Classification ascendante hiérarchique du lien moyen (UPGMA [1]) et arbres binaires aléatoires

1. unweighted pair-group method with arithmetic mean

L'objectif est d'écrire un programme et les fonctions python qui permettent de construire un arbre par classification ascendante hiérarchique du lien moyen à partir de la matrice de distances calculée préalablement. Vous n'utiliserez pas de bibliothèque pré-existante.

L'algorithme de cette méthode est donné ci-dessous (algorithme du livre de G. Perrière et C. Brochier[1] légèrement modifié).

1. Identifier les deux groupes pour lesquels la valeur de d_{ij} est minimale.

2. **Mettre à jour l'arbre**
   Créer un nouveau groupe C_u , de taille $n_u = n_i + n_j$, correspondant à $C_i \cup C_j$. Ce nouveau groupe ayant pour ancêtre le nœud u. et les branches reliant u à chacune des feuilles de C_i et de C_j sont de longueur $d_{ij}/2$.

3. **Mettre à jour la matrice**
   Calculer la distance entre C_u et chacun des k autres groupes présents dans la matrice D (exceptés C_i et C_j qui sont désormais sans signification) en utilisant la moyenne pondérée des distances :

$$d_{uk} = \frac{n_i}{n_i + n_j} d_{ik} + \frac{n_j}{n_i + n_j} d_{jk}$$

   Supprimer de D les lignes et colonnes qui correspondaient à C_i et C_j et a jouter la ligne et la colonne correspondant à C_u , avec les valeurs de d_{uk} calculées comme indiqué ci-dessus.

4. S'il reste un seul élément, arrêter, sinon retourner en 1.

1. Concepts et Méthodes En Phylogénie Moléculaire, Guy Perrière, Céline Brochier-Armanet, editeur : Springer Verlag France

Les données

Voici la matrice de distance que vous utiliserez pour vos tests. Recopier la dans votre programme sous forme de tableau 2D (liste de listes).

Table 1 – Distances calculées avec le modèle K80 entre les séquences d'ADN mitochondrial de cinq espèces d'Hominoïdes, exemple extrait du livre de G. Perrière et C. Brochier.

	Homme	Chimpanzé	Gorille	Orang-outan	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

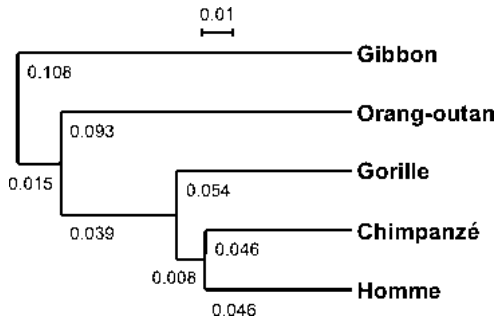

Figure 1 – Arbre obtenu par UPGMA avec la matrice de la table 1

Comment organiser votre programme

Remarques générales:
Les fonctions de manipulation d'arbre sont le plus souvent récursives.
Il est absolument indispensable de bien tester les fonctions au fur et à mesure qu'elles sont écrites.
Nous utiliserons une classe pour implémenter notre arbre, :

```
class Cl AB_:
    def  init_ ( s e l f ) : s e
        lf . id = ""
        s e l f . dist G=0
        self . dist D=0
        self .G = None
        self .D = None
```

```
self . pere = None
```

Les différents champs sont l'identifiant (id, de type *string*) la distance au fils gauche (distG, de type *float*) et au fils droit(distD, *float*), ainsi que le fils gauche (G, de type *Cl_AB*) et le fils droit (D, de type *Cl_AB*). C'est donc une structure de données autoréférentielle.

Étape 1

La première étape de l'algorithme nécessite une fonction qui recherche le minimum dans la matrice de distance (hors diagonale). Il faut donc écrire une fonction *trouverDistMin(mat)* qui prend comme argument la matrice de distances et retourne le couple d'indices (i,j) de la case contenant ce minimum.

Étape 2

La seconde étape est un peu plus complexe.

Au départ, nous avons autant d'arbres qu'il y a d'individus à regrouper, et nous allons fusionner ces arbres 2 par 2 jusqu'à n'en avoir plus qu'un.

Nous allons donc avoir besoin d'un fonction permettant de fusionner 2 arbres et de calculer la distance entre le nouveau nouveau noeud et les deux arbres (noeuds) qui sont fusionnés. Ainsi, dans l'exemple figure 1 et table 1, les deux premiers arbres fusionnés sont ceux du chimpanzé et de l'homme. La distance à gauche comme à droite est de 0.046 (0.092/2). Par contre, à l'étape suivante, les 2 arbres fusionnés sont le groupe (chimpanzé, homme) et le gorille ; la distance à gauche est de 0.008 et celle de droite de 0.054. En effet, la distance moyenne entre ces deux groupes est de 0.054 (cf. étape 3) et 0.054-0.046=0.08. Ainsi, la distance du nouveau noeud aux feuilles est des deux côtés de 0.054.

Vous écrirez la fonction *fusionAb(AG,AD,dij)* qui fusionne deux arbres, retourne le nouvel arbre, et met à jour les distances distG et distD pour que la distances du nouveau noeud aux feuilles soient de longueur dij. Vous aurez pour cela besoin d'écrire une fonction qui permet de calculer la distance s'un noeud à ses feuilles *lgBranche(A)*. Celle-ci prend don en argument un arbre et retourne la somme de toutes les distances entre la racine et une de ses feuilles.

Étape 3

Il faut dans cette étape mettre à jour la matrice ce qui sera fait dans une fonction *calcMatrice(i,j,matDist,A)*. Cette fonction prendre donc comme arguments *i* et *j* les indices des 2 lignes/colonnes (la matrice est carrée) qui doivent être fusionnées, *matDist*, la matrice de distances, qui sera modifiée, et *A*, l'arbre nouvellement construit à l'étape 2. Vous aurez besoin dans cette fonction des fonctions de manipulation de listes de python pour ôter et ajouter les lignes et les colonnes.

list.append(x)

list.insert(i, x)

list.pop(i)

Attention lors du calcul des nouvelles distances à bien pondérer par le nombre de feuilles (n_i et n_j des formules). Il vous faudra donc écrire une autre fonction permettant de compter le nombre de feuilles d'un arbre.

Les fonctions nécessaires aux 3 étapes étant écrites, il vous reste à écrire la fonction *UPGMA(matDist,ab)* qui à partir d'une matrice de distance et d'une liste d'arbre contenant un arbre feuille pour chaque séquence, calcule et retourne l'arbre des toutes les séquences.

Affichage

Lorsque la construction de l'arbre marchera, il faudra alors écrire la fonction permettant d'écrire l'arbre au format parenthésé Newick.. Pour afficher l'arbre, vous pourrez utiliser le programme njplot `http://pbil.univ-lyon1.fr/software/njplot.html`. (('Homme' :0.046, 'Chimpanze' :0.046) : 0.00825, 'Gorille' : 0.05425) est le sous arbre (homme, Chimpanzé, Gorille) de l'arbre représenté figure 1.