est correct si 12 sont les voisins. Il faut prouver que 12 sont les voisins quand on minimise S_{12}.