

Analyse de séquences et phylogénie moléculaire -

Méthodes de distances

Ecole doctorale E2M2 - 2015-2016

(<http://www.frangun.org>)

Céline Brochier (celine.brochier-armanet@univ-lyon1.fr)

Guy Perrière (guy.perriere@univ-lyon1.fr)

Analyse de séquences et phylogénie moléculaire (Céline Brochier-Armanet 2015-2016)

Méthodes de distances - Principe général

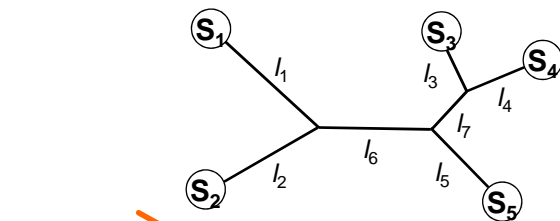
- Plusieurs méthodes (UPGMA, NJ, ME, etc.)
- Deux grandes étapes
 - Construction de la matrice de distances
 - Reconstruction d'un arbre phylogénique dont les $(\delta_{i,j})$ sont les + proches possibles des $(d_{i,j}) \Leftrightarrow$ minimise Q

$$Q = \sum_{i=1}^n \sum_{j=1}^n (d_{i,j} - \delta_{i,j})$$

Alignement de séquences homologues

Modèle d'évo.

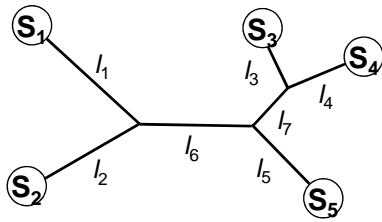
	S_1	S_2	S_3	S_4	S_5
S_1	—	$d_{1,2}$	$d_{1,3}$	$d_{1,4}$	$d_{1,5}$
S_2	$d_{2,1}$	—	$d_{2,3}$	$d_{2,4}$	$d_{2,5}$
S_3	$d_{3,1}$	$d_{3,2}$	—	$d_{3,4}$	$d_{3,5}$
S_4	$d_{4,1}$	$d_{4,2}$	$d_{4,3}$	—	$d_{4,5}$
S_5	$d_{5,1}$	$d_{5,2}$	$d_{5,3}$	$d_{5,4}$	—



Méthode de dist.

Méthodes de distances - Principe général

A chaque arbre on peut associer une distance δ entre séquences :



$$\delta(S_1, S_4) = l_1 + l_7 + l_6 + l_4$$

$d(S_1, S_4)$ = distance estimée entre les séquences S_1 et S_4 avec un modèle d'évolution

Il est possible de calculer les valeurs des longueurs des branches qui optimisent la ressemblance entre les δ et les distances évolutives d :

=> minimiser
$$Q = \sum_{1 \leq i < j \leq n} (d_{i,j} - \delta_{i,j})^2$$
 Solution générale de ce problème: Rzhetsky & Nei (1993) MBE 10:1073

Il est alors possible de calculer la longueur totale de l'arbre :
S = somme des longueurs de toutes les branches

forme d'arbre ==> «meilleures» longueurs des branches ==> longueur totale de l'arbre

UPGMA (*Unweighted pair-group method with arithmetic means*)

- Algorithme itératif de clustering: *création à chaque étape d'un nouveau cluster regroupant deux clusters proches* (Sokal et Michener, 1958)
- ⇒ L'arbre est construit "de bas en haut" : on part des feuilles et à chaque étape on rajoute un nœud au-dessus des précédents
- Complexité $O(n^3)$
- Condition d'application
 - **Hypothèse d'horloge moléculaire** ⇔ constance des taux d'évolution le long des lignées
- Caractéristiques des arbres obtenus
 - Ils sont racinés
 - Les longueurs des branches allant de la racine à n'importe quelle feuille sont égales (arbres ultramétriques)

UPGMA - Algorithme

1. Trouver les deux UTO i et j pour lesquels la distance d_{ij} est la plus petite
2. Créer le nouveau groupe (ij) contenant n_{ij} membres avec $n_{ij} = n_i + n_j$ (n_i et $n_j \Leftrightarrow$ nombre d'UTO dans les groupes i et j)
3. Connecter i et j dans l'arbre à un nouveau nœud (ij) qui correspond au nouveau groupe (ij) . Attribuer aux branches L_i et L_j connectant i à (ij) et j à (ij) la longueur $d_{ij}/2$
4. Calculer la distance entre le nouveau groupe (ij) et tous les autres groupes en utilisant $d_{ij,k} = n_i \times d_{ik} / (n_i + n_j) + n_j \times d_{jk} / (n_i + n_j)$
5. Eliminer les colonnes et les lignes correspondant aux groupes i et j et ajouter celles correspondant au nouveau groupe (ij)
6. Si il reste un seul élément dans la matrice, arrêter, sinon retourner en 1.

UPGMA - Application

Matrice de distances extraite de Sarich 1969



	Chien	Ours	Racoon	Belette	Phoque	Otarie	Chat	Singe
Chien	0	32	48	51	50	48	98	148
Ours	32	0	26	34	29	33	84	136
Racoon	48	26	0	42	44	44	92	152
Belette	51	34	42	0	44	38	86	142
Phoque	50	29	44	44	0	24	89	142
Otarie	48	33	44	38	24	0	90	142
Chat	98	84	92	86	89	90	0	148
Singe	148	136	152	142	142	142	148	0

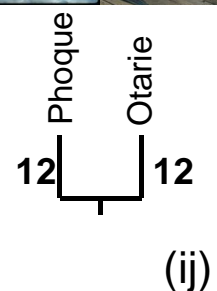
1. **Trouver les deux UTO i et j pour lesquels la distance d_{ij} est la plus petite**
2. Créer le nouveau groupe (ij) contenant n_{ij} membres avec $n_{ij} = n_i + n_j$ (n_i et $n_j \leftrightarrow$ nombre d'UTO dans les groupes i et j)
3. Connecter i et j dans l'arbre à un nouveau nœud (ij) qui correspond au nouveau groupe (ij). Attribuer aux branches L_i et L_j connectant i à (ij) et j à (ij) la longueur $d_{ij} / 2$
4. Calculer la distance entre le nouveau groupe (ij) et tous les autres groupes en utilisant $d_{ij,k} = n_i \times d_{ik} / (n_i + n_j) + n_j \times d_{jk} / (n_i + n_j)$
5. Éliminer les colonnes et les lignes correspondant aux groupes i et j et ajouter celles correspondant au nouveau groupe (ij)
6. Si il reste un seul élément dans la matrice, arrêter, sinon retourner en 1.



	Chien	Ours	Racoon	Belette	Phoque	Otarie	Chat	Singe
Chien	0	32	48	51	50	48	98	148
Ours	32	0	26	34	29	33	84	136
Racoon	48	26	0	42	44	44	92	152
Belette	51	34	42	0	44	38	86	142
Phoque	50	29	44	44	0	24	89	142
Otarie	48	33	44	38	24	0	90	142
Chat	98	84	92	86	89	90	0	148
Singe	148	136	152	142	142	142	148	0

1. Trouver les deux UTO i et j pour lesquels la distance d_{ij} est la plus petite
2. Créer le nouveau groupe (ij) contenant n_{ij} membres avec $n_{ij} = n_i + n_j$ (n_i et $n_j \Leftrightarrow$ nombre d'UTO dans les groupes i et j) $n_{ij} = n_i + n_j \Leftrightarrow n_{ij} = 1 + 1 = 2$
3. Connecter i et j dans l'arbre à un nouveau nœud (ij) qui correspond au nouveau groupe (ij) . Attribuer aux branches L_i et L_j connectant i à (ij) et j à (ij) la longueur $d_{ij}/2$
4. Calculer la distance entre le nouveau groupe (ij) et tous les autres groupes en utilisant $d_{ij,k} = n_i \times d_{ik} / (n_i + n_j) + n_j \times d_{jk} / (n_i + n_j)$
5. Éliminer les colonnes et les lignes correspondant aux groupes i et j et ajouter celles correspondant au nouveau groupe (ij)
6. Si il reste un seul élément dans la matrice, arrêter, sinon retourner en 1.

1. Trouver les deux UTO i et j pour lesquels la distance d_{ij} est la plus petite
2. Créer le nouveau groupe (ij) contenant n_{ij} membres avec $n_{ij} = n_i + n_j$ (n_i et $n_j \Leftrightarrow$ nombre d'UTO dans les groupes i et j)
3. Connecter i et j dans l'arbre à un nouveau nœud (ij) qui correspond au nouveau groupe (ij) . Attribuer aux branches L_i et L_j connectant i à (ij) et j à (ij) la longueur $d_{ij}/2$ soit $L_i = 24/2 = 12$ et $L_j = 24/2 = 12$
4. Calculer la distance entre le nouveau groupe (ij) et tous les autres groupes en utilisant $d_{ij,k} = n_i \times d_{ik} / (n_i + n_j) + n_j \times d_{jk} / (n_i + n_j)$
5. Éliminer les colonnes et les lignes correspondant aux groupes i et j et ajouter celles correspondant au nouveau groupe (ij)
6. Si il reste un seul élément dans la matrice, arrêter, sinon retourner en 1.



1. Trouver les deux UTO i et j pour lesquels la distance d_{ij} est la plus petite
2. Créer le nouveau groupe (ij) contenant n_{ij} membres avec $n_{ij} = n_i + n_j$ (n_i et $n_j \Leftrightarrow$ nombre d'UTO dans les groupes i et j)
3. Connecter i et j dans l'arbre à un nouveau nœud (ij) qui correspond au nouveau groupe (ij). Attribuer aux branches L_i et L_j connectant i à (ij) et j à (ij) la longueur $d_{ij} / 2$
4. Calculer la distance entre le nouveau groupe (ij) et tous les autres groupes en utilisant $d_{ij,k} = n_i \times d_{ik} / (n_i + n_j) + n_j \times d_{jk} / (n_i + n_j) = n_{\text{phoque}} \times d_{\text{phoque,chien}} / (n_{\text{phoque}} + n_{\text{otarie}}) + n_{\text{otarie}} \times d_{\text{otarie,chien}} / (n_{\text{phoque}} + n_{\text{otarie}}) = (50 + 48) / 2 = 49$
5. Éliminer les colonnes et les lignes correspondant aux groupes i et j et ajouter celles correspondant au nouveau groupe (ij)
6. Si il reste un seul élément dans la matrice, arrêter, sinon retourner en 1.



	Chien	Ours	Racoon	Belette	Phoque Otarie	Phoque	Otarie	Chat	Singe
Chien	0	32	48	51		50	48	98	148
Ours	32	0	26	34		29	33	84	136
Racoon	48	26	0	42		44	44	92	152
Belette	51	34	42	0		44	38	86	142
Phoque Otarie					0				
Phoque	50	29	44	44		0	24	89	142
Otarie	48	33	44	38		24	0	90	142
Chat	98	84	92	86		89	90	0	148
Singe	148	136	152	142		142	142	148	0



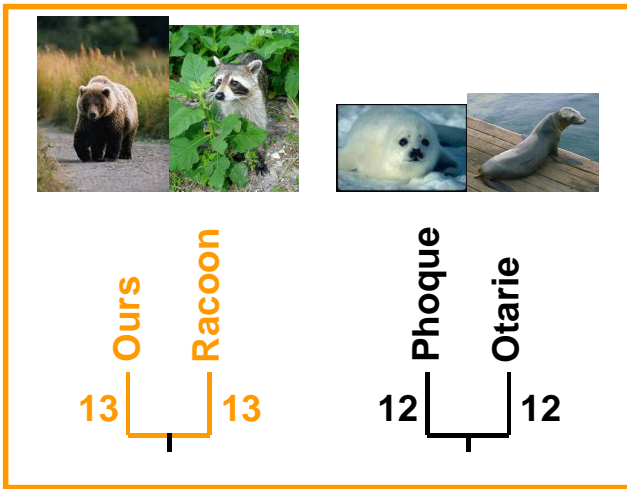
	Chien	Ours	Racoon	Belette	Phoque Otarie	Phoque	Otarie	Chat	Singe
Chien	0	32	48	51	$(50+48)/2 = 49$	50	48	98	148
Ours	32	0	26	34	$(29+33)/2 = 31$	29	33	84	136
Racoon	48	26	0	42	$(44+44)/2 = 44$	44	44	92	152
Belette	51	34	42	0	$(44+38)/2 = 41$	44	38	86	142
Phoque Otarie	$(50+48)/2 = 49$	$(29+33)/2 = 31$	$(44+44)/2 = 44$	$(44+38)/2 = 41$	0			$(89+90)/2 = 89.5$	$(142+142)/2 = 142$
Phoque	50	29	44	44		0	24	89	142
Otarie	48	33	44	38		24	0	90	142
Chat	98	84	92	86	$(89+90)/2 = 89.5$	89	90	0	148
Singe	148	136	152	142	$(142+142)/2 = 142$	142	142	148	0

1. Trouver les deux UTO i et j pour lesquels la distance d_{ij} est la plus petite
2. Créer le nouveau groupe (ij) contenant n_{ij} membres avec $n_{ij} = n_i + n_j$ (n_i et $n_j \Leftrightarrow$ nombre d'UTO dans les groupes i et j)
3. Connecter i et j dans l'arbre à un nouveau nœud (ij) qui correspond au nouveau groupe (ij) . Attribuer aux branches L_i et L_j connectant i à (ij) et j à (ij) la longueur $d_{ij} / 2$
4. Calculer la distance entre le nouveau groupe (ij) et tous les autres groupes en utilisant $d_{ij,k} = n_i \times d_{ik} / (n_i + n_j) + n_j \times d_{jk} / (n_i + n_j)$
5. **Éliminer les colonnes et les lignes correspondant aux groupes i et j et ajouter celles correspondant au nouveau groupe (ij)**
6. Si il reste un seul élément dans la matrice, arrêter, sinon retourner en 1.

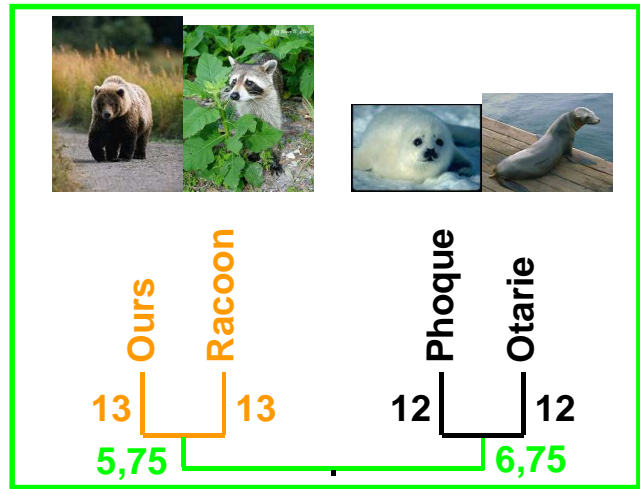


	Chien	Ours	Racoon	Belette	Phoque Otarie	Chat	Singe
Chien	0	32	48	51	49	98	148
Ours	32	0	26	34	31	84	136
Racoon	48	26	0	42	44	92	152
Belette	51	34	42	0	41	86	142
Phoque Otarie	49	31	44	41	0	89.5	142
Chat	98	84	92	86	89.5	0	148
Singe	148	136	152	142	142	148	0

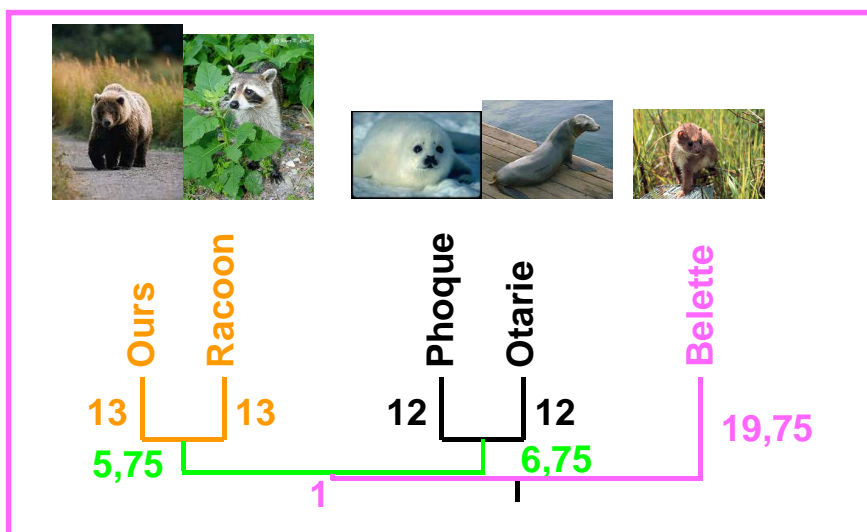
1. Trouver les deux UTO i et j pour lesquels la distance d_{ij} est la plus petite
2. Créer le nouveau groupe (ij) contenant n_{ij} membres avec $n_{ij} = n_i + n_j$ (n_i et $n_j \Leftrightarrow$ nombre d'UTO dans les groupes i et j)
3. Connecter i et j dans l'arbre à un nouveau nœud (ij) qui correspond au nouveau groupe (ij) . Attribuer aux branches L_i et L_j connectant i à (ij) et j à (ij) la longueur $d_{ij} / 2$
4. Calculer la distance entre le nouveau groupe (ij) et tous les autres groupes en utilisant $d_{ij,k} = n_i \times d_{ik} / (n_i + n_j) + n_j \times d_{jk} / (n_i + n_j)$
5. Éliminer les colonnes et les lignes correspondant aux groupes i et j et ajouter celles correspondant au nouveau groupe (ij)
6. Si il reste un seul élément dans la matrice, arrêter, sinon retourner en 1.



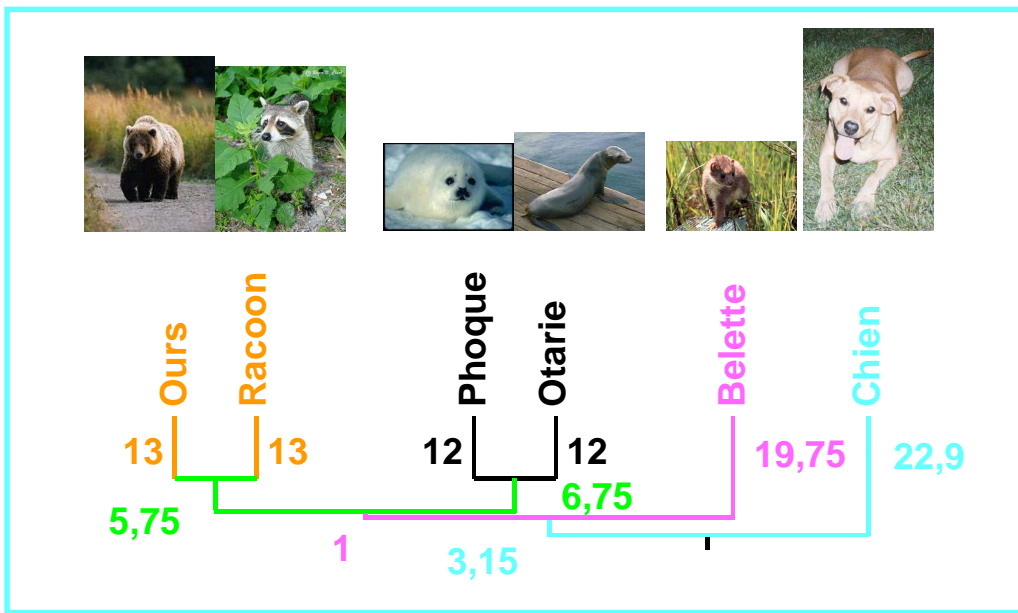
Deuxième agglomération



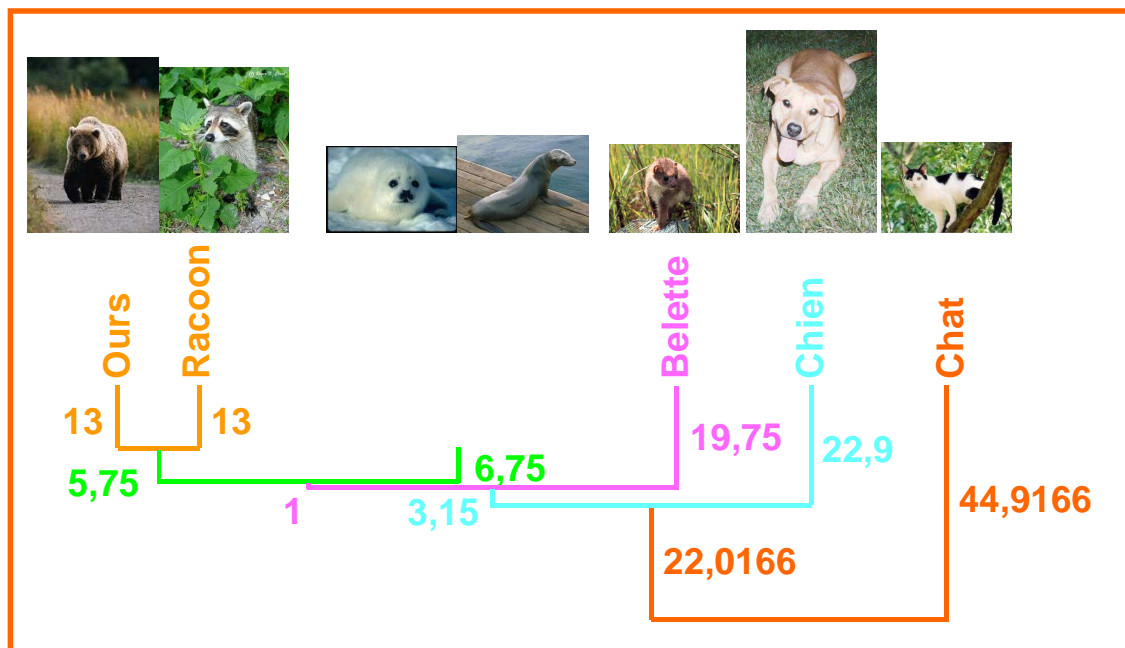
Troisième agglomération



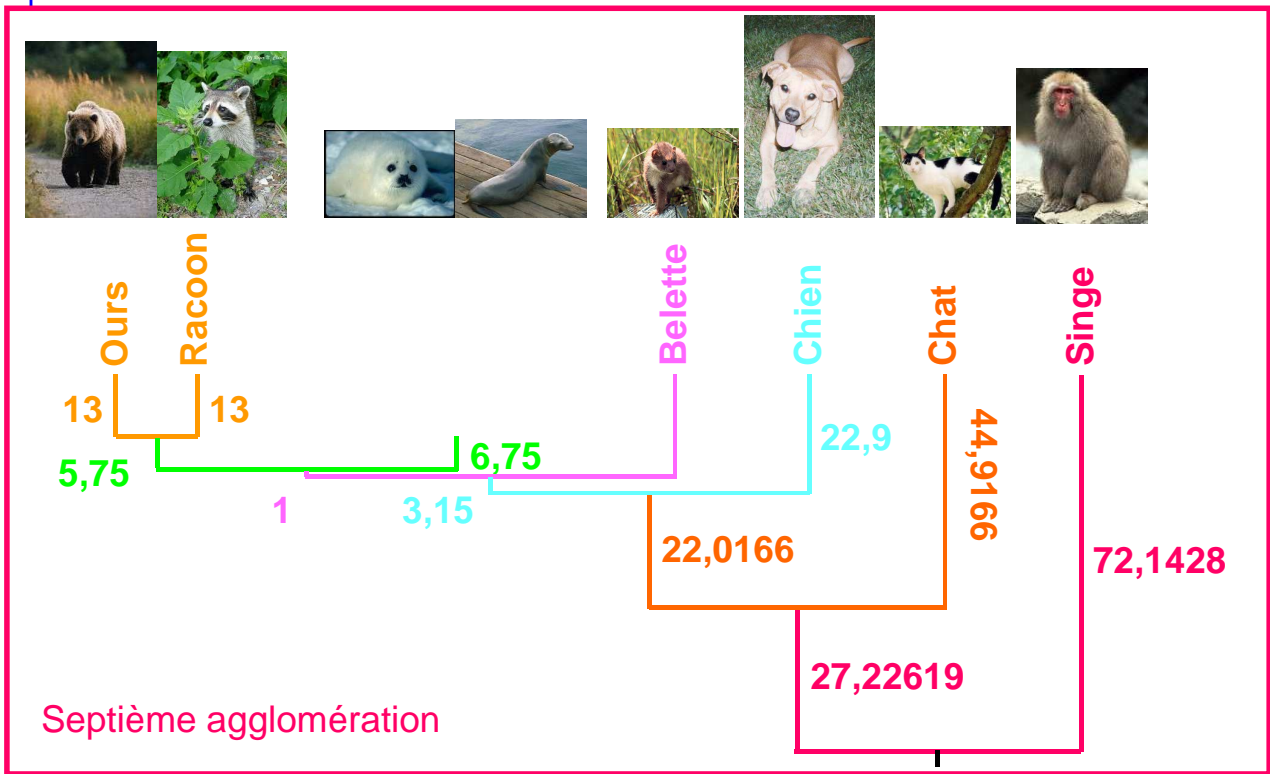
Quatrième agglomération



Cinquième agglomération

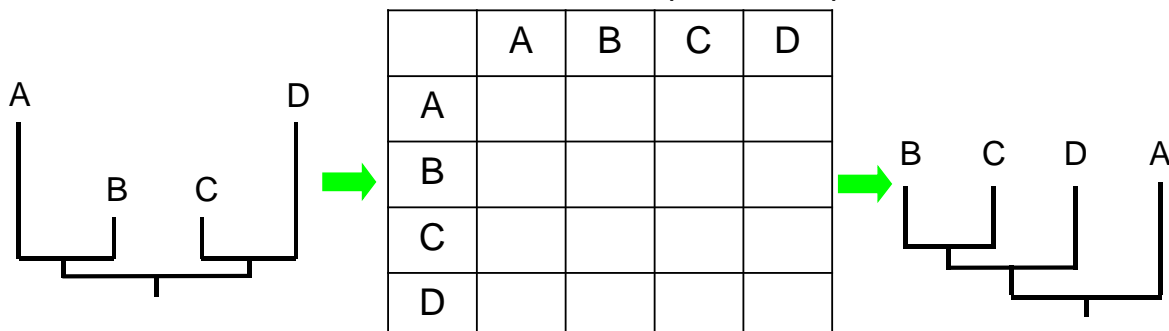


Sixième agglomération



Conclusions sur l'UPGMA

- Avantages de l'algorithme:
 - Rapidité & simplicité
- Critiques
 - Hypothèse de l'égalité des taux d'évolution entre les lignées.
 - ⇒ Résultats faux si les distances de la matrice n'obéissent pas au critère d'horloge moléculaire
 - ⇒ N'est presque plus utilisée
- Peut être réaliste si on étudie des espèces très proches



Méthodes des moindres carrés

- Principe général:
 - Chercher la topologie qui minimise les carrés des écarts entre les distances $d_{i,j}$ de la matrice et celles de l'arbre reconstruit

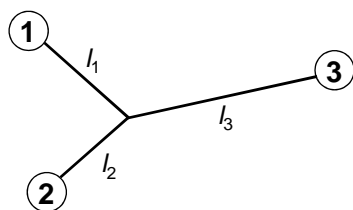
$$Q = \sum_{i < j} w_{i,j} (d_{i,j} - \delta_{i,j})^2$$

Avec $w_{i,j}$ une pondération associée à chaque couple i,j

- Implique l'exploration de toutes les topologies
 - Complexité de l'algorithme: $O(n^5)$
- Caractéristiques des arbres obtenus
 - Ils sont non racinés

Méthodes des moindres carrés – Méthode de Fitch et Margoliash – nb taxa = 3

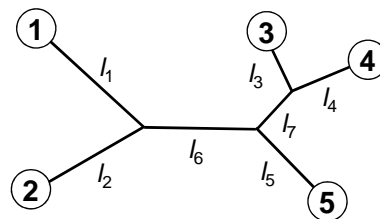
- Méthode permettant de calculer les longueurs de branches $\delta_{i,j}$ associées à une topologie
- Bonne estimation des valeurs obtenues par le calcul des moindres carrés standards
- Principe général: se ramener à chaque étape à un arbre à trois groupes



$$\begin{array}{l}
 d_{1,2} = l_1 + l_2 \\
 d_{1,3} = l_1 + l_3 \\
 d_{2,3} = l_2 + l_3
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{l}
 l_1 = (d_{1,2} + d_{1,3} - d_{2,3}) / 2 \\
 l_2 = (d_{1,2} + d_{2,3} - d_{1,3}) / 2 \\
 l_3 = (d_{1,3} + d_{2,3} - d_{1,2}) / 2
 \end{array}$$

Méthodes des moindres carrés – Méthode de Fitch et Margoliash – nb taxa > 3

- Identifier les deux taxa A et B pour lesquels la valeur $d_{i,j}$ est minimale
- Regrouper tous les autres taxa dans un ensemble noté C
 - Calculer la distance entre A et C comme étant la moyenne des distances séparant A et chaque séquence contenue dans C (Idem pour B et C)
- Calculer les longueurs de branches en suivant la procédure du cas à 3 taxa
- Regrouper A et B dans un ensemble {A,B}
- Calculer d'une nouvelle matrice de distance: calculer pour les taxa restant la moyenne des distances les séparant des éléments du groupe {A,B}
- Si il reste 3 éléments ou plus dans la matrice répéter l'algorithme
- Calcul de Q



Répéter cette procédure en initiant l'algorithme avec toutes les paires de taxa possibles

Méthode du minimum d'évolution

- Principe général:
 - Calculer pour chaque topologie possible les longueurs de branches en utilisant la méthode des moindres carrés
 - Retenir la topologie présentant la plus petite longueur totale S

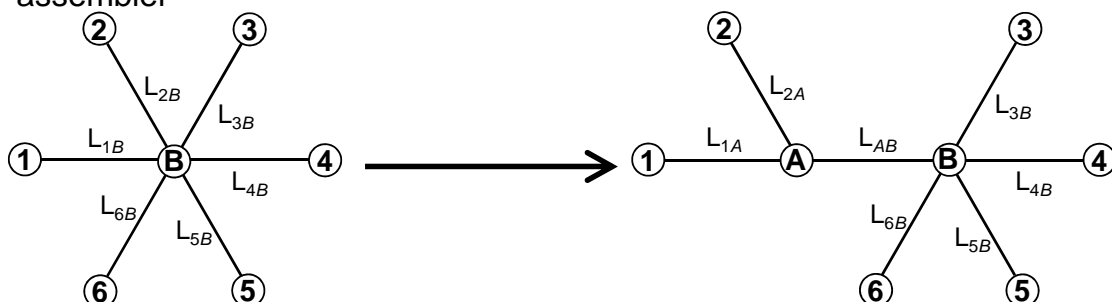
$$S = \sum_{k=1}^{2n-3} l_k$$

Le Neighbor-Joining (NJ)

- Développé par Saitou et Nei (1987) est une approximation de l'algorithme pour trouver l'arbre le plus court (minimum évolution)
- Avantages
 - Rapidité => permet de travailler avec un très grand nombre de taxons (plusieurs centaines)
 - Bonne approximation de la méthode du minimum d'évolution
 - Retrouve l'arbre vrai si la matrice de distances est un reflet exact d'un arbre
- Conditions d'application
 - Les taux d'évolution ne sont pas les mêmes dans toutes les lignées
- Caractéristiques des arbres obtenus
 - Ils sont non racinés
- Principe:
 - A chaque étape, rechercher le couple d'UTO qui minimise la longueur totale de l'arbre

Le neighbor-joining – Principe

- Point de départ = topologie en étoile
 - Étape 1 : Pour toutes les paires i,j possibles, calculer $S_{i,j}$ la longueur de l'arbre obtenu
 - Étape 2 : Retenir la paire i,j générant la plus petite valeur $S_{i,j}$; grouper i et j dans l'arbre
 - Étape 3 : Calculer les nouvelles distances d entre le groupe nouvellement formé et les séquences restantes
 - Étape 4 : Retourner à l'étape 1 si il reste plus de 4 séquences/groupes à assembler



Le neighbor-joining – Algorithme simplifié (Studier et Keppler 1988)

- (1) Pour chaque feuille i calculer $u_i = R_i / (m-2) = \sum_{k=1}^m d_{ik} / (m-2)$
 - (2) Choisir i et j pour lesquels $d_{ij} - u_i - u_j$ est la plus petite
 - (3) Joindre i et j . Calculer L_i la longueur de la branche reliant i au nouveau nœud A et L_j la longueur de la branche reliant j au nouveau nœud A comme étant
 - $L_i = d_{ij}/2 + (u_i - u_j)/2$
 - $L_j = d_{ij}/2 + (u_j - u_i)/2$
 - (4) Calculer la distance entre le nouveau nœud A et chaque autre feuille comme
 - $d(ij),k = (d_{ik} + d_{jk} - d_{ij}) / 2$
 - (5) Éliminer les colonnes et les lignes correspondant aux groupes i et j et ajouter celles correspondant au nouveau groupe (ij)
 - (6) Si il reste un seul élément dans la matrice connecter les deux derniers nœuds, sinon retourner en 1
-

Le neighbor-joining – Application

- (1) Pour chaque feuille i calculer $u_i = R_i / (m-2) = \sum_{k=1}^m d_{ik} / (m-2)$
Choisir i et j pour lesquels $d_{ij} - u_i - u_j$ est la plus petite
 - (3) Joindre i et j . Calculer L_i la longueur de la branche reliant i au nouveau nœud A et L_j la longueur de la branche reliant j au nouveau nœud A comme étant
 - $L_i = d_{ij}/2 + (u_i - u_j)/2$
 - $L_j = d_{ij}/2 + (u_j - u_i)/2$
 - (4) Calculer la distance entre le nouveau nœud A et chaque autre feuille comme
 - $d(ij),k = (d_{ik} + d_{jk} - d_{ij}) / 2$
 - (5) Éliminer les colonnes et les lignes correspondant aux groupes i et j et ajouter celles correspondant au nouveau groupe (ij)
 - (6) Si il reste un seul élément dans la matrice connecter les deux derniers nœuds, sinon retourner en 1
-



	Chien	Ours	Racoon	Belette	Phoque	Otarie	Chat	Singe
Chien	0							
Ours	32	0						
Racoon	48	26	0					
Belette	51	34	42	0				
Phoque	50	29	44	44	0			
Otarie	48	33	44	38	24	0		
Chat	98	84	92	86	89	90	0	
Singe	148	136	152	142	142	142	148	0
u_i	79,167	62,333	74,667	72,833	70,333	69,833	114,5	168,333

$$u_i = \sum_{k=1:k \neq i}^m d_{ik} / (m-2)$$

(1) Pour chaque feuille i calculer $u_i = R_i / (m-2) = \sum_{k=1}^m d_{ik} / (m-2)$

(2) Choisir i et j pour lesquels $d_{ij} - u_i - u_j$ est la plus petite

Joindre i et j . Calculer L_i la longueur de la branche reliant i au nouveau nœud A et L_j la longueur de la branche reliant j au nouveau nœud A comme étant

- $L_i = d_{ij}/2 + (u_i - u_j)/2$
- $L_j = d_{ij}/2 + (u_j - u_i)/2$

(4) Calculer la distance entre le nouveau nœud A et chaque autre feuille comme

- $d(ij),k = (d_{ik} + d_{jk} - d_{ij}) / 2$

(5) Éliminer les colonnes et les lignes correspondant aux groupes i et j et ajouter celles correspondant au nouveau groupe (ij)

(6) Si il reste un seul élément dans la matrice connecter les deux derniers nœuds, sinon retourner en 1



	Chien	Ours	Racoon	Belette	Phoque	Otarie	Chat	Singe
Chien	0	-109,5	-105,834	-101	-99,5	-101	-95.667	-99,5
Ours	32	0	-111	-101,166	-103,666	-99,166	-92,833	-94,666
Racoon	48	26	0	-105,5	-101	-100,5	-97,167	-91
Belette	51	34	42	0	-99,166	-104,666	-101,333	-99,166
Phoque	50	29	44	44	0	-116,166	-95.833	-96,666
Otarie	48	33	44	38	24	0	-94,333	-96,166
Chat	98	84	92	86	89	90	0	-134,833
Singe	148	136	152	142	142	142	148	0
u_i	79,167	62,333	74,667	72,833	70,333	69,833	114,5	168,333

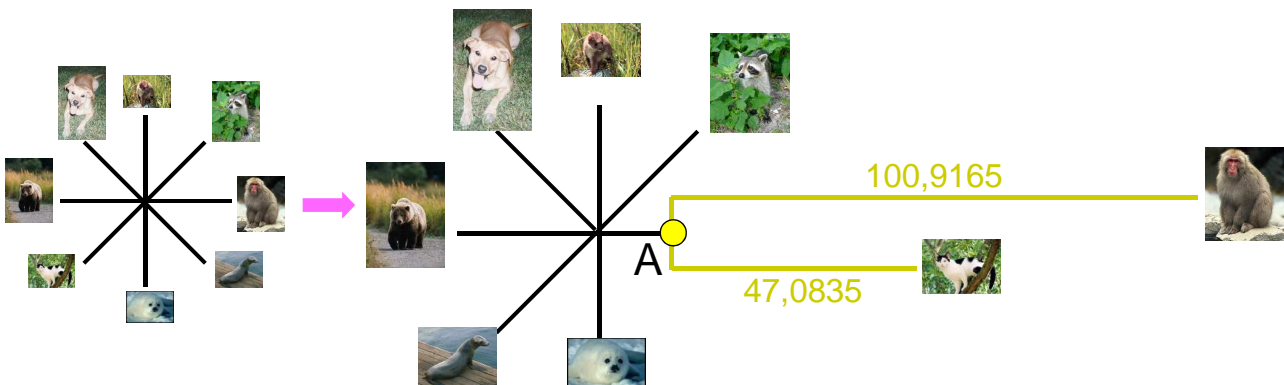
$D_{ij} - u_i - u_j$ exemple Ours/Chien : $32 - 79,167 - 62,333 = -109,5$

- (1) Pour chaque feuille i calculer $u_i = R_i / (m-2) = \sum_{k=1}^n d_{ik} / \sum_{k=1}^m d_{ik} / (m-2)$
- (2) Choisir i et j pour lesquels $d_{ij} - u_i - u_j$ est la plus petite
- (3) Joindre i et j . Calculer L_i la longueur de la branche reliant i au nouveau nœud A et L_j la longueur de la branche reliant j au nouveau nœud A comme étant

- $L_i = d_{ij}/2 + (u_i - u_j)/2 = 148/2 + (114,5 - 168,333)/2 = 47,0835$
- $L_j = d_{ij}/2 + (u_j - u_i)/2 = 148/2 + (168,333 - 114,5)/2 = 100,9165$

Calculer la distance entre le nouveau nœud A et chaque autre feuille comme

- $d(ij),k = (d_{ik} + d_{jk} - d_{ij}) / 2$



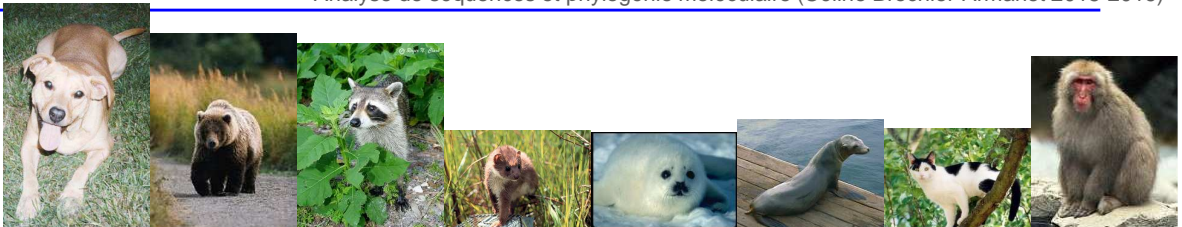
- (1) Pour chaque feuille i calculer $u_i = R_i / (m-2) = \sum_{k=1}^n d_{ik} / \sum_{k=1}^m d_{ik} / (m-2)$
- (2) Choisir i et j pour lesquels $d_{ij} - u_i - u_j$ est la plus petite
- (3) Joindre i et j . Calculer L_i la longueur de la branche reliant i au nouveau nœud A et L_j la longueur de la branche reliant j au nouveau nœud A comme étant
 - $L_i = d_{ij}/2 + (u_i - u_j)/2$
 - $L_j = d_{ij}/2 + (u_j - u_i)/2$
- (4) Calculer la distance entre le nouveau nœud A et chaque autre feuille comme
 - $d_{(ij),k} = (d_{ik} + d_{jk} - d_{ij}) / 2$
 Éliminer les colonnes et les lignes correspondant aux groupes i et j et ajouter celles correspondant au nouveau groupe (ij)
- (6) Si il reste un seul élément dans la matrice connecter les deux derniers nœuds, sinon retourner en 1



	Chien	Ours	Racoon	Belette	Phoque	Otarie	Chat	Singe
Chien	0							
Ours	32	0						
Racoon	48	26	0					
Belette	51	34	42	0				
Phoque	50	29	44	44	0			
Otarie	48	33	44	38	24	0		
Chat Singe	(98 148- 148	(84 136- 148	(92 152- 148	(86 142- 148	(89 142- 148	(90 142- 148		
Chat	98	84	92	86	89	90	0	
Singe	148	136	152	142	142	142	148	0

$$D_{(ij),k} = (D_{ik} + D_{jk} - D_{ij}) / 2$$

- (1) Pour chaque feuille i calculer $u_i = R_i / (m-2) = \sum_{k=1}^m d_{ik}$
 - (2) Choisir i et j pour lesquels $d_{ij} - u_i - u_j$ est la plus petite
 - (3) Joindre i et j . Calculer L_i la longueur de la branche reliant i au nouveau nœud A et L_j la longueur de la branche reliant j au nouveau nœud A comme étant
 - $L_i = d_{ij}/2 + (u_i - u_j)/2$
 - $L_j = d_{ij}/2 + (u_j - u_i)/2$
 - (4) Calculer la distance entre le nouveau nœud A et chaque autre feuille comme
 - $d(ij),k = (d_{ik} + d_{jk} - d_{ij}) / 2$
 - (5) Éliminer les colonnes et les lignes correspondant aux groupes i et j et ajouter celles correspondant au nouveau groupe (ij)
- Si il reste un seul élément dans la matrice connecter les deux derniers nœuds, sinon retourner en 1



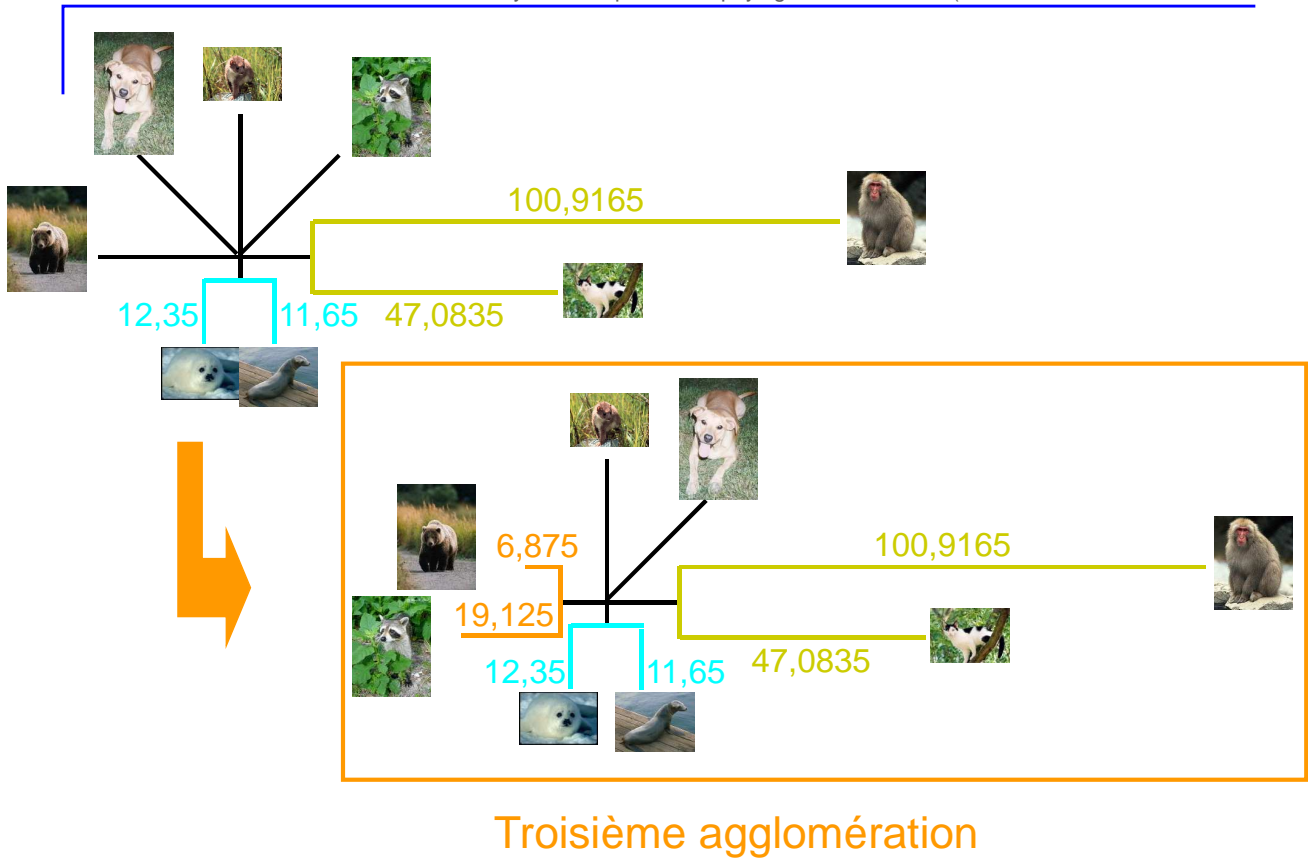
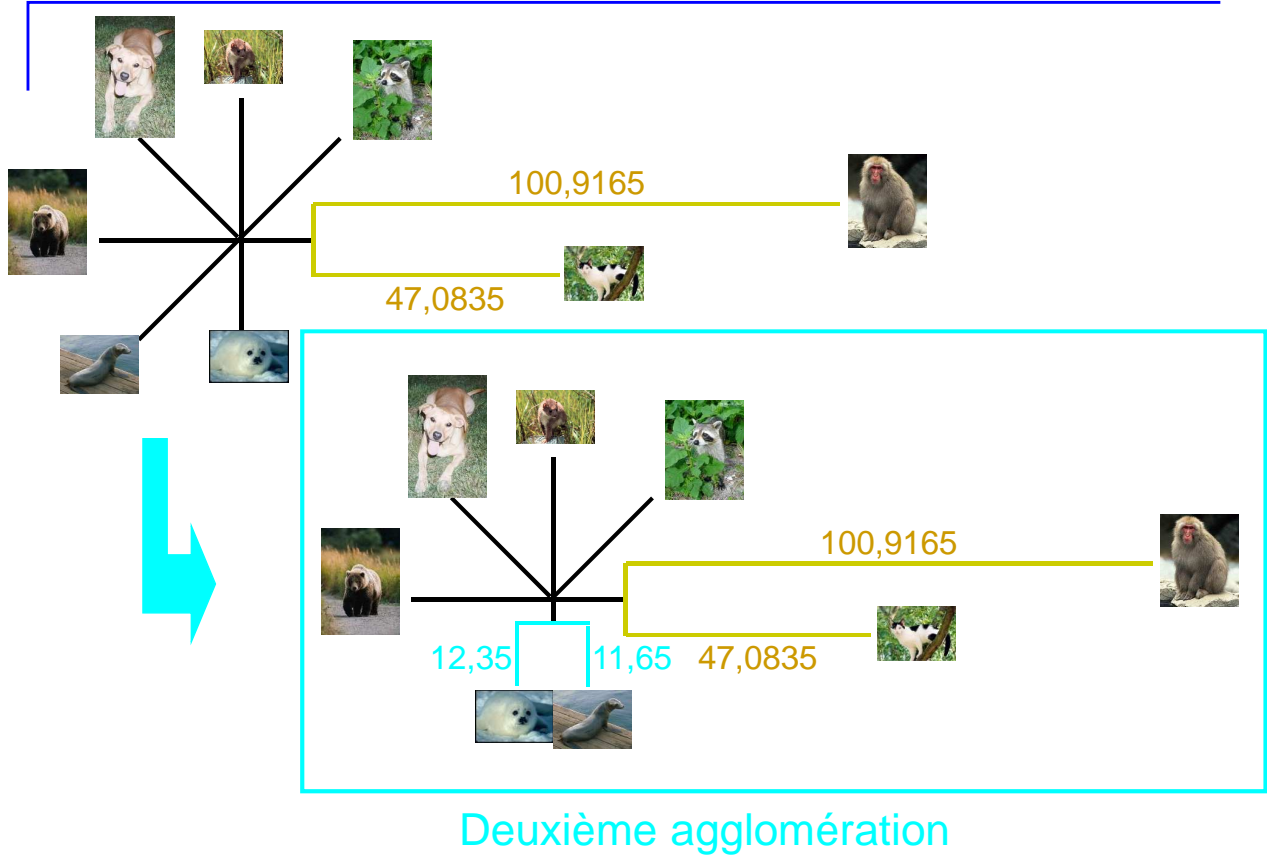
	Chien	Ours	Racoon	Belette	Phoque	Otarie	Chat	Singe
Chien	0							
Ours	32	0						
Racoon	48	26	0					
Belette	51	34	42	0				
Phoque	50	29	44	44	0			
Otarie	48	33	44	38	24	0		
Chat Singe	49	36	48	40	41,5	42		
Chat	98	84	92	86	89	90	0	
Singe	148	136	152	142	142	142	148	0

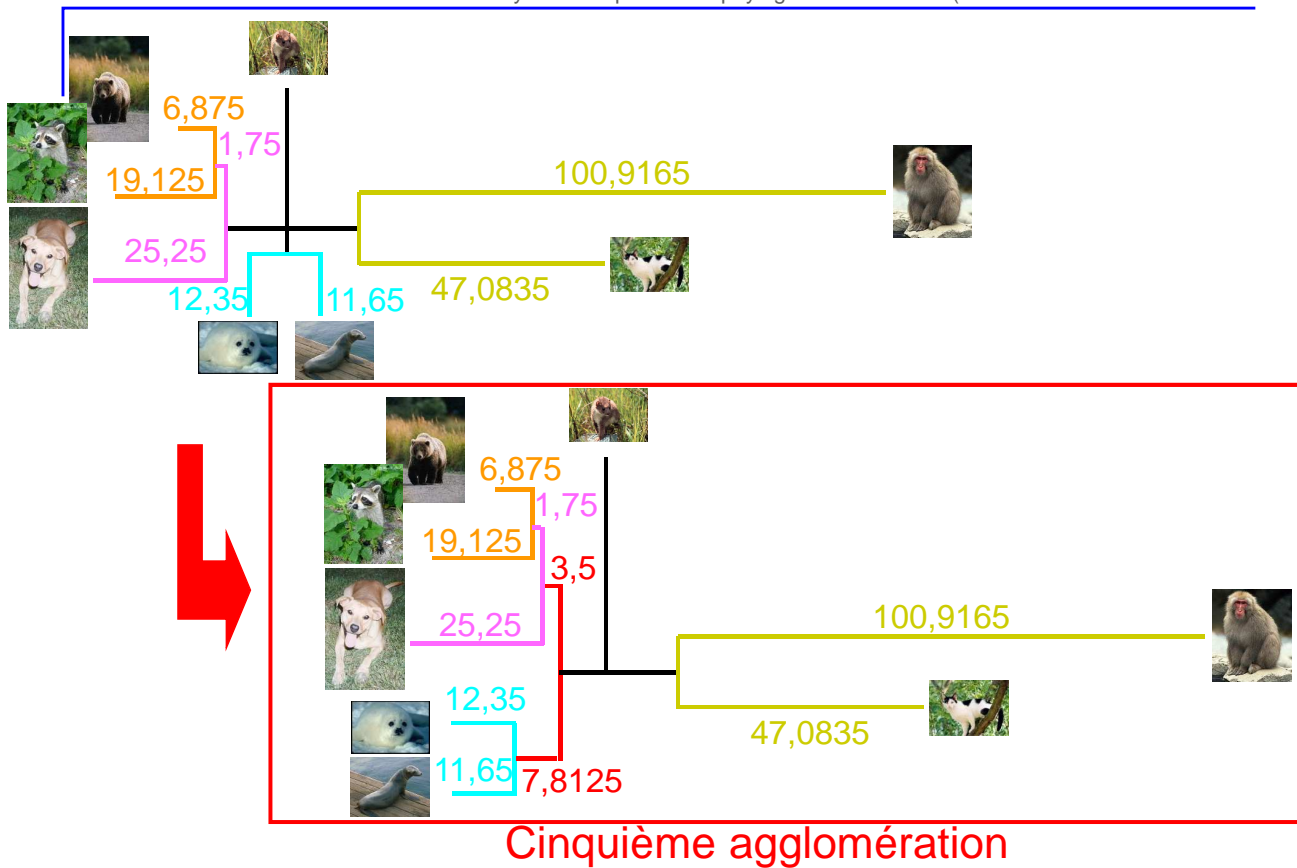
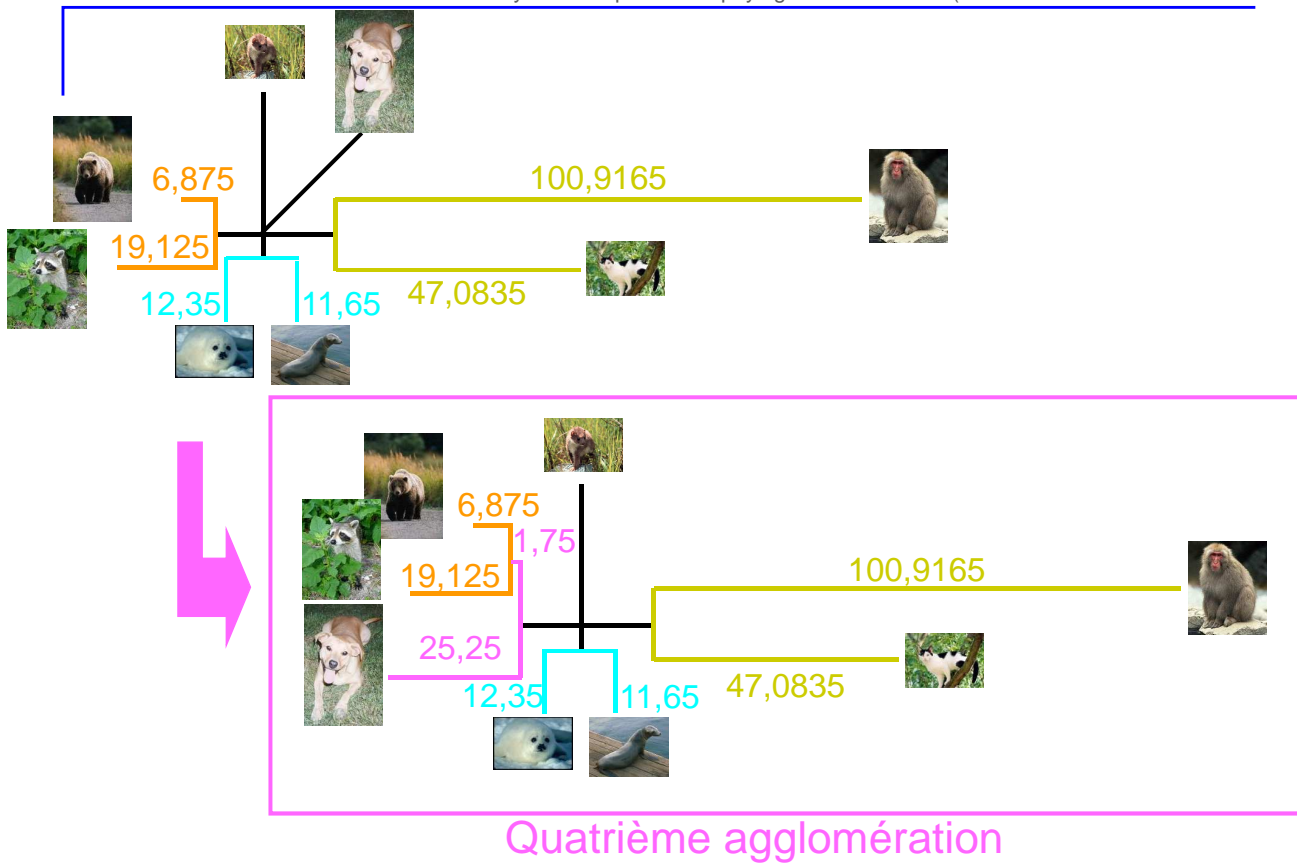
Supprimer les colonnes du chat et du singe

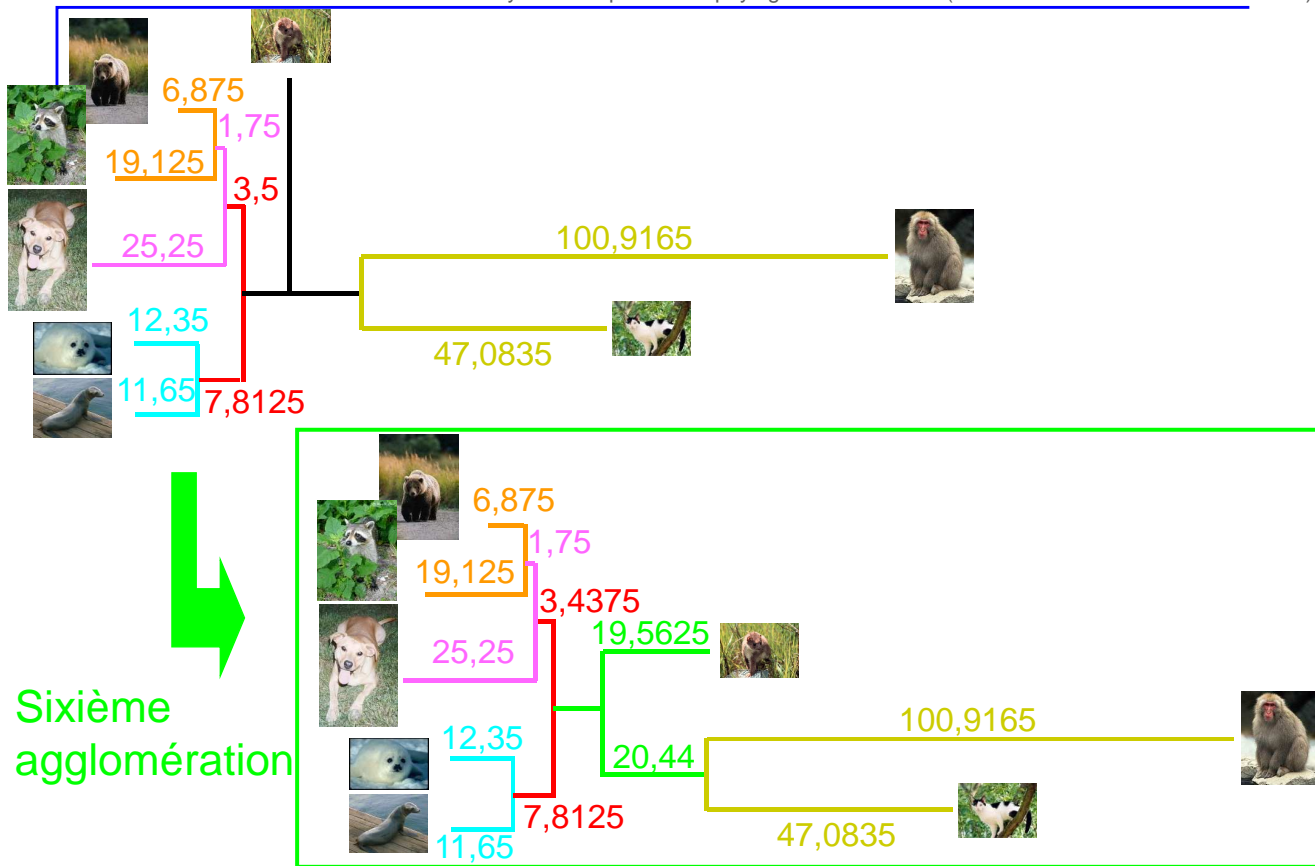


	Chien	Ours	Racoon	Belette	Phoque	Otarie	Chat Singe
Chien	0						
Ours	32	0					
Racoon	48	26	0				
Belette	51	34	42	0			
Phoque	50	29	44	44	0		
Otarie	48	33	44	38	24	0	
Chat Singe	49	36	48	40	41,5	42	0

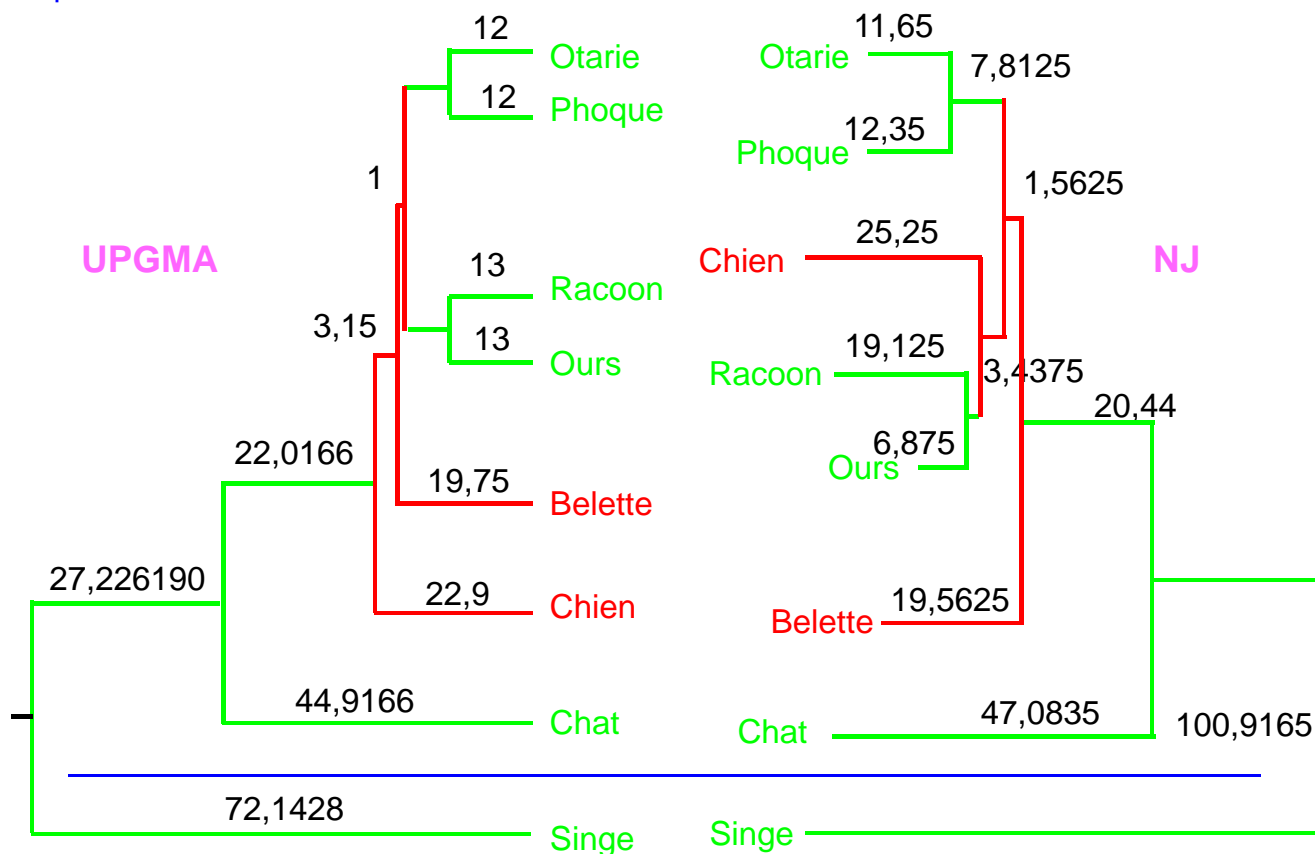
- (1) Pour chaque feuille i calculer $u_i = R_i / (m-2) = \sum_{k=1}^m d_{ik}$
- (2) Choisir i et j pour lesquels $d_{ij} - u_i - u_j$ est la plus petite
- (3) Joindre i et j . Calculer L_i la longueur de la branche reliant i au nouveau nœud A et L_j la longueur de la branche reliant j au nouveau nœud A comme étant
 - $L_i = d_{ij}/2 + (u_i - u_j)/2$
 - $L_j = d_{ij}/2 + (u_j - u_i)/2$
- (4) Calculer la distance entre le nouveau nœud A et chaque autre feuille comme
 - $d(ij),k = (d_{ik} + d_{jk} - d_{ij}) / 2$
- (5) Éliminer les colonnes et les lignes correspondant aux groupes i et j et ajouter celles correspondant au nouveau groupe (ij)
- (6) Si il reste un seul élément dans la matrice connecter les deux derniers nœuds, sinon retourner en 1







Comparaison UPGMA - NJ

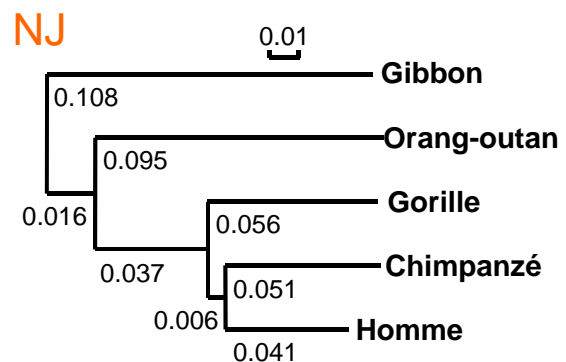
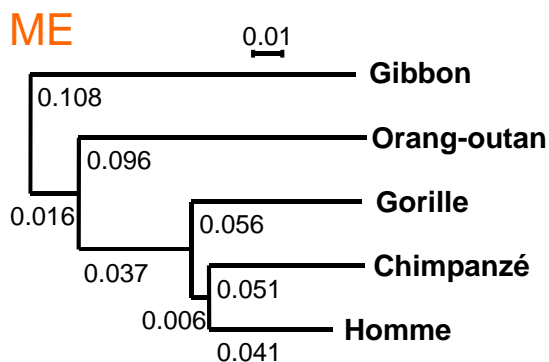
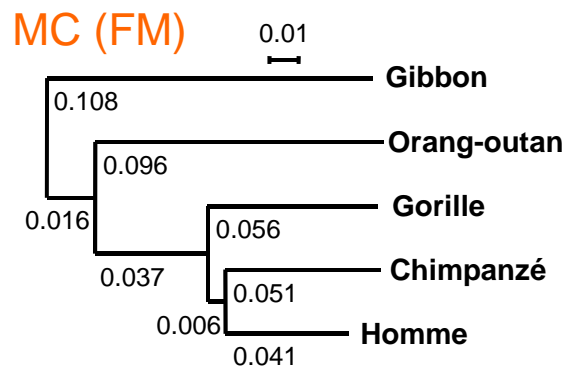
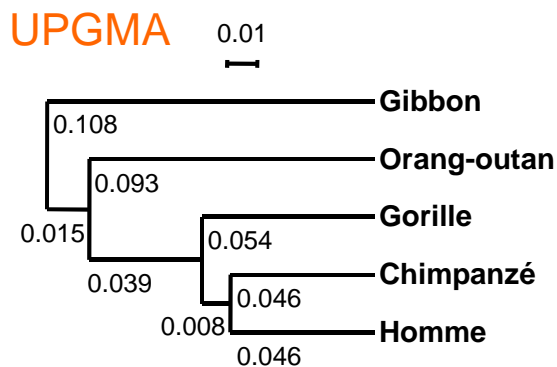


Exercices d'application

- Construisez les arbres par les différentes méthodes de distances pour la matrice de distances (K80) correspondant aux 5 séquences d'homoïdes

	Homme	Chimpanzé	Gorille	Orang-outang	Gibbon
Homme	0	0.092	0.106	0.177	0.207
Chimpanzé	0.092	0	0.111	0.193	0.218
Gorille	0.106	0.111	0	0.188	0.218
Orang-Outan	0.177	0.193	0.188	0	0.219
Gibbon	0.207	0.218	0.218	0.219	0

Solutions



Lectures conseillées

- « Biologie évolutive » Thomas, Lefèvre, Raymond (2010) de boeck
 - « Evolution » Barton, Briggs, Eisen, Goldstein, Patel (2007) Cold Spring Harbor Laboratory Press
 - « Concepts et méthodes en phylogénie moléculaire » Perrière & Brochier-Armanet (2010) Springer collection IRIS
-