Fiche de Biostatistique – Stage 7

Introduction à la classification hiérarchique

D. Chessel, J. Thioulouse & A.B. Dufour

Résumé

La fiche donne les principes généraux de la classification automatique. L'essentiel est consacré à la description des fonctions **`hclust`** et **`kmeans`** dans R.

Plan

1. Définitions : parties, partitions et hiérarchies

La bibliographie sur les méthodes de classification automatique est abondante. A titre d'exemple, celle qui est citée dans le logiciel R pour la fonction **hclust** du package **cluster** est la suivante :

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) The New S Language. Wadsworth & Brooks/Cole. (S version.)

Everitt, B. (1974). Cluster Analysis. London: Heinemann Educ. Books.

Hartigan, J. A. (1975). Clustering Algorithms. New York: Wiley.

Sneath, P. H. A. and R. R. Sokal (1973). Numerical Taxonomy. San Francisco: Freeman.

Anderberg, M. R. (1973). Cluster Analysis for Applications. Academic Press: New York.

Gordon, A. D. (1999). Classification. Second Edition. London: Chapman and Hall / CRC

Murtagh, F. (1985). "Multidimensional Clustering Algorithms", in COMPSTAT Lectures 4. Wuerzburg: Physica-Verlag (for algorithmic details of algorithms used).

Pour les francophones, on peut ajouter :

Benzecri, J.P. (1973). L'analyse des données. T1 : La taxinomie. Dunod.

Roux, M. (1985). Algorithmes de classification. Masson.

Diday, E., J. Lemaire, J. Pouget, and F. Testu. 1982. Elements d'analyse de données. Dunod, Paris.

Lebart, L., A. Morineau, and M. Piron. 1995. Statistique exploratoire multidimensionnelle. Dunod, Paris.

Parmi les références historiques, on notera :

Sokal, R. R., and P. H. A. Sneath. 1963. Principles of numerical taxonomy. Freeman and Co., San-Francisco.

Cormack, R. M. 1971. A review of classification. Journal of the Royal Statistical Society, A **134**:321-367.

Whittaker, R. H. 1973. Handbook of vegetation science. Part V. Ordination and classification of communities. Dr. W. Junk b.v., The Hague.

L'objectif principal des méthodes de classification automatique est de répartir les éléments d'un ensemble en groupes, c'est-à-dire d'établir une partition de cet ensemble. Différentes contraintes sont bien sûr imposées, chaque groupe devant être le plus homogène possible, et les groupes devant être les plus différents possibles entre eux.

De plus, on ne se contente pas d'une partition, mais on cherche une hiérarchie de parties, qui constituent un arbre binaire appelé le dendrogramme. Quelques définitions de bases sont donc indispensables. On considère ici des ensembles finis, donc des collections d'objets au sens habituel. ***A*** est un **ensemble** *:*

$$A = \{a_1, a_2, ..., a_n\} \Leftrightarrow a_j \in A \text{ pour } 1 \leq j \leq n$$

Une **partie** de A est un sous-ensemble :

$$B = \{b_1, b_2, ..., b_p\} \subseteq A \Leftrightarrow b_k \in A \text{ pour } 1 \leq k \leq p$$

Si on compte la partie vide et l'ensemble tout entier, il y a dans A 2^n parties. L'**ensemble de toutes les parties** de A se note $\mathfrak{P}(A)$. Si A est formé de *a, b, c et d,* $\mathfrak{P}(A)$ compte 16 éléments qui sont :

$$\varnothing$$
$$\{a\},\{b\},\{c\},\{d\}$$
$$\{a,b\},\{a,c\},\{a,d\},\{b,c\},\{b,d\},\{c,d\}$$
$$\{a,b,c\},\{a,b,d\},\{a,c,d\},\{b,c,d\}$$
$$\{a,b,c,d\}$$

L'ensemble des parties est muni de la relation d'**ordre partiel** défini par :

$$X \subseteq Y \Leftrightarrow (x \in X \Rightarrow x \in Y)$$

L'ordre est partiel car si il est vrai que :

$$\{a,d\} \subseteq \{a,c,d\}$$

les deux assertions suivantes sont fausses et les deux parties ne sont pas comparables :

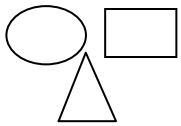
$$\{a,b,d\} \subseteq \{a,c,d\} \qquad \{a,c,d\} \subseteq \{a,b,d\}$$

Deux parties d'un ensemble sont soit chevauchantes (non égales et d'intersection non nulle), soit disjointes (sans élément commun, d'intersection nulle), soit incluses l'une dans l'autre, soit égales :

chevauchantes *disjointes* *incluses* *égales*

Une **partition** est un sous-ensemble de parties deux à deux disjointes dont la réunion fait l'ensemble tout entier.

$$\{A_1, A_2, ..., A_K\} \text{ partition de } A$$
$$\Updownarrow$$
$$i \neq j \Rightarrow A_i \cap A_j = \varnothing$$
$$\bigcup_{k=1}^{K} A_k = A$$



$$\{\{a,e,f,g\},\{b\},\{c,d\}\} \text{ est une partition de } \{a,b,c,d,e,f,g\}$$

Une partition équivaut à une **variable qualitative** ou *factor* définie sur les éléments de l'ensemble.

```
w1
 [1] bleu  vert  vert  jaune vert  bleu  jaune rouge rouge rouge vert  vert
[13] bleu  jaune vert  vert  vert  bleu  bleu  jaune rouge rouge rouge
Levels: bleu jaune vert rouge

w2 = split(1:23,w1)
w2

$bleu
```

```
[1]  1  6 13 18 19

$jaune
[1]  4  7 14 20

$vert
[1]  2  3  5 11 12 15 16 17

$rouge
[1]  8  9 10 21 22 23
```

Les composantes de la liste sont les parties, les noms des composantes sont les niveaux du facteur. Les méthodes d'ordination fournissent, comme leur nom l'indique, une ordination des individus : elles résument les données par un (ou plusieurs) score numérique (gradients des écologues ou variable latente des psychométriciens). Les méthodes de classification résument les données par une variable qualitative. Elles fournissent des partitions. Il n'y a pas de bonnes ou de mauvaises méthodes, mais des outils plus ou moins utiles pour parler des données. On peut les utiliser simultanément comme, par exemple, en représentant les groupes d'individus obtenus par classification sur le plan factoriel issu d'une méthode d'ordination.

Deux parties d'une partition d'un ensemble sont soit disjointes, soit égales. La relation d'inclusion entre parties se généralise à la relation de finesse entre partitions.

$$\{A_1, A_2, ..., A_K\} \text{ partition de } A \quad \{B_1, B_2, ..., B_L\} \text{ partition de } A$$
$$\{A_1, A_2, ..., A_K\} \prec \{B_1, B_2, ..., B_L\}$$
$$\Updownarrow$$
$$1 \le k \le K \Rightarrow \exists l \; 1 \le l \le L \text{ telle que } A_k \subseteq B_l$$

Une partition moins fine est, autre désignation, plus grossière. Par exemple :

$$\{\{a\},\{b\},\{c\},\{d\},\{e\}\} \prec \{\{a,b\},\{c,d\},\{e\}\} \prec \{\{a,b,c\},\{d,e\}\} \prec \{\{a,b,c,d,e\}\}$$

Un ensemble quelconque de parties est formée de parties chevauchantes, disjointes ou incluses. Un ensemble de parties formant une partition ne comporte que des parties disjointes recouvrant le tout. Entre ces deux classes, la première trop large pour être utile et la seconde trop étroite pour être nuancée, on trouve les hiérarchies de parties.

Une **hiérarchie** de partie de A est un ensemble de parties ayant quatre propriétés :

1) La partie vide en fait partie
2) Les parties réduites à un seul élément en font partie.
3) L'ensemble total *A* lui-même en fait partie.
4) Si *X* et *Y* en font partie, alors soit X et Y sont disjointes, soit X contient Y, soit Y contient X.

Par exemple, l'ensemble :

$$\{\{a\},\{b\},\{c\},\{d\},\{e\},\{a,b\},\{e,d\},\{a,b,c,d,e\}\}$$

est une hiérarchie de parties ou encore un n-arbre (Gordon, op. cit. p.69) :

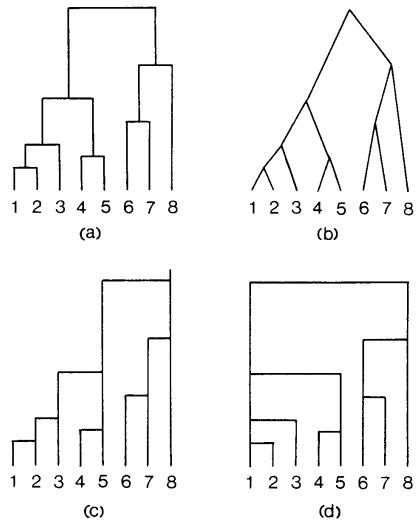
Un arbre est un graphe raciné : les feuilles sont les parties à un seul élément (qui sont toujours dans une hiérarchie), la racine est l'ensemble tout entier (qui est toujours dans la hiérarchie). Chaque

partie n'a qu'un ancêtre, à l'exclusion de la racine qui n'en n'a pas (sinon on trouverait deux parties chevauchantes ce qui n'existe pas dans une hiérarchie). Si l'arbre est binaire, chaque partie a deux descendants, à l'exclusion des feuilles qui n'en n'ont pas. On dit aussi que la hiérarchie est alors **totalement résolue**.

La hiérarchie est **valuée** si à chaque partie on peut associer une valeur numérique qui vérifie la définition :

$$X \subseteq Y \Leftrightarrow f(X) \leq f(Y)$$

Cette valeur place les feuilles tout en bas et la racine tout en haut. La représentation graphique d'une hiérarchie valuée s'appelle un **dendrogramme**. Il est essentiel de comprendre d'entrée que cette représentation est très peu contrainte :



A gauche on a une hiérarchie valuée formée des parties :

$$\left\{ \begin{array}{c} \varnothing, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\} \\ a = \{1,2\}, b = \{1,2,3\}, c = \{4,5\}, d = \{1,2,3,4,5\} \\ e = \{6,7\}, f = \{6,7,8\}, g = \{1,2,3,4,5,6,7,8\} \end{array} \right\}$$

A droite se trouvent quatre représentations possibles parmi un très grand nombre (Gordon, *op. cit.* p. 72). La présente fiche introduit à la recherche d'une hiérarchie valuée pour décrire des données numériques puis à celle d'une partition pour les résumer.

2. Distances entre individus

La recherche d'une hiérarchie valuée s'appelle une classification hiérarchique (*hierarchical clustering*). Une telle recherche s'appuie sur une notion de distances entre individus qui induit une mesure de **l'hétérogénéité** d'une partie basée sur les distances entre individus qui sont dedans et une mesure de **dissimilarité** entre deux parties basée sur la distance entre un individu de l'un et un individu de l'autre.

2.1. Distances écologiques

Il existe une multitude de manière de calculer des distances entre objets. Cormack (1971, *op. cit. summary*) parle de *burgeoning bibliography* et de *plethora of definitions of similarity*. Les données en présence-absence peuvent être transformées en matrices de distance. Deux objets (en écologie, lignes ou colonnes d'un tableau floristique ou faunistique) sont comparés sur une liste de valeurs. Ces valeurs sont réduites en 0-1 (1 si la valeur est strictement positive, 0 sinon). Deux relevés sont ainsi comparés par la liste des espèces présentes, deux espèces sont comparées par la liste des relevés dans lesquels elles sont présentes. Ces listes ont la forme :

```
0110000101 0010...
0101000110 0010...
```

On note *n* le nombre d'enregistrements qui est la somme de

a est le nombre de concordances 11
b le nombre de concordances 10
c le nombre de concordances 01
d le nombre de concordances 00.

Ainsi deux espèces sont présentes ensemble dans un même relevé *a* fois, deux relevés possèdent *a* espèces en commun. Les deux objets définissent donc la table de contingence 2-2 :

	1	0	Tot
1	a	b	$a+b$
0	c	d	$c+d$
Tot	$a+c$	$b+d$	n

Les quatre nombres de la table définissent une similarité entre les deux objets. On peut utiliser :

$S_1 = \frac{a}{a+b+c}$ Indice de communauté de Jaccard

$S_2 = \frac{a+d}{n}$ Indice de Sokal & Michener

$S_3 = \frac{a}{a+2(b+c)}$ Indice de Sokal & Sneath

$S_4 = \frac{a+d}{a+2(b+c)+d}$ Indice de Rogers et Tanimoto

$S_5 = \frac{2a}{2a+b+c}$ Indice de Sorensen

$S_6 = \frac{a-b-c+d}{n}$ Indice de Gower & Legendre

$S_7 = \frac{a}{\sqrt{(a+b)(a+c)}}$ Indice de Ochiai

$S_8 = \frac{ad}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$ Indice de Sockal & Sneath

$S_9 = \frac{ad-bc}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$ Phi de Pearson

$$S_{10} = \frac{a}{n}$$ avec l'unité si les deux objets sont identiques

On trouvera les références d'origine et les propriétés principales dans **1**. Ces indices sont tous inférieurs ou égaux à 1 et la distance associée est définie par :

$$D_k = \sqrt{1 - S_k}$$

Par exemple, pour l'indice de Jaccard, si deux relevés sont identiques, on a bien D = 0, et si deux relevés sont complètement différents (aucune espèce en commun), on a D = 1.

On peut les calculer par la fonction **dist.binary** dans **ade4** qui demandera de choisir :

```
1 = JACCARD index (1901) S3 coefficient of GOWER & LEGENDRE
s1 = a/(a+b+c) --> d = sqrt(1 - s)
2 = SOCKAL & MICHENER index (1958) S4 coefficient of GOWER & LEGENDRE
s2 = (a+d)/(a+b+c+d) --> d = sqrt(1 - s)
3 = SOCKAL & SNEATH(1963) S5 coefficient of GOWER & LEGENDRE
s3 = a/(a+2(b+c)) --> d = sqrt(1 - s)
4 = ROGERS & TANIMOTO (1960) S6 coefficient of GOWER & LEGENDRE
s4 = (a+d)/(a+2(b+c)+d) --> d = sqrt(1 - s)
5 = CZEKANOWSKI (1913) or SORENSEN (1948) S7 coefficient of GOWER & LEGENDRE
s5 = 2*a/(2*a+b+c) --> d = sqrt(1 - s)
6 = S9 index of GOWER & LEGENDRE (1986)
s6 = (a-(b+c)+d)/(a+b+c+d) --> d = sqrt(1 - s)
7 = OCHIAI (1957) S12 coefficient of GOWER & LEGENDRE
s7 = a/sqrt((a+b)(a+c)) --> d = sqrt(1 - s)
8 = SOKAL & SNEATH (1963) S13 coefficient of GOWER & LEGENDRE
s8 = ad/sqrt((a+b)(a+c)(d+b)(d+c)) --> d = sqrt(1 - s)
9 = Phi of PEARSON = S14 coefficient of GOWER & LEGENDRE
s9 = ad-bc)/sqrt((a+b)(a+c)(b+d)(d+c)) --> d = sqrt(1 - s)
10 = S2 coefficient of GOWER & LEGENDRE
s10 =  a/(a+b+c+d) --> d = sqrt(1 - s) and unit self-similarity
Select an integer (1-10): 0
```

Les données compilées par B. Hugueny (`hugueny@biomserv.univ-lyon1.fr`) **2** pour l'objet **westafrica** représentent cette tradition.

```
data(westafrica)
names(west africa)
dim(westafrica$tab)
```

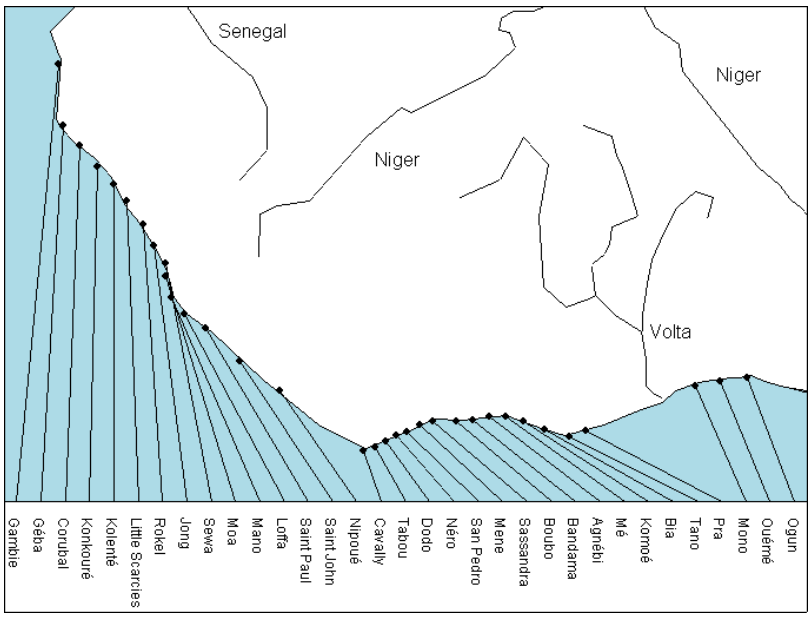
```
[1] 268  33
```

33 bassins des fleuves côtiers de l'Afrique de l'Ouest sont représentés par leur embouchure sur la figure reproductible dans R à partir de la carte de documentation de l'objet :

---

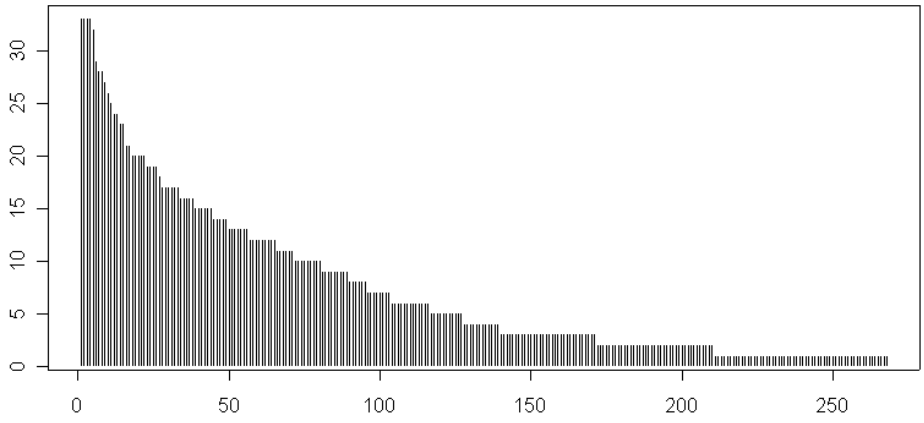
1 Gower, J.C. & Legendre, P. (1986) Metric and Euclidean properties of dissimilarity coefficients. *Journal of Classification* : 3, 5-48.

2 Paugy, D., Traoré, K. and Diouf, P.F. (1994) Faune ichtyologique des eaux douces d'Afrique de l'Ouest. In Diversité biologique des poissons des eaux douces et saumâtres d'Afrique. Synthèses géographiques, Teugels, G.G., Guégan, J.F. and Albaret, J.J. (Editors). *Annales du Musée Royal de l'Afrique Centrale, Zoologie*, N° 275, Tervuren, Belgique, 35-66.
Hugueny, B. (1989) Biogéographie et structure des peuplements de Poissons d'eau douce de l'Afrique de l'ouest : approches quantitatives. Thèse de doctorat, Université Paris 7.



268 espèces de Poissons ont été observées : chacune d'entre elles est présente ou absente dans chacun des bassins. On a un tableau faunistique avec 268 lignes-espèces et 33 colonnes-bassins.

```
freq=apply(westafrica$tab,1,sum)
freq=rev(sort(freq))
plot(freq,type="h")
```



Le graphe rang-fréquences, un objet traditionnel de l'écologie statistique [3]

Pour le biogéographe, les espèces sont des *marqueurs*, variables qui fabriquent de la différence sans qu'on ait besoin d'en connaître l'interprétation. Elles génèrent une distance entre sites :

```
westafrica.d=dist.binary(as.data.frame(t(westafrica$tab)),1)
westafrica.d
```

```
          GAMBIE   GEBA CRUBAL KONKOURE KOLENTE LSCARC  ROKEL   JONG   SEWA
GEBA      0.7468
CRUBAL    0.7827 0.7029
KONKOURE  0.8537 0.7983 0.7153
KOLENTE   0.8537 0.7888 0.7385   0.5695
LSCARC    0.8896 0.8461 0.7571   0.5816  0.6391
ROKEL     0.8885 0.8442 0.7762   0.6500  0.6138 0.6299
JONG      0.8410 0.7930 0.7148   0.5625  0.5984 0.6283 0.6040
SEWA      0.8636 0.8379 0.7845   0.6751  0.6447 0.7071 0.6713 0.5661
MOA       0.8774 0.8283 0.7896   0.6391  0.6391 0.6911 0.7018 0.6751 0.6063
```

[3] Daget, J. 1976. Les modèles mathématiques en écologie. Masson, Paris.

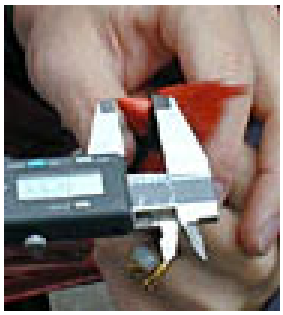
...

2.2. Distances morphométriques

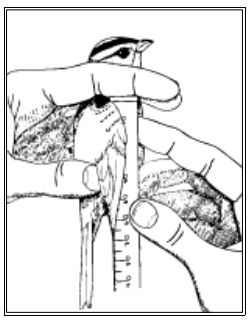
La morphométrie[4], qui se consacre aux variations de taille et de forme entre êtres vivants ou disparus, utilise soit des mesures quantitatives soit des points de repère (*landmarks*[5]). Les points de repère (en particulier les points de contour) permettent de définir la distance entre deux individus par la distance canonique entre leur ajustement à une même configuration de référence par des transformations procustéennes. On trouvera une introduction dans *http://pbil.univ-lyon1.fr/R/fichestd/tdr64.pdf*. Sur les mesures traditionnelles, l'élimination de la taille avant le calcul de distances se fait par le biais de la métrique de Mahalanobis dite encore métrique généralisée.

Les données réunies en France, en Californie et au Chili par des ornithologues[6] portent sur 129 espèces d'oiseaux et sont consignées dans la liste **ecomor** :

```
data(ecomor)
names(ecomor)
[1] "forsub"  "diet"    "habitat" "morpho"  "taxo"    "labels"  "categ"

ecomor$morpho
     wingl taill culml bilh billw tarsl midtl  weig
E033  44.1  27.7  22.6  2.3   2.7   2.7   6.1   3.1
E034  50.0  32.4  23.0  2.3   3.6   2.8   7.1   4.3
E035  47.0  25.2  20.8  2.3   3.1   2.5   6.4   3.4
E070 132.4  88.5  43.8  3.7   7.0   6.7  14.8  18.5
E071  63.8  42.4  22.7  2.4   3.9   5.3   8.6   5.8
E001 255.0 173.0  24.7 11.0  11.0  26.2  47.0 500.0
E031 213.5 162.0  26.3  7.2   8.7  25.7  42.1 408.5
E100 184.0 118.0  32.0  5.4   6.8  19.0  30.5 138.2

...
```



7



8

```
ecomor$labels
     latin                          abbr
E033 "Archilochus alexandri"        "Arc|ale"
E034 "Calypte anna"                 "Cal|ann"
```

---

4 Voir les définitions principales à http://life.bio.sunysb.edu/morph/glossary/gloss1.html

5 Bookstein, F. L. 1991. Morphometric Tools for Landmark Data. Geometry and Biology. Cambridge University Press: New York.

6 Blondel, J., F. Vuilleumier, L. F. Marcus, and E. Terouanne. 1984. Is there ecomorphological convergence among mediterranean bird communities of Chile, California, and France. Pages 141-213 *in* M. K. Hecht, B. Wallace, and R. J. MacIntyre, editors. Evolutionary Biology. Vol. 18. Plenum Press, New York.

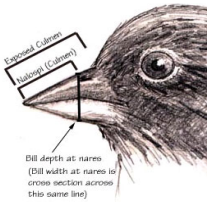
7 http://sio.si.edu/Nestwatch/What_is_Nestwatch/Catching_the_Birds/rulers_and_calipers.cfm

8 http://cm27personal.fal.buffalo.edu/birds/anatomy/molt/size.html

```
E035 "Calypte costae"                "Cal|cos"
E070 "Patagona gigas"                "Pat|gig"
E071 "Sephaniodes sephaniodes"       "Sep|sep"
E001 "Columba palumbus"              "Col|pal"
...
```

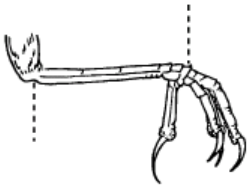
morpho définit la morphologie des oiseaux. Il contient les variables

`wingl`	longueur de l'aile en mm (*Wing length*)
`taill`	longueur de la queue en mm (*Tail lenght*)
`culm`	longueur du bec en mm (*Culmen length*)
`bilh`	hauteur du bec en mm (*Bill height*)
`billw`	largeur du bec en mm (*Bill width*)
`tarsl`	hauteur du tarse en mm (*Tarsus length*)
`midtl`	longueur de l'orteil médian en mm (*Middle toe length*)
`weig`	poids en g (*Weight*)



9


10


8

On travaille en général après transformation logarithmique :

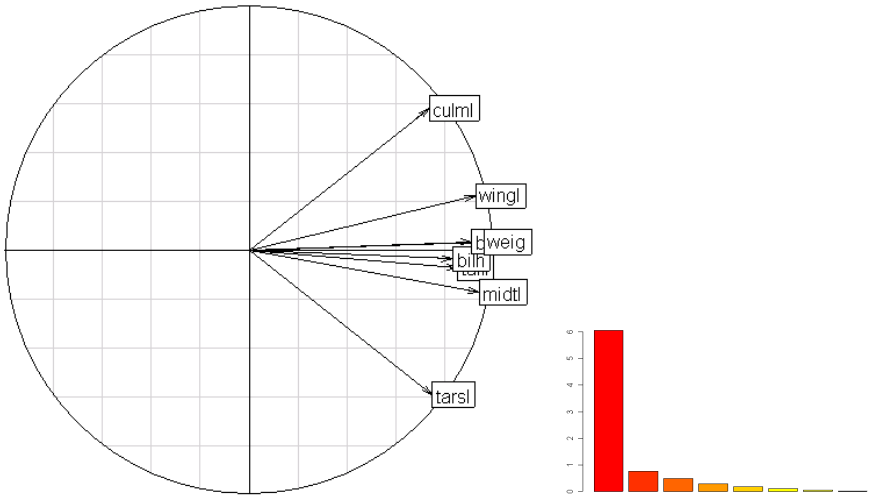
```
molo=log(ecomor$morpho)
```

Quand on décrit la différence entre deux espèces, on peut utiliser la distance canonique :

$$d_{ij}^{cano} = \sqrt{\sum_{k=1}^{p}\left(x_{ik}-x_{jk}\right)^2} \Leftrightarrow d_{ij}^{2\,cano} = \sum_{k=1}^{p}\left(x_{ik}-x_{jk}\right)^2$$

Lorsque les variables sont corrélées entre elles (effet taille) :

```
s.corcircle(dudi.pca(molo)$co)
```

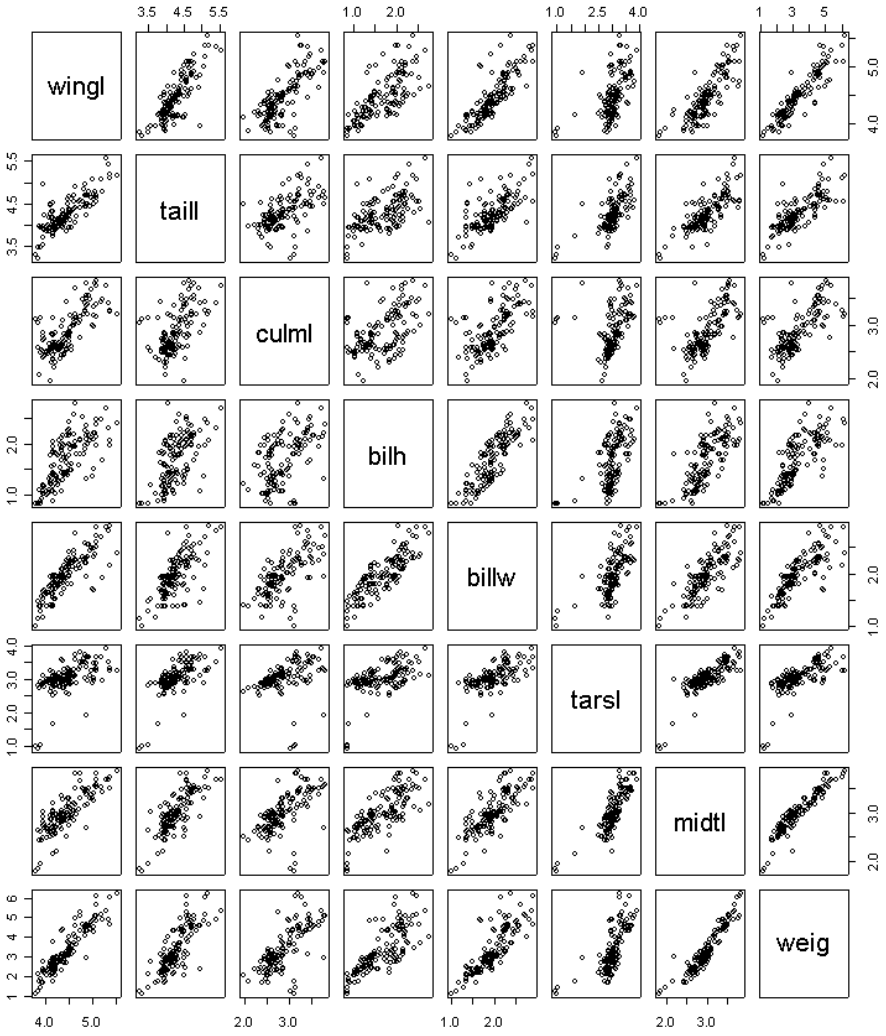


9 http://home.pacifier.com/~neawanna/observatory/sparrow/measure.html

10 http://www.ummz.lsa.umich.edu/birds/WOSManual/7.FormandFunction.pdf

une différence de taille entre deux espèces est enregistrée sur chacune des variables et ce fait élimine numériquement les autres nuances.

```
pairs(molo)
```



Remarque : la variable tarsl est mal conditionnée.

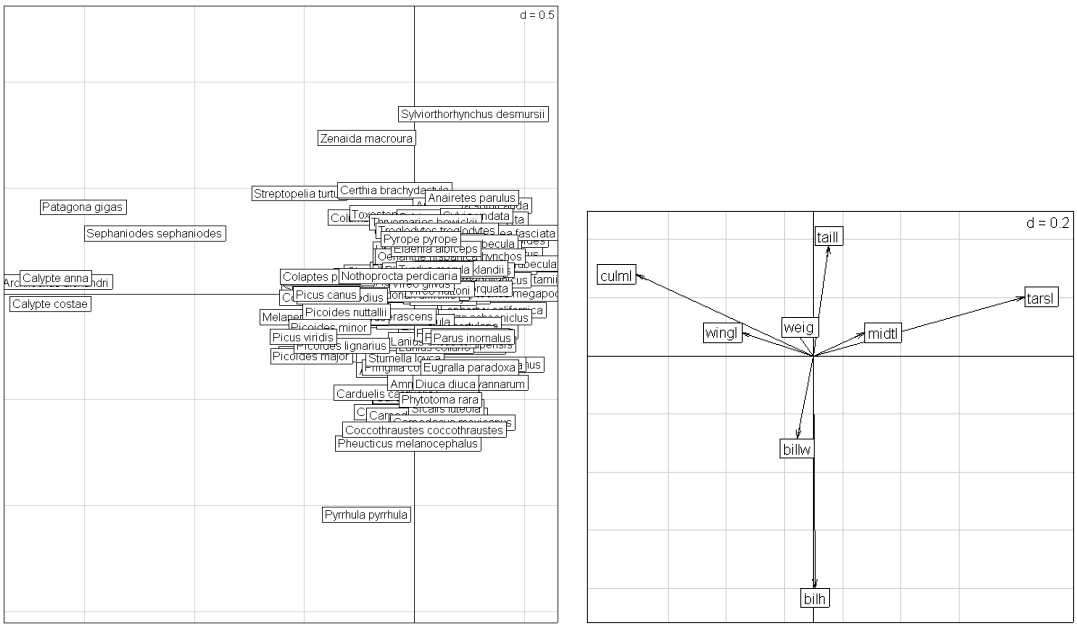
```
apply(molo,2,var)
 wingl  taill  culml   bilh  billw  tarsl  midtl   weig
0.1503 0.1595 0.1571 0.2237 0.1558 0.2081 0.1603 1.2134
```

En outre toutes les variables ont des variances comparables : le poids fait exception. Or le poids est en g, équivalent d'un mm^3 en dimension, il aurait fallu travailler avec la racine cubique du poids, donc avec le tiers du logarithme, donc avec une variance 9 fois plus petite.

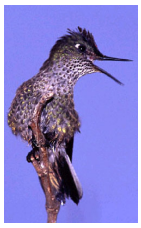
```
molo$weig=molo$weig/3
apply(molo,2,var)
 wingl  taill  culml   bilh  billw  tarsl  midtl   weig
0.1503 0.1595 0.1571 0.2237 0.1558 0.2081 0.1603 0.1348
```

Reste à se débarrasser de la corrélation dans la mesure des différences inter spécifiques, c'est-à-dire minimiser la question de la taille au profit de celle de la forme [11]. On peut centrer les logarithmes par individus, deux individus en relation d'isométrie se retrouve alors à la même valeur :

```
molocl=as.data.frame(t(apply(molo,1,function(x) return(x-mean(x)))))
molocl.pca=dudi.pca(molocl,scale=F,scannf=F)
s.arrow(molocl.pca$co)
s.label(molocl.pca$li,lab=ecomor$labels[,"latin"])
```



Les cinq espèces à gauche : tout dans le bec, rien dans les pattes :

 **12**  **13**  **14**  **15**  **16**

http://www.hbw.com/ibc/phtml/buscar.phtml

Les trois espèces les plus à droite : strictement l'inverse :

---

11 Yoccoz, N. G. 1993. Morphométrie et analyses multidimensionelles. Une revue des méthodes séparant taille et forme. Pages 73-99 in J. D. Lebreton and B. Asselain, editors. Biométrie et Environnement. Masson, Paris.

12 http://www.oceanoasis.org/fieldguide/caly-ann.html

13 http://www.avesphoto.com/website/n0209CAM/species/n0209CAM-4.htm

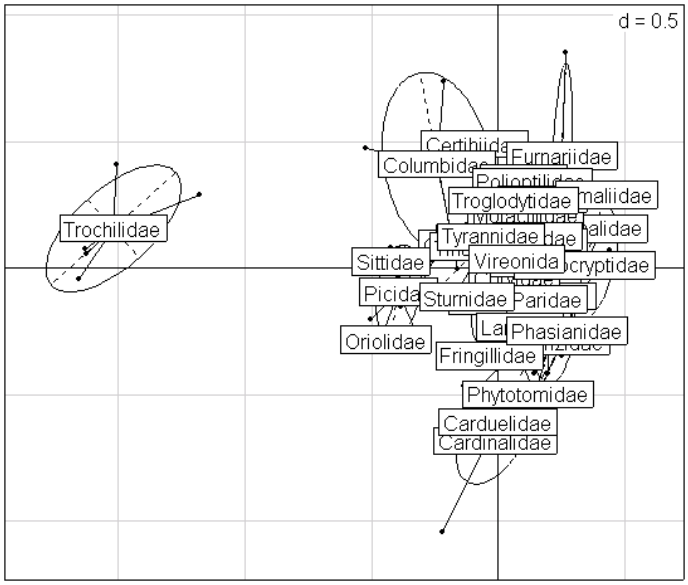
14 http://weaselhead.org/learn/birds_black-chinned_hummingbird.asp

15 http://www.agualtiplano.net/bases/animales/62.htm

16 http://www.greglasley.net/gbfire.html

 **17**  **18**  **19**

```
s.class(molocl.pca$li,ecomor$taxo$Family)
```



Une famille très spéciale

```
d1 = dist.quant(molocl)
1 = Canonical
d1 = ||x-y|| A=Identity
2 = Joreskog
d2=d2 = ||x-y|| A=1/diag(cov)
3 = Mahalanobis
d3 = ||x-y|| A=inv(cov)
Select an integer (1-3): 1
```

Les données acquises et rendues disponibles par J.M. Lascaux **20** forment l'objet **lascaux**.

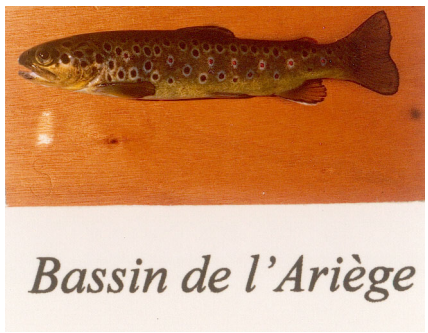
```
data(lascaux) # voir http://pbil.univ-lyon1.fr/R/fichestd/TDR61.pdf
```

17 http://www.avesdechile.cl/170.htm

18 http://www.avesdechile.cl/174en.htm

19 http://stockpix.com/stock/animals/birds/songbirds/wrens/4484.htm

20 Lascaux, J. M. 1996. Analyse de la variabilité morphologique de la truite commune (Salmo trutta L.) dans les cours d'eau du bassin pyrénéen méditerranéen. Thèse de doctorat en sciences agronomiques, INP Toulouse.




```
la = as.data.frame(t(na.omit(t(lascaux$morpho))))
la = log(la)
names(la)
```

```
LS      Longueur standard
MD      Distance bout du museau - insertion de la dorsale
MAN     Distance bout du museau - insertion de l'anale
MPEL    Distance bout du museau - insertion de la pelvienne
MPEC    Distance bout du museau - insertion de la pectorale
DAD     Distance insertion de la dorsale - insertion de l'adipeuse
DAN     Distance insertion de la dorsale - insertion de l'anale
DPEC    Distance insertion de la dorsale - insertion de la pectorale
ADC     Distance insertion de l'adipeuse - départ de la caudale
ADAN    Distance insertion de l'adipeuse - insertion de l'anale
ADPEC   Distance insertion de l'adipeuse - insertion de la pectorale
PECPEL  Distance insertion de la pectorale - insertion de la pelvienne
PELAN   Distance insertion de la pelvienne - insertion de l'anale
ANC     Distance insertion de l'anale - départ de la caudale
LPRO    Longueur préorbitale
DO      Diamètre de l'orbite
LPOO    Longueur postorbitale
LTET    Longueur de la tête
LAD     Longueur de l'adipeuse
LD      Longueur de la dorsale
LC      Longueur de la caudale
LAN     Longueur de l'anale
HAN     Hauteur de l'anale
LPELG   Longueur de la pelvienne gauche
LPECG   Longueur de la pectorale gauche
ETET    Largeur de la tête (au niveau des orbites)
```

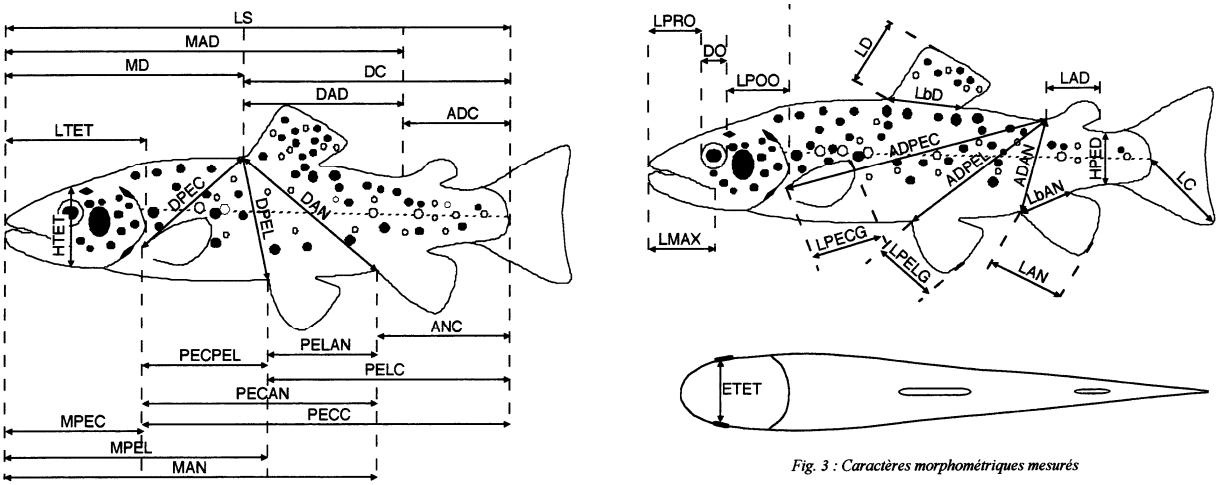
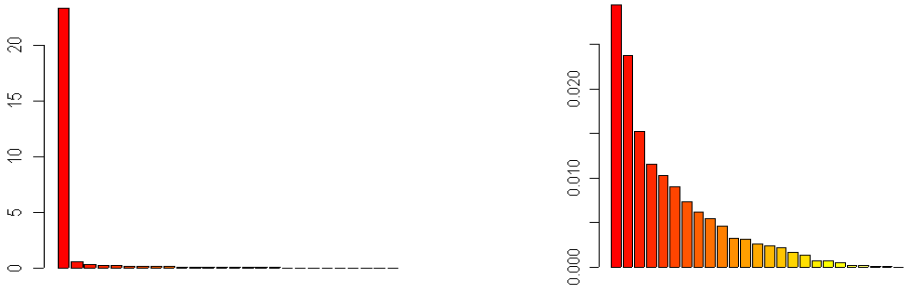


Fig. 3 : Caractères morphométriques mesurés

Après centrage des logarithmes, il reste une structure à interpréter.

```
dudi.pca(la)
lacl=as.data.frame(t(apply(la,1,function(x) x-mean(x))))
dudi.pca(lacl,scale=F)
```



Pour avoir un bon résumé de la distance morphométriques entre les 306 truites :

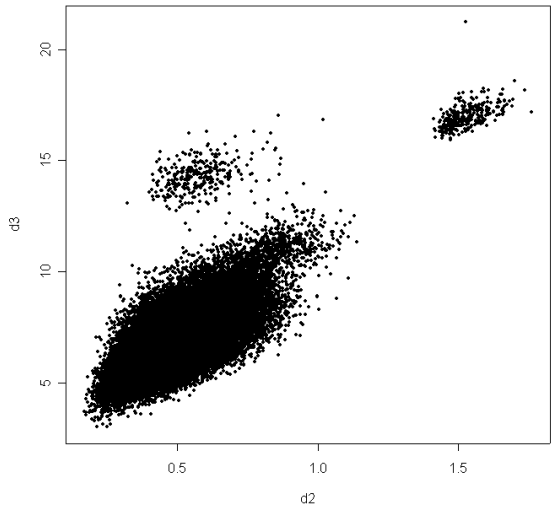
```
d2 = dist.quant (lacl,1)
```

Une stratégie alternative passe par la métrique de Mahalanobis :

```
d3 = dist.quant (la,3)
```

Ce n'est pas la même chose :

```
plot(d2,d3,pch=20)
```

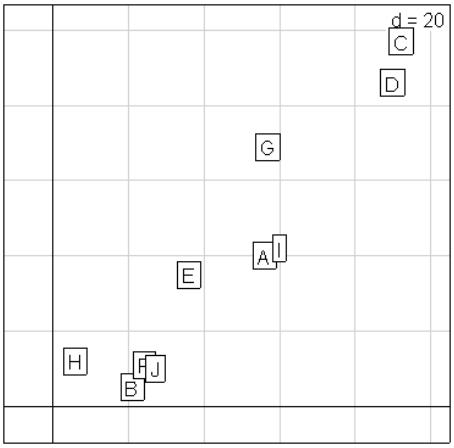


Ce nuage de points comporte 46665 points(couple de deux individus) :

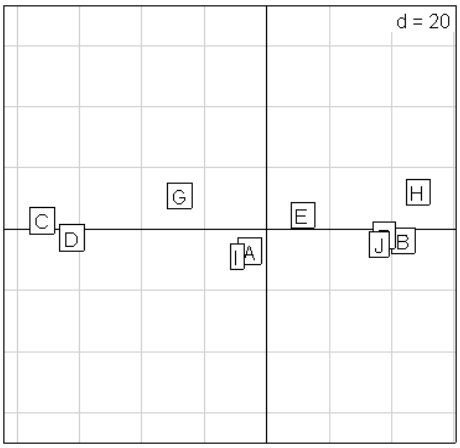
```
length(d3)
[1] 46665
 306*305/2
[1] 46665
```

La métrique de Mahalanobis n'élimine pas complètement l'effet taille : elle en modifie seulement l'importance relative. Le calcul est simple. Pour le comprendre, prenons 10 points :

```
x = c(56,21,92,90,36,24,57,6,60,27)
y = c(40,5,97,86,35,11,69,12,42,10)
xy=data.frame(x,y)
row.names(xy)=LETTERS[1:10]
s.label(xy) #A
```



A



B

```
dist.quant(xy,1)
```

```
      A       B       C       D       E       F       G       H       I
B  49.497
C  67.417 116.211
D  57.201 106.405  11.180
E  20.616  33.541  83.546  74.277
F  43.186   6.708 109.636  99.905  26.833
G  29.017  73.430  44.822  37.121  39.962  66.731
H  57.306  16.553 120.917 111.946  37.802  18.028  76.485
I   4.472  53.759  63.632  53.254  25.000  47.508  27.166  61.774
J  41.725   7.810 108.600  98.717  26.571   3.162  66.189  21.095  45.967
```

La distance de G à C (44.8) est plus grande que la distance de G à I (27.2). Ceci vient de la corrélation entre les deux variables : une différence de taille compte double : c'est la redondance. Si on fait l'ACP centrée du tableau, on change de repères mais pas de distances. Le sens des axes n'a aucune importance :

```
w1=dudi.pca(xy,scale=FALSE,scannf=FALSE)
s.label(w1$li) #B
dist.quant(w1$li,1)
```

```
      A       B       C       D       E       F       G       H       I
B  49.497
C  67.417 116.211
D  57.201 106.405  11.180
E  20.616  33.541  83.546  74.277
F  43.186   6.708 109.636  99.905  26.833
G  29.017  73.430  44.822  37.121  39.962  66.731
H  57.306  16.553 120.917 111.946  37.802  18.028  76.485
I   4.472  53.759  63.632  53.254  25.000  47.508  27.166  61.774
J  41.725   7.810 108.600  98.717  26.571   3.162  66.189  21.095  45.967
```

La distance de Mahalanobis est obtenue en substituant les coordonnées normalisées (les composantes principales) aux coordonnées simples :

```
dist.quant(w1$l1,1)
```

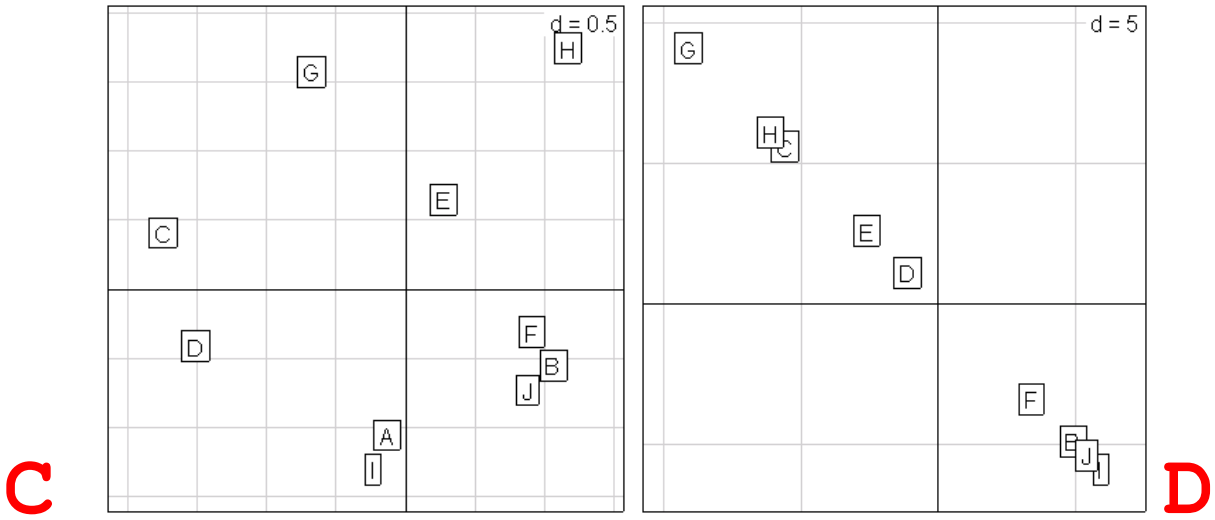
```
      A      B      C      D      E      F      G      H      I
B 1.2951
C 2.1755 2.9627
D 1.5203 2.5746 0.8526
E 1.7513 1.4332 2.0326 2.0771
F 1.2754 0.2866 2.7422 2.4160 1.1466
G 2.6831 2.7438 1.5782 2.1524 1.3274 2.4590
H 3.0843 2.2942 3.2036 3.4383 1.4121 2.0690 1.8555
I 0.2658 1.4948 2.2808 1.5567 2.0145 1.5068 2.9077 3.3497
J 1.0574 0.2595 2.8543 2.4059 1.5042 0.4216 2.7777 2.4900 1.2446
```

```
dist.quant(xy,3)
```

```
      A      B      C      D      E      F      G      H      I
B 1.2951
C 2.1755 2.9627
D 1.5203 2.5746 0.8526
E 1.7513 1.4332 2.0326 2.0771
F 1.2754 0.2866 2.7422 2.4160 1.1466
G 2.6831 2.7438 1.5782 2.1524 1.3274 2.4590
```

```
H 3.0843 2.2942 3.2036 3.4383 1.4121 2.0690 1.8555
I 0.2658 1.4948 2.2808 1.5567 2.0145 1.5068 2.9077 3.3497
J 1.0574 0.2595 2.8543 2.4059 1.5042 0.4216 2.7777 2.4900 1.2446
```

```
s.label(w1$l1) #C
```



C D

La normalisation des coordonnées déforme fortement le nuage en enlevant de l'importance aux directions principales, le défaut associé est évidemment qu'elle en donnera trop aux dernières qui ne sont souvent que du bruit. **A utiliser avec précaution**. La métrique canonique sur les logarithmes doublement centrés est plus robuste. Mais elle élimine radicalement les variations de taille.

```
xydc=scale(xy,scale=F)
xydc = t(scale(t(xydc),scale=F))
s.label(xydc) #D
```

Exercice : pourquoi cette somme vaut-elle 2 ?

```
cor(xydc,w1$li)
    Axis1   Axis2
x  0.3892 -0.9211
y -0.3892  0.9211
sum(cor(xydc,w1$li)^2)
[1] 2
```

2.3. Distances génétiques

En génétique, on calcule la distance entre deux populations à partir d'un échantillon d'individus. Chaque individu fournit ses allèles pour un certain nombre de loci et la population est plutôt un ensemble d'allèles qu'un ensemble d'individus (hypothèse de Hardy-Weinberg). On peut calculer avec la fonction `dist.genet` [21] les options les plus classiques.

Soit **A** un tableau de fréquences alléliques avec t lignes (populations) et m colonnes (formes allèliques). Soit v le nombre de loci. Le locus j a $m(j)$ formes allèliques.

$$m = \sum_{j=1}^{v} m(j)$$

[21] à partir de la version 1.2 de ade4 dans R.

Pour la $i^{ème}$ ligne et la $k^{ème}$ modalité de la variable j, on note la valeur □ (1≤i≤t, 1≤j≤v, et 1≤k≤$m(j)$), la valeur du tableau des données brutes (en général un effectif d'allèles). Soit :

$$a_{ij}^{+} = \sum_{k=1}^{m(j)} a_{ij}^{k} \text{ et } p_{ij}^{k} = \frac{a_{ij}^{k}}{a_{ij}^{+}}$$

Soit le tableau $\mathbf{P} = \left[p_{ij}^{k} \right]$ (tableau de fréquences alléliques) et les paramètres :

$$p_{ij}^{+} = \sum_{k=1}^{m(j)} p_{ij}^{k} = 1, \quad p_{i+}^{+} = \sum_{j=1}^{v} p_{ij}^{+}, \quad p_{++}^{+} = \sum_{j=1}^{v} p_{i+}^{+} = tv$$

On calcule des matrices de distances entre populations en utilisant les fréquences □.

1 — Distance de Nei **22** (Voir **23**) :

$$D_1(a,b) = -Ln\left(\frac{\sum_{k=1}^{v}\sum_{j=1}^{m(k)} p_{aj}^{k} p_{bj}^{k}}{\sqrt{\sum_{k=1}^{v}\sum_{j=1}^{m(k)} \left(p_{aj}^{k}\right)^2} \sqrt{\sum_{k=1}^{v}\sum_{j=1}^{m(k)} \left(p_{bj}^{k}\right)^2}} \right)$$

2 — Distance angulaire ou de Edwards **24** (Voir **25**) :

$$D_2(a,b) = \sqrt{1 - \frac{1}{v}\sum_{k=1}^{v}\left(\sum_{j=1}^{m(k)} \sqrt{p_{aj}^{k} p_{bj}^{k}}\right)}$$

3 — Coefficient de coancestralité ou distance de Reynolds **26** :

$$D_3(a,b) = \sqrt{\frac{\sum_{k=1}^{v}\sum_{j=1}^{m(k)} \left(p_{aj}^{k} - p_{bj}^{k}\right)^2}{2\sum_{k=1}^{v}\left(1 - \sum_{j=1}^{m(k)} p_{aj}^{k} p_{bj}^{k}\right)}}$$

4 — Distance euclidienne classique ou de Rogers **27** (Voir **28**) :

---

22 Nei, M. (1972) Genetic distances between populations. *American Naturalist* : 106. 283-292.
Nei M. (1978) Estimation of average heterozygosity and genetic distance from a small number of individuals. *Genetics*, 23, 341-369

23 Avise, J. C. 1994. Molecular markers, natural history and evolution. Chapman & Hall, London.

24 Edwards, A.W.F. (1971) Distance between populations on the basis of gene frequencies. Biometrics : 27, 873-881. Cavalli-Sforza L.L. & Edwards A.W.F. (1967) Phylogenetic analysis: models and estimation procedures. Evolution, 32, 550-570

25 Hartl, D.L. & Clark, A.G. (1989) *Principles of population genetics*. Sinauer Associates, Sunderland, Massachussetts. 1-682 (p. 303).

26 Reynolds, J. B., B. S. Weir, and C. C. Cockerham. 1983. Estimation of the coancestry coefficient: basis for a short-term genetic distance. *Genetics* **105**:767-779.

27 Rogers, J.S. (1972) Measures of genetic similarity and genetic distances. *Studies in Genetics*, Univ. Texas Publ. 7213: 145-153.

$$D_4(a,b) = \frac{1}{v}\sum_{k=1}^{v}\sqrt{\frac{1}{2}\sum_{j=1}^{m(k)}\left(p_{aj}^{k} - p_{bj}^{k}\right)^2}$$

5 — Distance génétique absolue ou de Provesti [29] :

$$D_5(a,b) = \frac{1}{2v}\sum_{k=1}^{v}\sum_{j=1}^{m(k)}\left|p_{aj}^{k} - p_{bj}^{k}\right|$$

Les distances génétiques ont (ou n'ont pas) des propriétés très particulières ayant un sens en génétique des populations : on les considère ici comme un exemple d'utilisation intensive de la notion de distances : voir les explications fondamentales dans [30].

```
data(microsatt)
```

microsatt regroupe les fréquences alléliques sur 9 loci microsatellites pour 18 races bovines (taurins et zébus européens et africains) préparées par D. Laloë (ugendla@dga2.jouy.inra.fr)[31].

```
micro.gen <- count2genet(microsatt$tab) #21
d1=dist.genet(micro.gen,1)
d2=dist.genet(micro.gen,2)
d3=dist.genet(micro.gen,3)
d4=dist.genet(micro.gen,4)
d5=dist.genet(micro.gen,5)
pairs(cbind(d1,d2,d3,d4,d5))
```

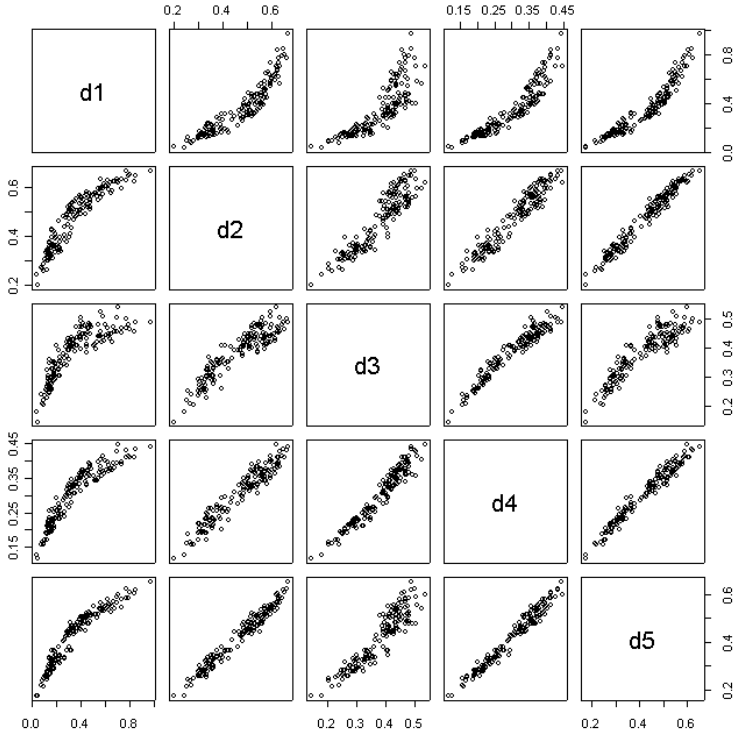
---

[28] Avise, J.C. (1994) *Molecular markers, natural history and evolution*. Chapman & Hall, London. 1-511 (p. 95).

[29] Prevosti A. (1974) La distancia genética entre poblaciones. Miscellanea Alcobé, 68, 109-118. Prevosti A., Ocaña J. & Alonso G. (1975) Distances between populations of Drosophila subobscura, based on chromosome arrangements frequencies. *Theoretical and Applied Genetics*, 45, 231-241

[30] Sanchez-Mazas A. (2003) Cours de Génétique Moléculaire des Populations. Cours VIII Distances génétiques - Représentation des populations.
URL http://anthro.unige.ch/GMDP/Alicia/GMDP_dist.htm

[31] Moazami-Goudarzi, K., D. Laloe, J. P. Furet, and F. Grosclaude. 1997. Analysis of genetic relationships between 10 cattle breeds with 17 microsatellites. Animal Genetics, 28, 338–345.
Souvenir Zafindrajaona, P.,Zeuh V. ,Moazami-Goudarzi K., Laloe D., Bourzat D., Idriss A., and Grosclaude F. (1999) Etude du statut phylogénétique du bovin Kouri du lac Tchad à l'aide de marqueurs moléculaires. Revue d'Elevage et de Médecine Vétérinaire des pays Tropicaux, 55, 155–162.
Moazami-Goudarzi, K., Belemsaga D. M. A., Ceriotti G., Laloe D. , Fagbohoun F., Kouagou N. T., Sidibé I., Codjia V., Crimella M. C., Grosclaude F. and Touré S. M. (2001) Caractérisation de la race bovine Somba à l'aide de marqueurs moléculaires. Revue d'Elevage et de Médecine Vétérinaire des pays Tropicaux, 54, 1–10.



2.4. Distances variées

dist.genet calcule des distances génétiques, **dist.binary** calcule des distances écologiques, **dist.quant** calcule des distances morphométriques, **dist.prop** calcule des distances sur des profils simples, **dist.neig** sur un graphe de voisinage, **dist.dudi** transforme un triplet statistique quelconque en matrices de distances, soit entre lignes, soit entre colonnes.

On utilisera **dist.dudi** pour obtenir, par exemple, la métrique du Chi2 : c'est celle qui est en oeuvre dans l'analyse des correspondances.

```
data(HairEyeColor)
## Aggregate over sex:
x <- apply(HairEyeColor, c(1, 2), sum)
x=as.data.frame(x)
x
```

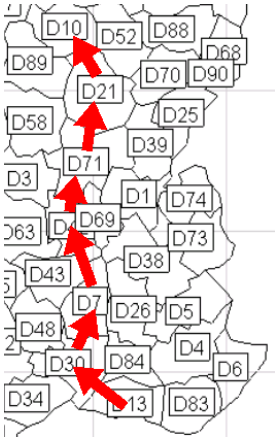
```
      Brown Blue Hazel Green
Black    68   20    15     5
Brown   119   84    54    29
Red      26   17    14    14
Blond     7   94    10    16
```

```
as.matrix(dist.dudi(dudi.coa(x,scannf=FALSE)))
```

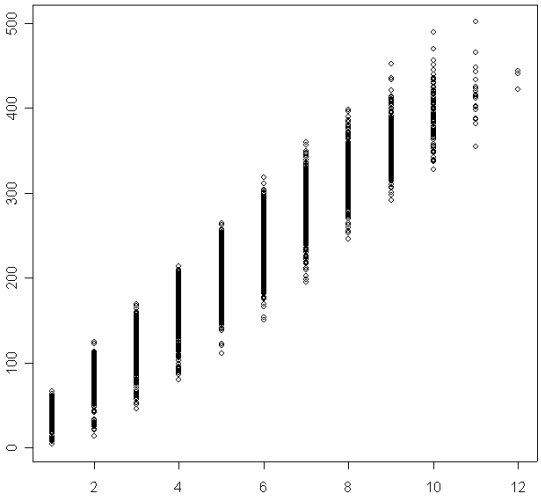
```
       Black  Brown    Red Blond
Black 0.0000 0.4462 0.6535 1.348
Brown 0.4462 0.0000 0.3164 0.991
Red   0.6535 0.3164 0.0000 1.043
Blond 1.3483 0.9910 1.0426 0.000
```

```
a1 = x[1,]/sum(x[1,])
a2 = x[2,]/sum(x[2,])
tot = apply(x,2,sum)
tot = tot/sum(tot)
```

```
a1
       Brown   Blue  Hazel  Green
Black 0.6296 0.1852 0.1389 0.0463
a2
       Brown   Blue  Hazel  Green
Brown 0.4161 0.2937 0.1888 0.1014
tot
 Brown   Blue  Hazel  Green
0.3716 0.3632 0.1571 0.1081
sqrt(sum(((a1-a2)^2)/tot))
[1] 0.4462
```




```
data(elec88)
s.label(elec88$xy,area=elec88$area)
d0 <- dist.neig(elec88$neig)
as.matrix(d0)[10,13]
[1] 6
d1 = dist.quant(elec88$xy,1)
as.matrix(d1)[10,13]
[1] 272.4
plot(d0,d1)
```



En abscisse distances en nombre de frontières traversées, en ordonnée distances à vol d'oiseau. Il existe une multitude de connexion entre distances, voisinages, différences et similarités.

3. Dissimilarités entre parties d'un ensemble

Nous venons de voir quelques unes des nombreuses façons d'obtenir des distances. Si on veut être plus précis, il vaut mieux préciser et parler en général de **dissimilarités**. Si *n* individus forment un ensemble, une matrice de dissimilarités est une matrice carrée *n-n,* de termes positifs ou nuls, symétrique et de diagonale nulle, donc vérifiant *:*

$$1 \le i \le n \Rightarrow d_{ii} = 0$$
$$1 \le i \le n \quad 1 \le j \le n \Rightarrow d_{ij} \ge 0$$
$$1 \le i \le n \quad 1 \le i \le n \Rightarrow d_{ij} = d_{ji}$$

Une dissimilarité est **métrique** (définition 1 dans **1**) si en outre :

$$1 \le i \le n \quad 1 \le j \le n \quad 1 \le k \le n \Rightarrow d_{ij} \le d_{ik} + d_{kj}$$

Une dissimilarité métrique est appelée **distance** p. 59 dans [32] (dont la lecture est vivement recommandée pour en savoir plus).

Une dissimilarité est **euclidienne** (définition 2 dans [1]) si il existe *n* points dans un espace euclidien dont les distances deux à deux sont exactement les dissimilarités considérées. On parle alors de distance euclidienne.

Une dissimilarité est **ultramétrique** si on a :

$$1 \le i \le n \quad 1 \le j \le n \quad 1 \le k \le n \Rightarrow d_{ij} \le \max\left(d_{ik}, d_{jk}\right)$$

La plupart des dissimilarités utilisées sont des distances euclidiennes.

```
data(mafragh)
for(k in 1:10) {
      w=dist.binary(mafragh$flo,k)
      print(is.euclid(w))
}
```

```
[1] TRUE
[1] TRUE
...
[1] TRUE
```

```
for(k in 1:10) {
      w=dist.binary(mafragh$flo,k)
      print(is.euclid(w^2))
}
```

```
[1] FALSE
[1] FALSE
...
[1] FALSE
[1] TRUE
```

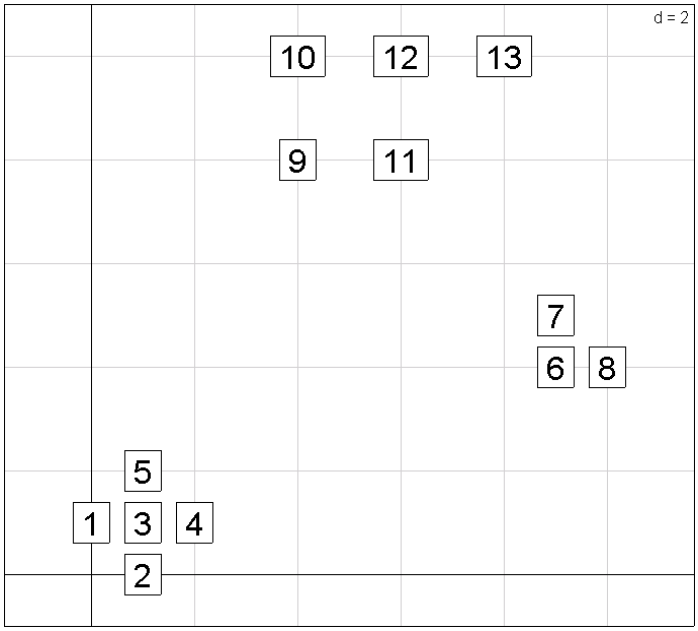
On peut toujours approcher une dissimilarité par une distance euclidienne : voir

http://pbil.univ-lyon1.fr/R/fichestd/tdr67.pdf

Ce qui nous intéresse maintenant, c'est le lien entre distances entre éléments et hiérarchie de parties Une hiérarchie définit d'abord une distance entre individus qui a des propriétés remarquables.

3.1. Ultramétrique entre individus dérivée d'une hiérarchie valuée

32 Lebart, L., A. Morineau, and M. Piron. 1995. Statistique exploratoire multidimensionnelle. Dunod, Paris.



13 points dans le plan

On peut prendre une illustation simple :

```
x = c(0,1,1,2,1,9,9,10,4,4,6,6,8)
y = c(1,0,1,1,2,4,5,4,8,10,8,10,10)
xy=cbind.data.frame(x,y)
s.label(xy)

xy.d=dist(xy)
xy.d
```

```
        1      2      3      4      5      6      7      8      9     10     11     12
2   1.414
3   1.000  1.000
4   2.000  1.414  1.000
5   1.414  2.000  1.000  1.414
6   9.487  8.944  8.544  7.616  8.246
7   9.849  9.434  8.944  8.062  8.544  1.000
8  10.440  9.849  9.487  8.544  9.220  1.000  1.414
9   8.062  8.544  7.616  7.280  6.708  6.403  5.831  7.211
10  9.849 10.440  9.487  9.220  8.544  7.810  7.071  8.485  2.000
11  9.220  9.434  8.602  8.062  7.810  5.000  4.243  5.657  2.000  2.828
12 10.817 11.180 10.296  9.849  9.434  6.708  5.831  7.211  2.828  2.000  2.000
13 12.042 12.207 11.402 10.817 10.630  6.083  5.099  6.325  4.472  4.000  2.828  2.000
```

Matrice de distances entre 13 points

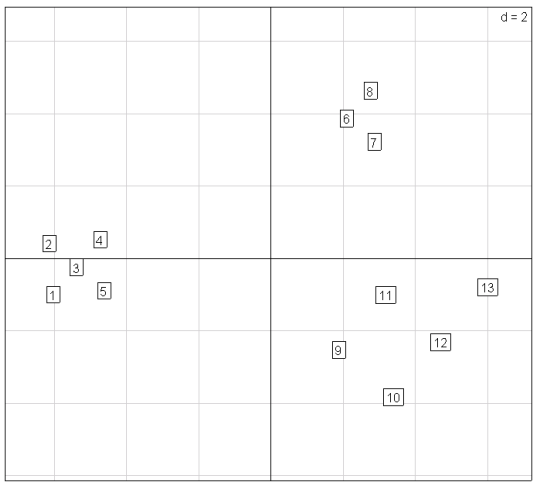
Évidemment cette matrice est euclidienne, puisque les points sont dans un espace euclidien :

```
is.euclid(xy.d)
```

```
[1] TRUE
```

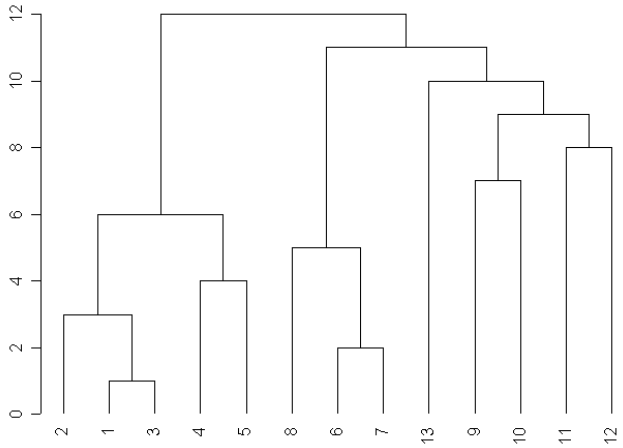
L'analyse en coordonnées principales retouve le plan des données :

```
s.label(dudi.pco(xy.d,scannf=F)$li)
```



13 points dans le plan : une autre représentation

Si on se donne une hiérarchie valuée de parties absolument arbitraire, par exemple :



Hiérarchie valuée : La distance entre 2 et 5 vaut 6

on peut associer à cette hiérarchie valuée une distance entre individu. La distance entre *a* et *b* est le niveau de la plus petite partie qui contient *a* et *b*. La plus petite partie contenant 2 et 5 est :

$$\{2,1,3,4,5\}$$

Cette plus petite partie existe toujours car la partie formée par l'ensemble tout entier est dans la hiérarchie (tout en haut). La distance entre 2 et 5 vaut ainsi 6. Globalement :

	2	1	3	4	5	8	6	7	13	9	10	11	12
2	0	3	3	6	6	12	12	12	12	12	12	12	12
1	3	0	1	6	6	12	12	12	12	12	12	12	12
3	3	1	0	6	6	12	12	12	12	12	12	12	12
4	6	6	6	0	4	12	12	12	12	12	12	12	12
5	6	6	6	4	0	12	12	12	12	12	12	12	12
8	12	12	12	12	12	0	5	5	11	11	11	11	11
6	12	12	12	12	12	5	0	2	11	11	11	11	11
7	12	12	12	12	12	5	2	0	11	11	11	11	11
13	12	12	12	12	12	11	11	11	0	10	10	10	10
9	12	12	12	12	12	11	11	11	10	0	7	9	9
10	12	12	12	12	12	11	11	11	10	7	0	9	9
11	12	12	12	12	12	11	11	11	10	9	9	0	8
12	12	12	12	12	12	11	11	11	10	9	9	8	0

Cette distance est ultramétrique (**32** p. 160). Ceci signifie que pour trois individus arbitraires *x*, *y* et *z* on a toujours :

□

En effet soit A = $x|y$ la plus petite partie contenant x et y, B = $x|z$ la plus petite partie contenant x et z et $C = y|z$ la plus petite partie contenant y et z. B et C on un élément en commun donc l'un des deux est contenu au sens large dans l'autre. Par exemple B est contenu dans C. Donc x, y et z sont tous les trois dans C et A est contenu dans C donc :

$$x \mid y \subseteq y \mid z \Rightarrow d(x,y) \le d(z,y) \Rightarrow d(x,y) \le \max\left(d(x,z), d(y,z)\right)$$

Les ultramétriques ont des propriétés très particulières. Pour trois points arbitraires, on peut toujours affirmer que l'une des trois distances deux à deux est plus petite au sens large que les deux autres (pour trois nombres, il y en a toujours un inférieur ou égal aux deux autres). Alors :

$$\left.\begin{array}{l} d(x,y) \le d(x,z) \\ d(x,y) \le d(y,z) \end{array}\right\} \Rightarrow \left\{\begin{array}{l} d(x,z) \le \max\left(d(x,y), d(y,z)\right) \Rightarrow d(x,z) \le d(y,z) \\ d(y,z) \le \max\left(d(y,x), d(x,z)\right) \Rightarrow d(y,z) \le d(x,z) \end{array}\right\} \Rightarrow d(x,z) = d(y,z)$$

3.2. Hiérarchie valuée dérivée d'une ultramétrique entre individus

On vise à inverser la construction précédente. Il ne s'agit pas de construire une distance entre individus à partir d'une hiérarchie mais au contraire de construire une hiérarchie à partir d'une dissimilarité. Pour obtenir une hiérarchie, il convient de regrouper deux classes à tout moment puisqu'une nouvelle classe est soit disjointe des autres soit emboîtée. La procédure suivante fait cela :

Étape 1 : On dispose d'une matrice de dissimilarités entre n individus qui est une distance ultramétrique. Chaque individu donne une partie réduite à lui-même à laquelle on attribue la valeur 0. La matrice de distance entre individus devient une matrice de distances entre parties. Prendre la plus petite valeur de cette matrice et faire avec le couple correspondant une partie à deux éléments. Attribuer à cette nouvelle partie la valeur égale à cette plus petite valeur. On a alors n-1 parties.

	2	1	3	4	5	8	6	7	13	9	10	11	12
2	0	3	3	6	6	12	12	12	12	12	12	12	12
1	3	0	**1**	6	6	12	12	12	12	12	12	12	12
3	3	**1**	0	6	6	12	12	12	12	12	12	12	12
4	6	6	6	0	4	12	12	12	12	12	12	12	12
5	6	6	6	4	0	12	12	12	12	12	12	12	12
8	12	12	12	12	12	0	5	5	11	11	11	11	11
6	12	12	12	12	12	5	0	2	11	11	11	11	11
7	12	12	12	12	12	5	2	0	11	11	11	11	11
13	12	12	12	12	12	11	11	11	0	10	10	10	10
9	12	12	12	12	12	11	11	11	10	0	7	9	9
10	12	12	12	12	12	11	11	11	10	7	0	9	9
11	12	12	12	12	12	11	11	11	10	9	9	0	8
12	12	12	12	12	12	11	11	11	10	9	9	8	0

$$A = \{1,3\} \rightarrow f(A) = 1$$

Étape 2 : Calculer une nouvelle matrice de dissimilarité en remplaçant le couple regroupé par la partie formée. On remarque que pour tout élément conservé, les distances aux deux éléments groupés sont égales (triangles isocèles à petits côtés sur la base) et donc définissent la distance au groupe :

	2	**A**	4	5	8	6	7	13	9	10	11	12
2	0	3	6	6	12	12	12	12	12	12	12	12
A	3	0	6	6	12	12	12	12	12	12	12	12
4	6	6	0	4	12	12	12	12	12	12	12	12
5	6	6	4	0	12	12	12	12	12	12	12	12
8	12	12	12	12	0	5	5	11	11	11	11	11
6	12	12	12	12	5	0	**2**	11	11	11	11	11
7	12	12	12	12	5	**2**	0	11	11	11	11	11
13	12	12	12	12	11	11	11	0	10	10	10	10
9	12	12	12	12	11	11	11	10	0	7	9	9

```
10 12 12  12 12 11 11 11 10  7  0  9  9
11 12 12  12 12 11 11 11 10  9  9  0  8
12 12 12  12 12 11 11 11 10  9  9  8  0
```

$$B = \{6,7\} \rightarrow f(B) = 2$$

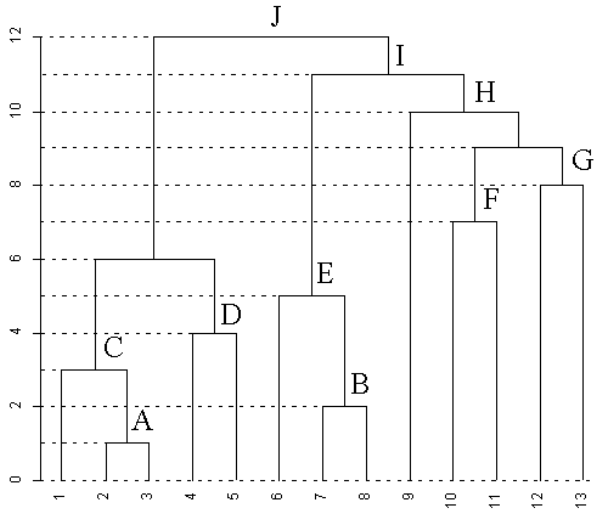
Étape 3 : Recommencer jusqu'à ce qu'il ne reste que la classe regroupant le tout :

```
    2  A  4  5  8  B 13  9 10 11 12
2   0  3  6  6 12 12 12 12 12 12 12
A   3  0  6  6 12 12 12 12 12 12 12
4   6  6  0  4 12 12 12 12 12 12 12
5   6  6  4  0 12 12 12 12 12 12 12
8  12 12 12 12  0  5 11 11 11 11 11
B  12 12 12 12  5  0 11 11 11 11 11
13 12 12 12 12 11 11  0 10 10 10 10
9  12 12 12 12 11 11 10  0  7  9  9
10 12 12 12 12 11 11 10  7  0  9  9
11 12 12 12 12 11 11 10  9  9  0  8
12 12 12 12 12 11 11 10  9  9  8  0
```

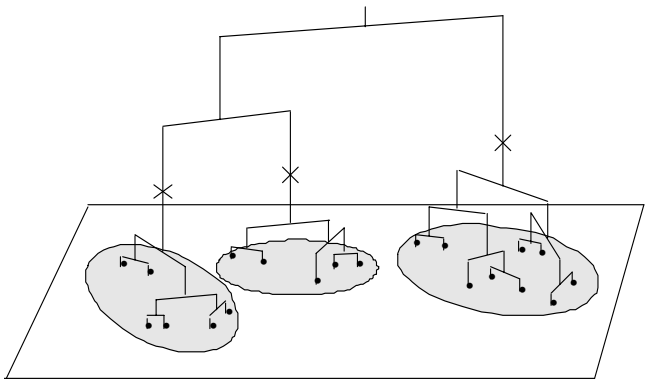
$$C = \{1,2,3\} \rightarrow f(C) = 3$$

…

Une hiérarchie indicée donne une distance ultramétrique, une distance ultramétrique donne une hiérarchie indicée. Le passage de l'un à l'autre est univoque et inversible : il y a équivalence stricte entre l'un et l'autre. La recherche d'une distance ultramétrique, qui est celle d'une hiérarchie valuée, se fait par une méthode de classification hiérarchique.



Le tout est parfaitement résumé dans **32** p. 155 par la figure :



3.3. CAH et distances entre parties

Parmi les méthodes de classification hiérarchique on distingue des ascendantes et les descendantes. Les ascendantes créent une partie en regroupant deux parties existantes. les descendantes divisent au contraire une partie existante pour en faire deux nouvelles. Pour regrouper il faut un critère. Au début, il est naturel de regrouper les deux individus les plus proches au sens de la dissimilarité de départ. Mais immédiatement après cette opération on peut regrouper soit des individus, soit un individu et une classe, soit, un peu plus tard, deux classes. Plusieurs stratégies peuvent alors s'insérer dans le schéma général :

Étape 1 : On dispose d'une matrice de dissimilarités entre *n* individus. Chaque individu donne une partie réduite à lui-même à laquelle on attribue la valeur 0. Prendre la plus petite valeur de cette matrice et faire avec le couple correspondant une partie à deux éléments. Attribuer à cette nouvelle partie une valeur positive. On a alors *n*-1 parties.

Étape 2 : A chaque pas, on a *m* parties et une valeur $h(i)$ associée à chacune d'entre elles. Regrouper deux d'entre elles sur le critère **M** et attribuer à la réunion une valeur *h* supérieure ou égale à la valeur des deux composantes.

Étape 3 : Recommencer jusqu'à ce qu'il ne reste que la classe regroupant le tout et lui attribuer une valeur supérieure à toutes les autres.

Chaque procédé qui définit **M** et *h*, respectivement le choix pour le regroupement et la fonction de valuation donne une CAH particulière. Parmi les plus répandues de ces procédés figurent d'abord ceux qui sont basés sur les distances entre parties.

3.3.1. Lien simple ou *single linkage* ou saut minimum ou *nearest neighbour*

single linkage, le lien simple ou lien du saut minimum, ou celui du plus proche voisin définit la distance entre deux parties par :

$$D(A,B) = \min_{a\in A, b\in B}\left(d(a,b)\right)$$

Dans l'exemple:

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.00	1.41	1.00	2.00	1.41	9.49	9.85	10.44	8.06	9.85	9.22	10.82	12.04
2	1.41	0.00	1.00	1.41	2.00	8.94	9.43	9.85	8.54	10.44	9.43	11.18	12.21
3	1.00	1.00	0.00	1.00	1.00	8.54	8.94	9.49	7.62	9.49	8.60	10.30	11.40
4	2.00	1.41	**1.00**	0.00	1.41	7.62	8.06	8.54	7.28	9.22	8.06	9.85	10.82
5	1.41	2.00	1.00	1.41	0.00	8.25	8.54	9.22	6.71	8.54	7.81	9.43	10.63
6	9.49	8.94	8.54	7.62	8.25	0.00	1.00	1.00	6.40	7.81	5.00	6.71	6.08
7	9.85	9.43	8.94	8.06	8.54	1.00	0.00	1.41	5.83	7.07	4.24	5.83	5.10
8	10.44	9.85	9.49	8.54	9.22	1.00	1.41	0.00	7.21	8.49	5.66	7.21	6.32
9	8.06	8.54	7.62	7.28	6.71	6.40	5.83	7.21	0.00	2.00	2.00	2.83	4.47
10	9.85	10.44	9.49	9.22	8.54	7.81	7.07	8.49	2.00	0.00	2.83	2.00	4.00
11	9.22	9.43	8.60	8.06	7.81	5.00	**4.24**	5.66	2.00	2.83	0.00	2.00	2.83
12	10.82	11.18	10.30	9.85	9.43	6.71	5.83	7.21	2.83	2.00	2.00	0.00	2.00
13	12.04	12.21	11.40	10.82	10.63	6.08	5.10	6.32	4.47	4.00	2.83	2.00	0.00

$$D(\{1,2,3\},\{4,5\}) = 1\; D(\{9\},\{10,11,12,13\}) = 2\; D(\{6,7,8\},\{9,10,11,12,13\}) = d(7,11) = 4.243$$

On réunit alors à chaque pas de la CAH les deux parties présentes les plus proches. Au début la distance entre deux parties ne contenant chacune qu'un élément est égale à la distance entre ces éléments. On agrège donc 1 et 3 (les deux premiers qui conviennent) qui sont distants de 1 et on

forme A = {1,3}. La matrice de distances est mise à jour : des deux lignes regroupés on ne garde que la plus petite des deux valeurs :

```
       2     A     4     5    6    7     8    9    10   11    12    13
2   0.00  1.00  1.41  2.00 8.94 9.43  9.85 8.54 10.44 9.43 11.18 12.21
A   1.00  0.00  1.00  1.00 8.54 8.94  9.49 7.62  9.49 8.60 10.30 11.40
4   1.41  1.00  0.00  1.41 7.62 8.06  8.54 7.28  9.22 8.06  9.85 10.82
5   2.00  1.00  1.41  0.00 8.25 8.54  9.22 6.71  8.54 7.81  9.43 10.63
6   8.94  8.54  7.62  8.25 0.00 1.00  1.00 6.40  7.81 5.00  6.71  6.08
7   9.43  8.94  8.06  8.54 1.00 0.00  1.41 5.83  7.07 4.24  5.83  5.10
8   9.85  9.49  8.54  9.22 1.00 1.41  0.00 7.21  8.49 5.66  7.21  6.32
9   8.54  7.62  7.28  6.71 6.40 5.83  7.21 0.00  2.00 2.00  2.83  4.47
10 10.44  9.49  9.22  8.54 7.81 7.07  8.49 2.00  0.00 2.83  2.00  4.00
11  9.43  8.60  8.06  7.81 5.00 4.24  5.66 2.00  2.83 0.00  2.00  2.83
12 11.18 10.30  9.85  9.43 6.71 5.83  7.21 2.83  2.00 2.00  0.00  2.00
13 12.21 11.40 10.82 10.63 6.08 5.10  6.32 4.47  4.00 2.83  2.00  0.00
```

On recommence en regroupant 2 et *A*. Alors *B* = {1,2,3} :

```
       B     4     5    6    7     8    9    10   11    12    13
B   0.00  1.00  1.00 8.54 8.94  9.49 7.62  9.49 8.60 10.30 11.40
4   1.00  0.00  1.41 7.62 8.06  8.54 7.28  9.22 8.06  9.85 10.82
5   1.00  1.41  0.00 8.25 8.54  9.22 6.71  8.54 7.81  9.43 10.63
6   8.54  7.62  8.25 0.00 1.00  1.00 6.40  7.81 5.00  6.71  6.08
7   8.94  8.06  8.54 1.00 0.00  1.41 5.83  7.07 4.24  5.83  5.10
8   9.49  8.54  9.22 1.00 1.41  0.00 7.21  8.49 5.66  7.21  6.32
9   7.62  7.28  6.71 6.40 5.83  7.21 0.00  2.00 2.00  2.83  4.47
10  9.49  9.22  8.54 7.81 7.07  8.49 2.00  0.00 2.83  2.00  4.00
11  8.60  8.06  7.81 5.00 4.24  5.66 2.00  2.83 0.00  2.00  2.83
12 10.30  9.85  9.43 6.71 5.83  7.21 2.83  2.00 2.00  0.00  2.00
13 11.40 10.82 10.63 6.08 5.10  6.32 4.47  4.00 2.83  2.00  0.00
```

puis *C* = {1,2,3,4} puis *D* = {1,2,3,4,5} et :

```
       D     6    7     8    9    10   11    12    13
D   0.00  7.62 8.06  8.54 6.71  8.54 7.81  9.43 10.63
6   7.62  0.00 1.00  1.00 6.40  7.81 5.00  6.71  6.08
7   8.06  1.00 0.00  1.41 5.83  7.07 4.24  5.83  5.10
8   8.54  1.00 1.41  0.00 7.21  8.49 5.66  7.21  6.32
9   6.71  6.40 5.83  7.21 0.00  2.00 2.00  2.83  4.47
10  8.54  7.81 7.07  8.49 2.00  0.00 2.83  2.00  4.00
11  7.81  5.00 4.24  5.66 2.00  2.83 0.00  2.00  2.83
12  9.43  6.71 5.83  7.21 2.83  2.00 2.00  0.00  2.00
13 10.63  6.08 5.10  6.32 4.47  4.00 2.83  2.00  0.00
```

puis *E* = {6,7}, *F* = {6,7,8} :

```
       D     F     9    10   11    12    13
D   0.00  7.62  6.71  8.54 7.81  9.43 10.63
F   7.62  0.00  5.83  7.07 4.24  5.83  5.10
9   6.71  5.83  0.00  2.00 2.00  2.83  4.47
10  8.54  7.07  2.00  0.00 2.83  2.00  4.00
11  7.81  4.24  2.00  2.83 0.00  2.00  2.83
12  9.43  5.83  2.83  2.00 2.00  0.00  2.00
13 10.63  5.10  4.47  4.00 2.83  2.00  0.00
```

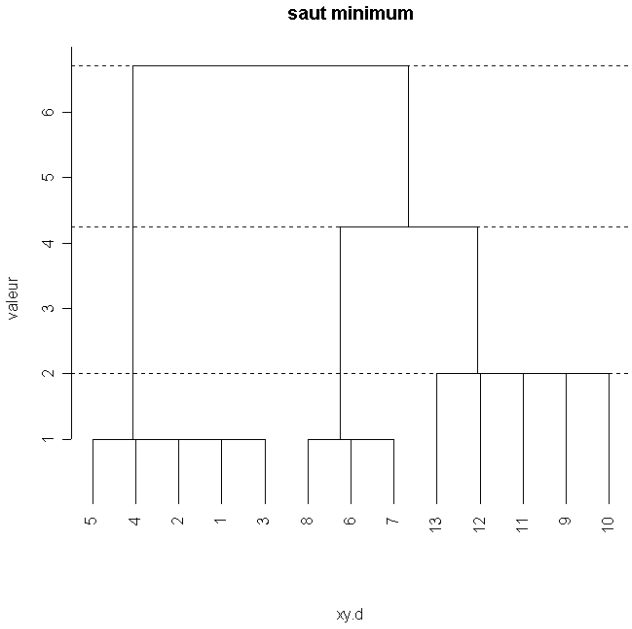
puis *G* = {9,10}, *H* = {9,10,11}, *I* = {9,10,11,12}, *J* = {9,10,11,12,13}

```
       D     F     J
D   0.00  7.62  6.71
F   7.62  0.00  4.24
J   6.71  4.24  0.00
```

enfin *K* = {6,7,8,9,10,11,12,13}

```
       D     K
D   0.00  6.71
K   7.62  0.00
```

et *L* = {1,2,3,4,5,6,7,8,9,10,11,12,13}



Pendant toute la procédure, il se passe en permanence le même enchaînement : on prend la plus petite valeur de la distance encore disponible. Cette valeur concerne les parties X et Y qui sont réunies. Toutes les autres sont conservées et :

$$D(X \cup Y, W) = \min\left(D(X,W), D(Y,W)\right) \geq D(X,Y)$$

La distance entre les deux parties réunies augmente à chaque étape et sert de fonction de valuation :

```
plot(hclust(xy.d,"single"),hang=-1,ylab="valeur",main="saut minimum",sub="")
abline(h=c(2.000, 4.243, 6.7085),lty=2) # ci-dessus
```

3.3.2. Lien complet ou *complete linkage* ou agrégation par le diamètre

La CAH à lien complet (*complete linkage*) ou lien d'agrégation par le diamètre ou *furthest neighbour* est exactement la même procédure que la précédente pour une nouvelle distance entre parties dérivée de la distance entre individus. La distance entre deux parties est définie par :

$$D(A,B) = \max_{a \in A, b \in B}\left(d(a,b)\right)$$

Dans l'exemple:

$$D\left(\{1,2,3\},\{4,5\}\right) = 2 \; D\left(\{9\},\{10,11,12,13\}\right) = 4.472 \; D\left(\{6,7,8\},\{9,10,11,12,13\}\right) = d(8,10) = 8.485$$

La fonction hclust utilise la méthode "**single**" pour le lien simple et "**complete**" pour le lien complet. Pour la même distance on a :

```
h1 <- hclust(xy.d,"single")
h1
Call:
hclust(d = xy.d, method = "single")

Cluster method   : single
Distance         : euclidean
Number of objects: 13
```

Pour savoir ce qu'il y a dans l'objet :

```
unclass(h1)
$merge
      [,1] [,2]
 [1,]   -1   -3 # P1 = {1,3}
 [2,]   -2    1 # P2 = {1,3,2}
 [3,]   -4    2 # P3 = {1,3,2,4}
 [4,]   -5    3 # P4 = {1,3,2,4,5}
 [5,]   -6   -7 # P5 = {6,7}
 [6,]   -8    5 # P6 = {6,7,8}
 [7,]   -9  -10 # P7 = {9,10}
 [8,]  -11    7 # P8 = {9,10,11}
 [9,]  -12    8 # P9 = {9,10,11,12}
[10,]  -13    9 # P10 = {9,10,11,12,13}
[11,]    6   10 # P11 = {6,7,8,9,10,11,12,13}
[12,]    4   11 # P12 = {1,3,2,4,5,6,7,8,9,10,11,12,13}
```

La composante **merge** décrit la hiérarchie de parties. -1, -2, ... désigne les points de départs, 1, 2, ... désigne les parties constituées par la réunion des deux parties en face. On a réécrit les parties P1, P2, ... pour illustrer.

La composante **height** donne les valeurs de la hiérarchie :

```
$height
 [1] 1.000 1.000 1.000 1.000 1.000 1.000 2.000 2.000 2.000 2.000 4.243 6.708

...
```

```
h2 <- hclust(xy.d,"complete")
unclass(h2)
$merge
      [,1] [,2]
 [1,]   -1   -3
 [2,]   -6   -7
 [3,]   -2    1
 [4,]   -4   -5
 [5,]   -8    2
 [6,]    3    4
 [7,]   -9  -10
 [8,]  -11  -12
 [9,]    7    8
[10,]  -13    9
[11,]    5   10
[12,]    6   11
$height
 [1]  1.000  1.000  1.414  1.414  1.414  2.000  2.000  2.000  2.828  4.472  8.485 12.207
```

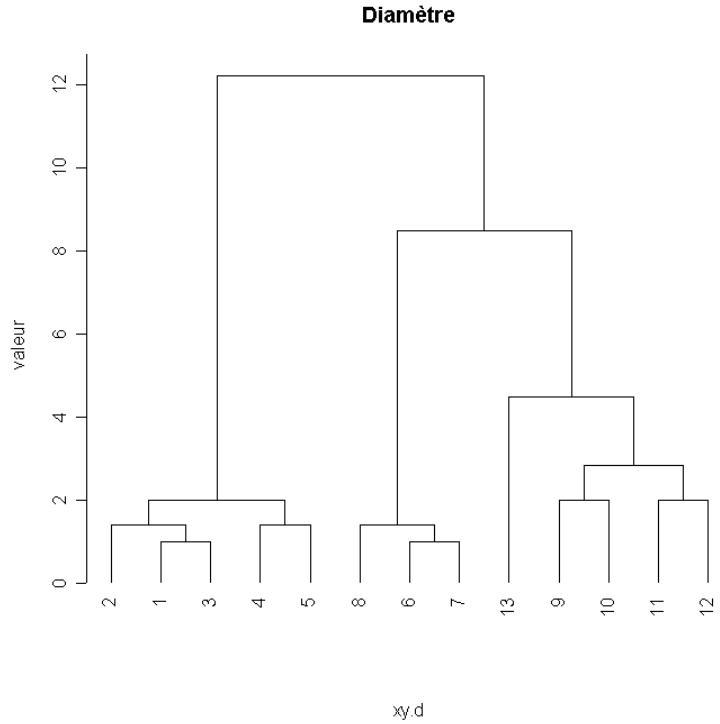
On peut vérifier la règle d'attribution de la valeur. La partie 9 regroupe les parties 7 et 8. 7 est formé de 9 et 10, 8 est formé de 11 et 12.

$$D\left(\{9,10\},\{11,12\}\right)=\max\left(d\left(9,11\right),d\left(10,12\right),d\left(10,11\right),d\left(10,12\right)\right)=\max(2,2.828)=2.828$$

```
       1     2     3     4     5    6    7     8    9    10   11    12    13
1   0.00  1.41  1.00  2.00  1.41 9.49 9.85 10.44 8.06  9.85 9.22 10.82 12.04
...
8  10.44  9.85  9.49  8.54  9.22 1.00 1.41  0.00 7.21  8.49 5.66  7.21  6.32
9   8.06  8.54  7.62  7.28  6.71 6.40 5.83  7.21 0.00  2.00 2.00  2.83  4.47
10  9.85 10.44  9.49  9.22  8.54 7.81 7.07  8.49 2.00  0.00 2.83  2.00  4.00
11  9.22  9.43  8.60  8.06  7.81 5.00 4.24  5.66 2.00  2.83 0.00  2.00  2.83
...

$order
 [1]  2  1  3  4  5  8  6  7 13  9 10 11 12
```

Cette composante donne l'ordre de placement des points pour le dessin de la hiérarchie (dendrogramme)

```
plot(h2,hang=-1,ylab="valeur",main="Diamètre",sub="")
```




```
$labels
 [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13"

$method
[1] "complete"

$call
hclust(d = xy.d, method = "complete")

$dist.method
[1] "euclidean"
```

3.3.3. Lien moyen ou *average linkage* ou UGPMA ou *group average*

average linkage, le lien moyen ou lien d'agrégation UGPMA (*Unweighted Pair Group Method of Agregation)* définit la distance entre deux parties par :

$$D(A,B) = \underset{a\in A, b\in B}{\text{moyenne}}\left(d(a,b)\right)$$

Dans l'exemple:

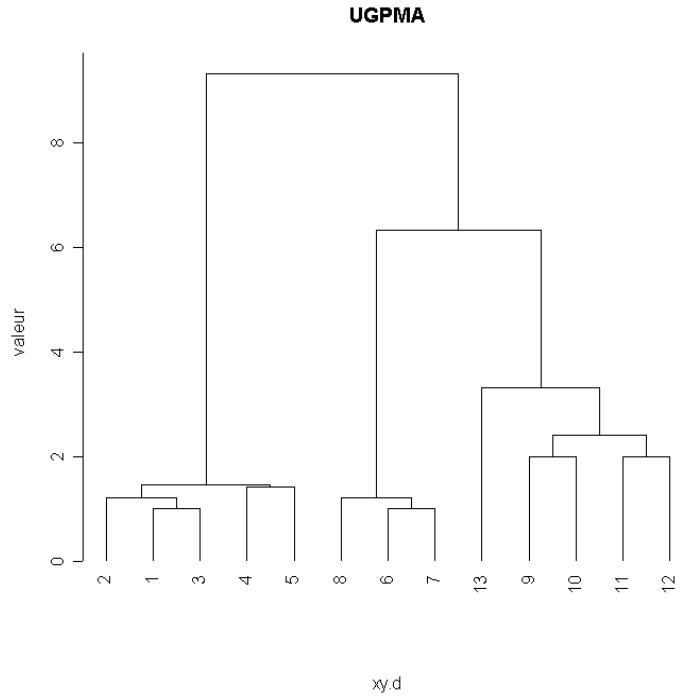
$$D\left(\{1,2,3\},\{4,5\}\right) = \frac{1}{6}\left(d_{14}+d_{15}+d_{24}+d_{25}+d_{34}+d_{35}\right) = \frac{1}{6}\left(2+\sqrt{2}+\sqrt{2}+2+1+1\right) = 1.4714$$

La CAH associée utilise cette distance en conservant l'ensemble des autres caractéristiques.

```
h3 <- hclust(xy.d,"average")
h3$height
 [1] 1.000 1.000 1.207 1.207 1.414 1.471 2.000 2.000 2.414 3.325 6.331 9.319
h3$merge
      [,1] [,2]
 [1,]   -1   -3 # P1 = {1,3}
 [2,]   -6   -7
 [3,]   -2    1 # P3 = {1,2,3}
 [4,]   -8    2
 [5,]   -4   -5 # P5 = {4,5}
 [6,]    3    5 # P6 regroupe P3 et P5
 [7,]   -9  -10
 [8,]  -11  -12
```

```
 [9,]    7    8
[10,]  -13    9
[11,]    4   10
[12,]    6   11
```

```
plot(h3,hang=-1,ylab="valeur",main="UGPMA",sub="")
```

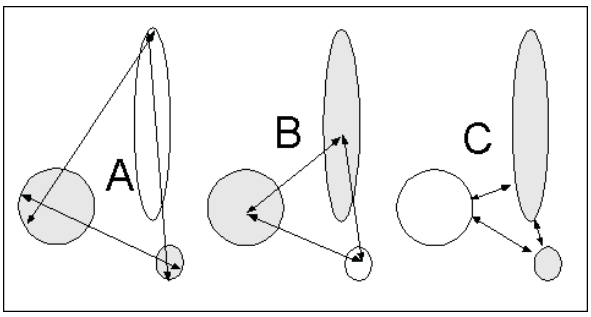


La mise à jour de la distance entre partie après un regroupement se fait ici simplement par :

$$D(A \cup B, X) = \underset{w \in A \cup B, x \in X}{\text{moyenne}}\left(d(w,x)\right) = \frac{n_A D(A,X) + n_B D(B,X)}{n_A + n_B}$$

3.4. CAH et inertie intra-classe

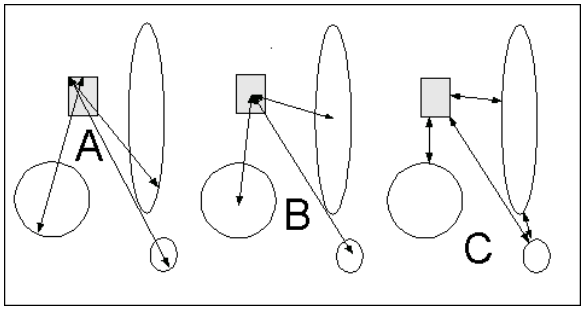
L'introduction d'une distance entre parties et la valuation d'un regroupement par la distance entre les deux parties regroupées permet d'introduire un critère **M** et une fonction *f* dans l'algorithme général.



CAH sur distances entre parties. A lien du diamètre. B lien moyen. C lien du saut minimum. La distance minimum est la valeur du groupement.

Ce n'est pas l'unique possibilité. Ce qui caractérise les trois méthodes de base est que la distance entre deux parties est clairement connue avant toute opération dès qu'est connue la distance entre

éléments de départ. Rajouter à la figure un point supplémentaire ou une partie supplémentaire ne changera rien à ces distances.



En fait la CAH n'utilise qu'une petite partie de la distance induite dans l'ensemble des parties : celle qui est nécessaire pour faire le choix du groupement à chaque étape. On n'a pas besoin de toutes les distances entre parties mais d'une toute petite fraction. Pour 10 éléments on a 1024 parties donc 524288 distances mais on n'utilise que 45+9+8+7+...+1 =90 de ces valeurs. Dans la CAH, il n'est pas nécessaire d'avoir une distance entre les parties mais un procédé d'actualisation d'une distance entre parties utiles. Et comme on garde l'essentiel à l'exception du groupement on a besoin d'un procédé du type :

$$\big(D(A,X),D(B,X)\big)\mapsto D(A\cup B,X)$$

Ceci fonctionnera tant que :

$$D(A\cup B,X)\geq D(A,B)$$

Il suffira alors, vu qu'on a regroupé les plus proches, que :

$$D(A\cup B,X)\geq D(A,X)$$
$$D(A\cup B,X)\geq D(B,X)$$

La méthode de Ward, ou du moment d'ordre 2, utilise ce principe sans être obligé d'expliciter à priori une distance entre parties. J.H. Ward est cité comme ayant écrit un des premiers articles[33] sur la CAH mais Cormack (1971, op. cit. Table 2) renvoie à [34] qui cite pourtant la méthode de Ward en introduction. Ce n'est pas un problème mineur parce qu'il y a d'entrée une ambiguïté fondamentale. On retrouve souvent cette difficulté qui consiste à introduire, avec Ward, la variance intra comme fonction de valeur d'une partition, cette variance intra étant définie sur un nuage de points dans un espace euclidien, en particulier à partir d'un tableau d'ACP normée. Or, jusqu'à présent nous avons séparé ce qui touche au calcul de la distance et ce qu'on en fait.

Évidemment, en partant d'un tableau (en particulier d'ACP normée ou de coordonnées factorielles), on peut calculer une distance (en particulier canonique), puis en partant d'une distance on peut calculer une CAH. Mais en partant d'un tableau on peut aussi calculer des centres de gravité par classe, donc des variances intra-classes comme dans **34** :

$$E_t=\sum_{i=1}^{k_t}\sum_{j=1}^{m}\left(X_{ijt}-\tilde{X}_{jt}\right)^2$$

33 Ward, J. H. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association* 58:238-244.

34 Wishart, D. 1969. An algorithm for hierarchical classification. *Biometrics* 25:165-170.

où on reconnaît la valeur de la variable j sur le $i^{\text{ème}}$ point de la classe t et l'écart à la moyenne correspondante. La somme des carrés des écarts à la moyenne de la classe est une mesure de l'hétérogénéité de cette classe.

On a exactement la même notion avec :

$$E_t = \sum_{i=1}^{k_t} \left\| \mathbf{X}_{it} - \tilde{\mathbf{X}}_t \right\|^2 = \frac{1}{2k_t} \sum_{i=1}^{k_t} \sum_{m=1}^{k_t} \left\| \mathbf{X}_{it} - \mathbf{X}_{mt} \right\|^2$$

```
w=rnorm(20)
w
 [1] -0.2442 -1.7657 -0.7850 -0.4544 -0.7872 -0.3182  0.2401 -0.5747 -0.7229 -1.3428
[11]  0.7168  0.6291 -0.3274 -0.9868  1.6626  0.2406 -1.0464 -1.1690  0.3930 -0.1003
sum((w-mean(w))^2)
[1] 12.6
var(w)*19 # var utilise (n-1)
[1] 12.6
w.d = dist(w)
sum(w.d^2)/20 # on compte une fois chaque paire
[1] 12.6
```

La variance n'est pas une propriété propre aux nuages de points : la variance est aussi la somme des carrés des écarts deux à deux entre points et existe hors d'une distance euclidienne. On peut alors continuer à ne raisonner que sur les distances ce qui soulève un nouveau problème. Entre deux lignes du tableau la distance est la racine de la somme des carrés des écarts et la variance intra-classe est la somme des carrés des distances entre points. Mais le carré de la distance est aussi une dissimilarité et on a une autre variance intraclasse avec la somme des distances (et non des carrés) deux à deux.

En n'utilisant qu'une distance d on dira que l'inertie de la classe A est (entre barres le nombre d'éléments de A) :

$$I(A) = \frac{1}{|A|} \sum_{x \in A, y \in A} d(x,y)$$

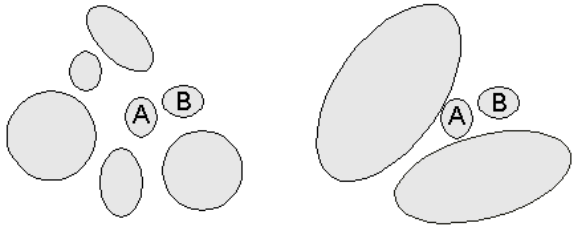
On retrouve la notion ordinaire, dans un espace euclidien, avec :

$$d(x,y) = \frac{1}{2} \|x - y\|^2$$

Pour une partition formée de K classes disjointes on a alors une inertie intraclasse :

$$I(A_1, A_2, ..., A_K) = \sum_{k=1}^{K} \frac{1}{|A_k|} \sum_{x \in A_k, y \in A_k} d(x,y)$$

Quand chaque élément est dans une classe, l'inertie intra-classe est nulle. Cette quantité est vue par Wishart comme une erreur commise en remplaçant les données par la moyenne des valeurs d'une classe mais est étendue à toute distance (qu'on puisse ou non faire des moyennes par classe). Elle mesure la variabilité totale dans (***within***) les classes. Si on regroupe deux individus a et b, elle augmente de $d(a,b)$. Au début la matrice de distance peut être considérée comme la matrice donnant pour chaque couple de point la quantité qui caractérise **l'augmentation** de l'inertie intraclasse dans le regroupement. Cette quantité ne dépend pas de l'état du regroupement :



Le regroupement de A et B provoquera dans les deux cas la même **variation** d'inertie intraclasse, à savoir :

$$\Delta(A,B)=\frac{1}{|A\cup B|}\sum_{x,y\in A\cup B}d(x,y)-\frac{1}{|A|}\sum_{x,y\in A}d(x,y)-\frac{1}{|B|}\sum_{x,y\in B}d(x,y)$$

L'essentiel est alors de savoir comment cette quantité est actualisée dans le groupement, c'est-à-dire quelle fonction se trouve dans le passage :

$$\Delta(A,X),\Delta(B,X)\mapsto\Delta(A\cup B,X)$$

On note, pour simplifier :

$$b(X,Y)=\sum_{x\in X,y\in Y}d(x,y)\quad\text{et}\quad w(X)=\frac{1}{|X|}b(X,X)$$

Pour deux parties disjointes - les calculs ne concernent que ce cas par construction, le nombre d'éléments de la réunion est toujours égal à la somme des nombres d'éléments des composantes - le regroupement induit une variation d'inertie intra qui vaut par définition :

$$\Delta(X,Y)=w(X\cup Y)-w(X)-w(Y)$$

On en déduit par décomposition de la somme :

$$\sum_{x\in X\cup Y,y\in X\cup Y}d(x,y)=\sum_{x\in X,y\in X}d(x,y)+\sum_{x\in Y,y\in Y}d(x,y)+2\sum_{x\in X,y\in Y}d(x,y)$$

que :

$$\left(|X|+|Y|\right)\Delta(X,Y)=2b(X,Y)-\frac{|Y|}{|X|}b(X,X)-\frac{|X|}{|Y|}b(Y,Y)$$

Alors, appliquée deux fois, la relation donne :

$$\left(|A|+|X|\right)\Delta(A,X)+\left(|B|+|X|\right)\Delta(B,X)=$$
$$2b(A,X)-\frac{|X|}{|A|}b(A,A)-\frac{|A|}{|X|}b(X,X)+2b(B,X)-\frac{|X|}{|B|}b(B,B)-\frac{|B|}{|X|}b(X,X)$$

$$(|A|+|X|)\Delta(A,X)+(|B|+|X|)\Delta(B,X)=$$
$$2b(A\cup B,X)-\frac{|A\cup B|}{|X|}b(X,X)-\frac{|X|}{|A|}b(A,A)-\frac{|X|}{|B|}b(B,B)=$$
$$(|A|+|B|+|X|)\Delta(A\cup B,X)+$$
$$\frac{|X|}{|A\cup B|}b(A\cup B,A\cup B)-\frac{|X|}{|A|}b(A,A)-\frac{|X|}{|B|}b(B,B)=$$
$$(|A|+|B|+|X|)\Delta(A\cup B,X)+\Delta(A,B)$$

Il reste :

$$\Delta(A\cup B,X)=\frac{(|A|+|X|)\Delta(A,X)}{(|A|+|B|+|X|)}+\frac{(|B|+|X|)\Delta(B,X)}{(|A|+|B|+|X|)}-\frac{|X|\Delta(A,B)}{(|A|+|B|+|X|)}$$

On retrouve ce résultat dans **hclust**.

```
xy.d
```

```
        1      2      3      4      5      6      7      8      9     10     11     12
2   1.414
3   1.000  1.000
4   2.000  1.414  1.000
5   1.414  2.000  1.000  1.414
6   9.487  8.944  8.544  7.616  8.246
7   9.849  9.434  8.944  8.062  8.544  1.000
8  10.440  9.849  9.487  8.544  9.220  1.000  1.414
9   8.062  8.544  7.616  7.280  6.708  6.403  5.831  7.211
10  9.849 10.440  9.487  9.220  8.544  7.810  7.071  8.485  2.000
11  9.220  9.434  8.602  8.062  7.810  5.000  4.243  5.657  2.000  2.828
12 10.817 11.180 10.296  9.849  9.434  6.708  5.831  7.211  2.828  2.000  2.000
13 12.042 12.207 11.402 10.817 10.630  6.083  5.099  6.325  4.472  4.000  2.828  2.000
```

```
unclass(hclust(xy.d,"ward"))
```

```
$merge
      [,1] [,2]
 [1,]   -1   -3 # d(1,3) = 1
 [2,]   -6   -7 # d(6,7) = 1
 [3,]   -2    1 # agrégation de 2 à 1-3 : 2*1.414/3 + 2/3 - 1/3 =  1.276
 [4,]   -8    2 # agrégation de 8 à 6-7 : 2/3 + 2*1.414/3 - 1/3 =  1.276
 [5,]   -4   -5 # d(4,5) = 1.414
 [6,]    3    5 # voir A ci-dessous
 [7,]   -9  -10 # d(9,10) = 2.000
 [8,]  -11  -12 # d(11,12) = 2.000
 [9,]  -13    8 # agrégation de 13 à 11-12 : 2*2.828/3 + 2*2/3 - 2/3 = 2.552
[10,]    7    9
[11,]    4   10
[12,]    6   11 # voir B ci-dessous
$height
 [1]  1.000  1.000  1.276  1.276  1.414 1.772  2.000  2.000  2.552  4.231 18.276 41.934
```

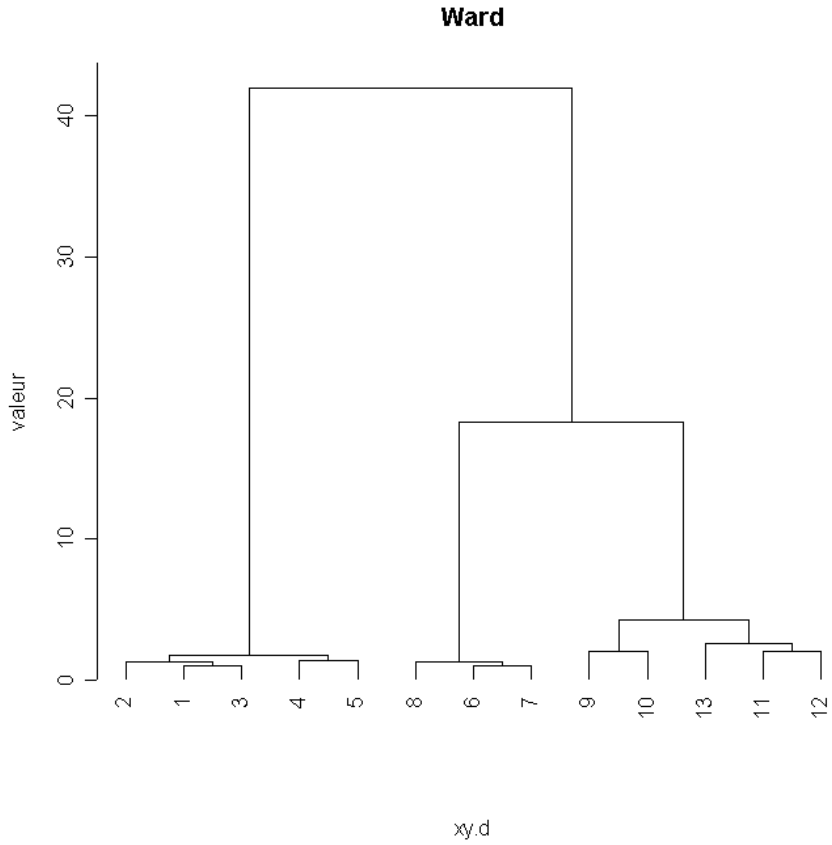
Commentaire A : P3 est une partie à 3 éléments {1,2,3}. P5 est une partie à deux éléments {4,5}. La réunion de P3 et P5 augmente l'inertie intraclasse de :

```
x1=xy[1:3,]
x2=xy[4:5,]
x3=xy[1:5,]
2*(sum(dist(x3))/5- sum(dist(x2))/2- sum(dist(x1))/3)
```

```
[1] 1.772
```

N.B. Ce résultat est obtenu par la mise à jour de la matrice de distance : on vérifie ici le contenu.

```
plot(hclust(xy.d,"ward"),hang=-1,ylab="valeur",main="Ward",sub="")
```

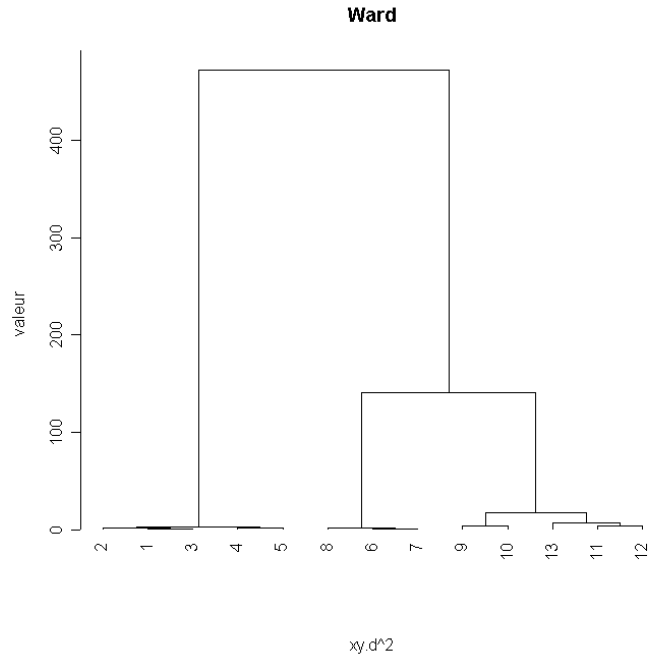


Commentaire B :

```
x1=xy[1:5,]
x2=xy[6:13,]
2*(sum(xy.d)/13- sum(dist(x1))/5- sum(dist(x2))/8)
[1] 41.934
```

En fait, pour être complètement cohérent entre la procédure et la partie euclidienne de la théorie, il faudrait faire :

```
plot(hclust(xy.d^2,"ward"),hang=-1,ylab="valeur",main="Ward",sub="")
```



3.5. Stratégies de CAH

Il existe encore d'autres options dans **`hclust`** :

```
"ward","single","complete","average","mcquitty","median","centroid".
```

On connaît maintenant les quatre premières. Celle dite de Ward définit d'abord la valeur d'un regroupement (l'augmentation de l'inertie intra) qui sert de mesure de dissimilarités entre parties (deux parties dont le regroupement augmente peu l'inertie intra sont proches), alors que dans les liens sur distances, la distance entre individus définit une distance entre parties qui définit une valeur (la valeur du groupement est la distance entre les deux parties regroupées). Dans tous les cas, il s'agit d'algorithme et donc d'économie de moyen. Il faut faire *n* - 1 fois la même chose, en particulier, dès qu'on a décidé de regrouper les parties *A* et *B* de recalculer le critère qui sera utilisé au tour suivant.

1 - ward

$$D(A\cup B,X)=\frac{(n_A+n_X)D(A,X)}{(n_A+n_B+n_X)}+\frac{(n_B+n_X)D(B,X)}{(n_A+n_B+n_X)}-\frac{n_X D(A,B)}{(n_A+n_B+n_X)}$$

2 - single

$$D(A\cup B,X)=\min\left(D(A,X),D(B,X)\right)=\frac{1}{2}D(A,X)+\frac{1}{2}D(B,X)-\frac{1}{2}\left|D(A,X)-D(B,X)\right|$$

3 - complete

$$D(A\cup B,X)=\max\left(D(A,X),D(B,X)\right)=\frac{1}{2}D(A,X)+\frac{1}{2}D(B,X)+\frac{1}{2}\left|D(A,X)-D(B,X)\right|$$

4- average

$$D(A\cup B,X)=\frac{n_A D(A,X)+n_B D(B,X)}{n_A+n_B}$$

5- mcquitty

$$D(A\cup B,X)=\frac{D(A,X)+D(B,X)}{2}$$

6- median

$$D(A\cup B,X)=\frac{1}{2}D(A,X)+\frac{1}{2}D(B,X)-\frac{1}{4}D(A,B)$$

7- centroid

$$D(A\cup B,X)=\frac{n_A D(A,X)}{(n_A+n_B)}+\frac{n_B D(B,X)}{(n_A+n_B)}-\frac{n_A n_B D(A,B)}{(n_A+n_B)^2}$$

La fonction **hclust** reproduit pour l'essentiel le tableau des *Sorting strategies* de Cormack (1971, op. cit. p.331) qui contient la bibliographie fondamentale. Ces éléments permettent de s'en servir.

4. Utilisation des hiérarchies

Il faut d'abord savoir qu'il s'agit d'outils descriptifs qui n'ont pas de propriétés d'optimalité. En effet, on pourrait aussi envisager de comparer toutes les hiérarchies possibles, et choisir celle qui optimise le critère choisi. Malheureusement, le nombre total de hiérarchies possibles est beaucoup trop grand, même pour un petit nombre d'objets. Pour *n* objets, le nombre de hiérarchies possibles est :

$$\frac{(2n-3)!}{2^{n-2}(n-2)!}$$

```
nhiera <- function(n) {
        a = lgamma(2*n-2)
        a = a - (n-2)* log(2)
        a = a- lgamma(n-1)
        return(exp(a))
}
```

```
nhiera(3)
[1] 3

nhiera(10)
[1] 34459425

nhiera(20)
[1] 8.2e+21
```

L'exploration impossible et on est donc conduit à utiliser des heuristiques, c'est-à-dire des algorithmes empiriques dont les propriétés d'optimalité sont relatives. Ceci a des conséquences importantes : d'un part, on n'est pas sûr que la hiérarchie obtenue est celle qui optimise le critère, et d'autre part, les algorithmes utilisés sont souvent coûteux en temps de calcul et/ou en place mémoire.

Contrairement aux méthodes d'analyse de données basées sur l'algèbre linéaire (méthodes d'ordination), les méthodes de classification se caractérisent donc par le fait qu'il n'existe pas de

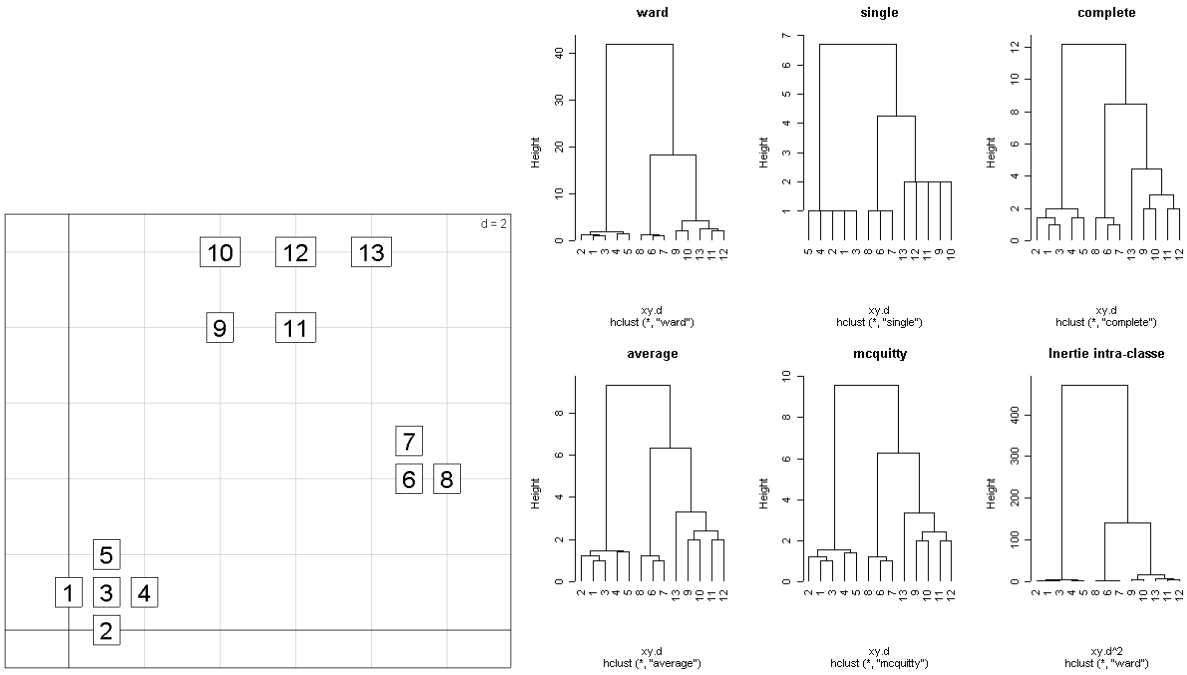
méthode canonique, mais au contraire un grand nombre d'algorithmes empiriques. De même, on n'a en général pas unicité des solutions obtenues : le plan factoriel F1xF2 de l'ACP d'un tableau est unique, et indépendant des algorithmes de calcul des éléments propres, ce qui n'est pas le cas d'un dendrogramme.

La première difficulté vient du fait que les trois méthodes de CAH basées sur des distances induites entre parties donnent des arbres qui peuvent avoir des formes très différentes : la méthode du saut minimum conduit en général à des arbres très aplatis, avec accrochages successifs des objets un à un, ce qui conduit à la formation de chaînes. La méthode du diamètre a au contraire tendance à former des arbres très éclatés, avec formation de groupes isolés. Dans le cas général, il vaut donc mieux utiliser la méthode de la distance moyenne (UPGMA). La seconde difficulté vient du fait que ces méthodes acceptent en entrée aussi bien la distance que son carré, son cube, sa racine, ... ce qui fait jouer aux grandes distances des rôles plus ou moins déterminants. Les exemples qui suivent donneront des repères.

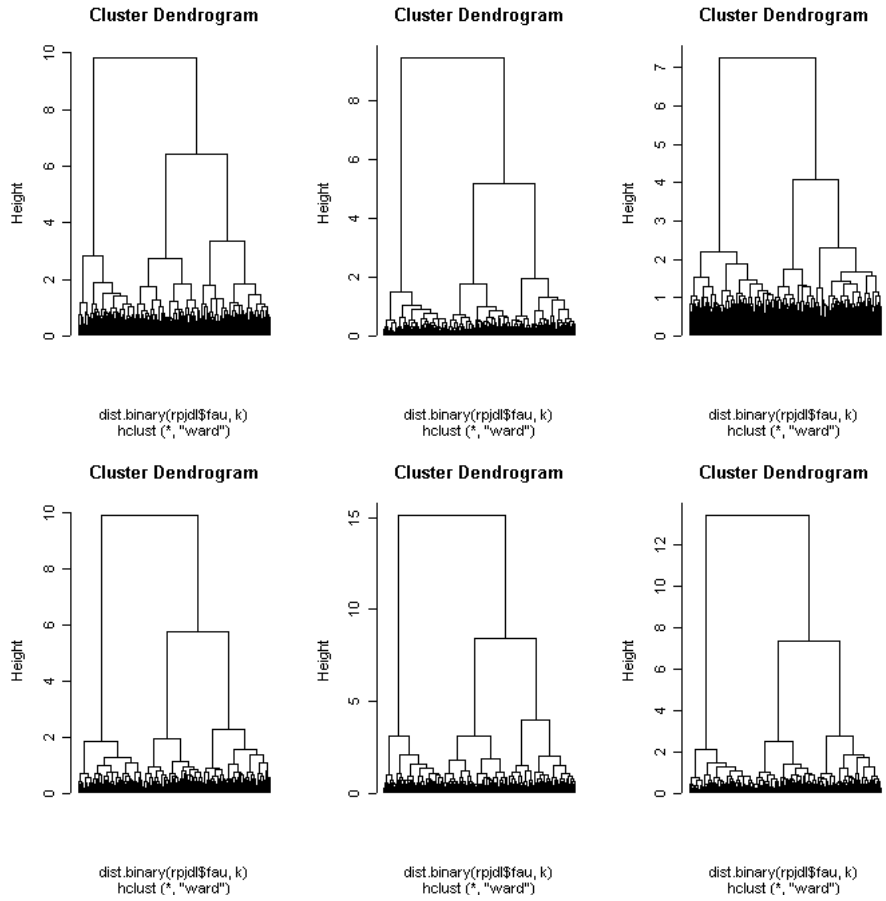
4.1. Couper l'arbre

Une troncature de l'arbre à un niveau donné fournit une partition de l'ensemble. C'est une des fonction de base de la CAH que d'indiquer globalement la présence de *clusters*. Certainement, si à un certain niveau l'inertie intra augmente brutalement et fortement, on peut considérer que la vision de sous-ensembles dans les données est pertinente.

```
par(mfrow=c(2,3))
possible <-   c("ward", "single", "complete", "average", "mcquitty","median",
"centroid")
for(k in 1:5) plot(hclust(xy.d,possible[k]),hang=-1,main=possible[k])
plot(hclust(xy.d^2,"ward"),hang=-1,main="Inertie intra-classe")
```

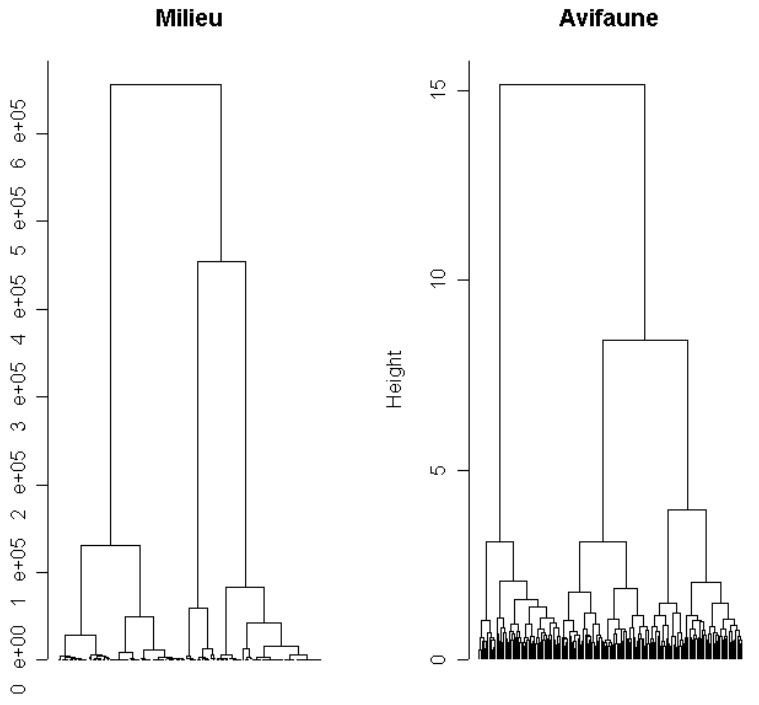



```
data(rpjdl)
par(mfrow=c(2,3))
for     (k     in     1:6)     plot(hclust(dist.binary(rpjdl$fau,k),"ward"),hang=-
1,lab=rep("",nrow(rpjdl$fau)))
```



La distance importe peu : les relevés faunistiques semblent pouvoir être classés.

```
par(mfrow=c(1,2))
plot(hclust(dist(rpjdl$mil)^2,"ward"),hang=-
1,lab=rep("",nrow(rpjdl$fau)),main="Milieu")
plot(hclust(dist.binary(rpjdl$fau,5),"ward"),hang=-
1,lab=rep("",nrow(rpjdl$fau)),main="Avifaune")
```



On peut être curieux de voir le lien existant entre les 3 classes qui se dessinent dans chaque dendrogramme :

```
fmil = as.factor(cutree(hclust(dist(rpjdl$mil)^2,"ward"),3))
favi = as.factor(cutree(hclust(dist.binary(rpjdl$fau,5),"ward"),3))
table(fmil,favi)
    favi
fmil 1  2  3
   1 54 18  1
   2  3 44 41
   3  0  2 19
```

Il est peu probable que ce soit un hasard !

```
fmil = as.factor(cutree(hclust(dist(rpjdl$mil)^2,"ward"),6))
favi = as.factor(cutree(hclust(dist.binary(rpjdl$fau,5),"ward"),6))
table(fmil,favi)
    favi
fmil 1  2  3  4  5  6
   1 40  9  6  0  0  0
   2  5  4  3  5  0  1
   3  0 11  3 30  4  4
   4  0  0  0  3 11 22
   5  0  0  0  2 11  0
   6  0  0  0  0  3  5
```

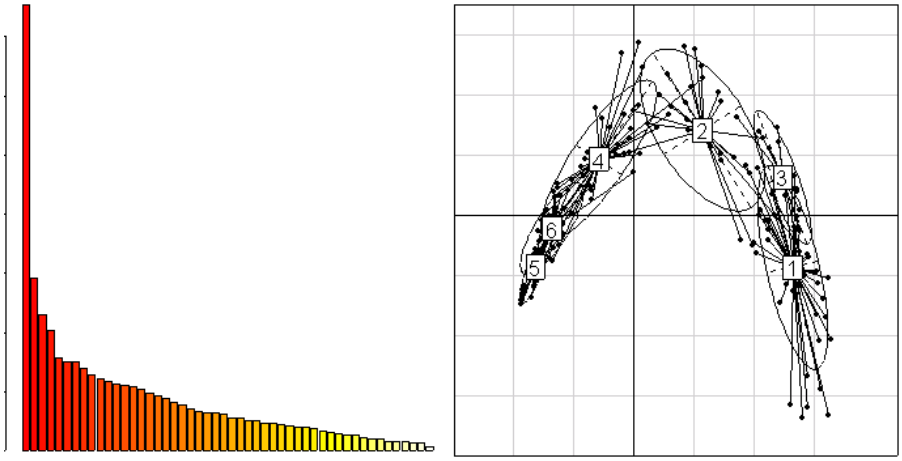
4.2. CAH et ordination

Le lien entre classification et carte factorielle est apparue très tôt et de manière intéressante.

> *Le gros avantage que nous voyons dans la méthode d'analyse factorielle [AFC] est qu'elle donne directement la figure représentative de l'ensemble à classer et ce avec une totale objectivité évidemment.* **35**

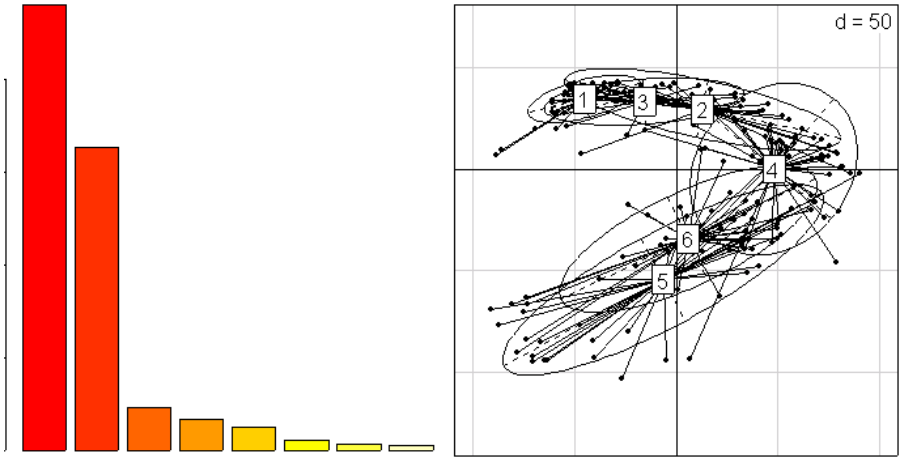
L'objectif est de classer (G. Roux est phytosociologue) et l'outil est l'ordination.

```
s.class(dudi.coa(rpjdl$fau)$li,favi)
Select the number of axes: 2
```




```
s.class(dudi.pca(rpjdl$mil,scale=F)$li,favi)
Select the number of axes: 2
```

35 Roux, G., and M. Roux. 1967. A propos de quelques méthodes de classification en phytosociologie. *Revue de Statistique Appliquée* XV:59-72. p. 69.

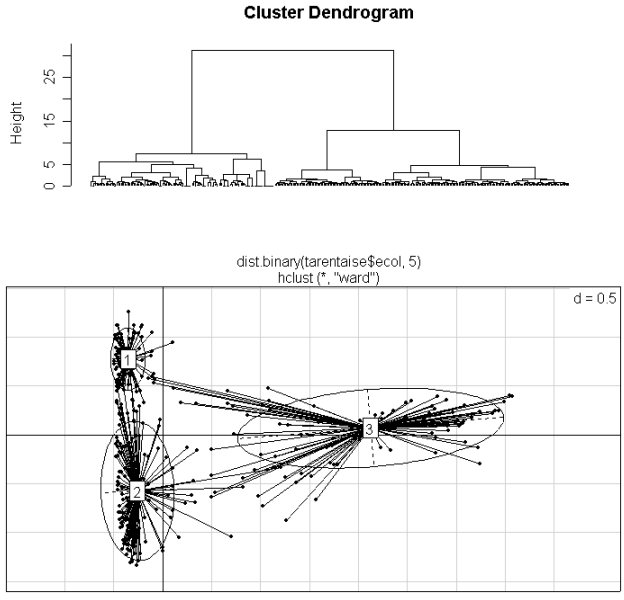


La classification qui semble pertinente renvoie à l'ordination. Il arrive que ce soit l'inverse. La seule chose à ne pas demander à ces outils, c'est *la vérité*.

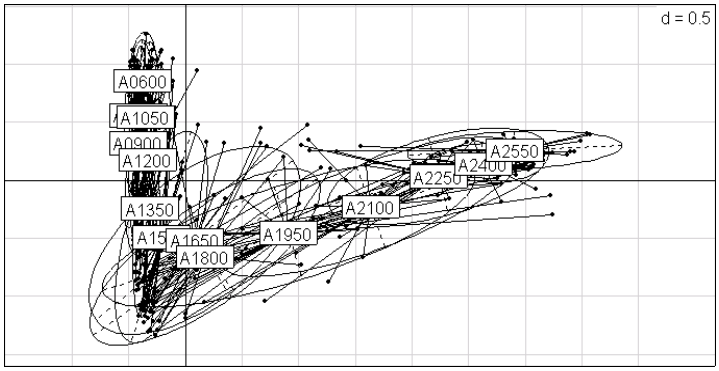
> *In most practical applications, reciprocal averaging gives stand ordinations which are similar to those derived by principal components analysis of standardized data. As a general method for use in phytosociological contexts it is preferable because it generates good simultaneous species ordinations. The rationale of the method is close to that of gradient analysis, so that it is more suitable than principal components analysis for displaying strong floristic gradients.*[36]

L'AFC est une bonne méthode pour représenter les relevés à classer et une bonne méthode pour trouver des gradients.

```
data(tarentaise)
par(mfrow=c(2,1))
h0 <-hclust(dist.binary(tarentaise$ecol,5),"ward")
plot(h0,hang=-1,lab=rep("",nrow(tarentaise$ecol)))
s.class(dudi.coa(tarentaise$ecol,scannf=F)$li,as.factor(cutree(h0,3)))
 s.class(dudi.coa(tarentaise$ecol,scannf=F)$li,tarentaise$envir$alti)
```

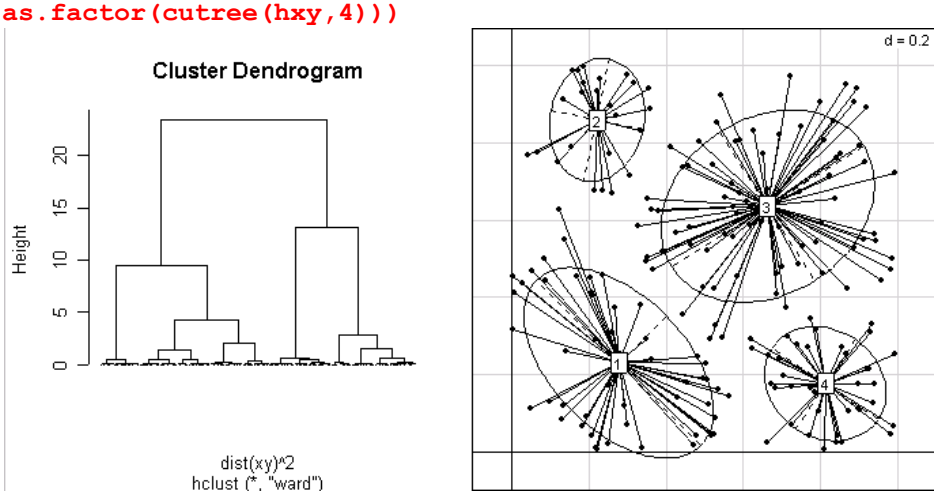


36 Hill, M. O. 1973. Reciprocal averaging : an eigenvector method of ordination. *Journal of Ecology* 61:237-249.



Il est rare que les structures observées s'adaptent facilement aux modèles censés les décrire. Il y a ici des coupures (changement d'étage de végétation vers 1900m) et des gradients partiels qui s'articulent par continuité. Il est facile de se faire berner.

```
x=runif(200) ; y=runif(200)
xy=cbind.data.frame(x,y)
hxy=hclust(dist(xy)^2,"ward")
plot(hxy,hang=-1,lab=rep("",200))
s.class(xy,as.factor(cutree(hxy,4)))
```



200 points au hasard donne une bonne partition !

Ce dont nous avons besoin c'est d'une méthode rigoureuse qui extraie des structures à partir des données [37] (p. 6).

On pourrait dire maintenant que nous avons besoin de plusieurs méthodes qui extraie des structures de plusieurs types et que, tôt ou tard, la méthode rigoureuse, si elle est seule, produira une vilainie.

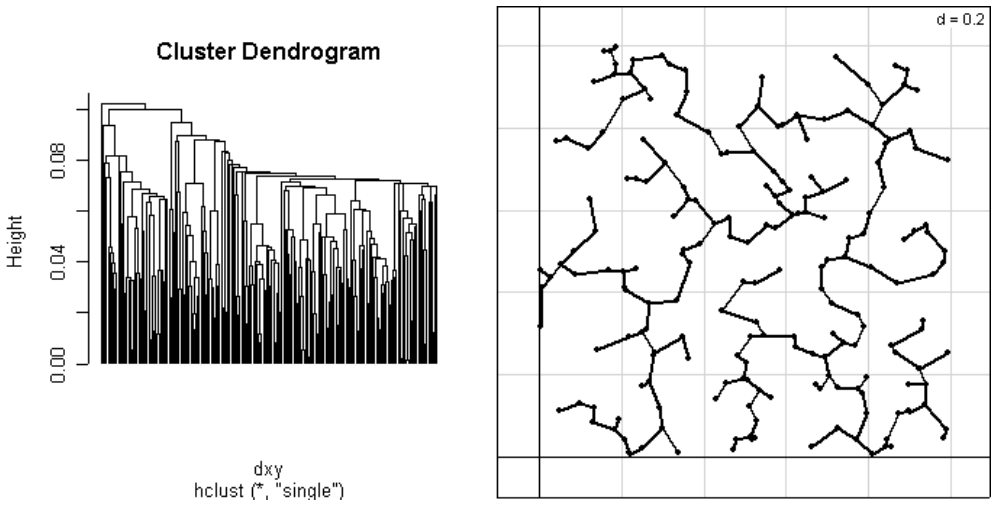
4.3. Arbre de longueur minimale et plus proche voisin

Le lien du plus proche voisin est cohérent avec l'arbre de longueur minimale MST qui est un outil graphique à connaître.

```
dxy = dist(xy)
```

[37] Benzécri, J. P., and Coll. 1973. L'analyse des données. II L'analyse des correspondances. Bordas, Paris.

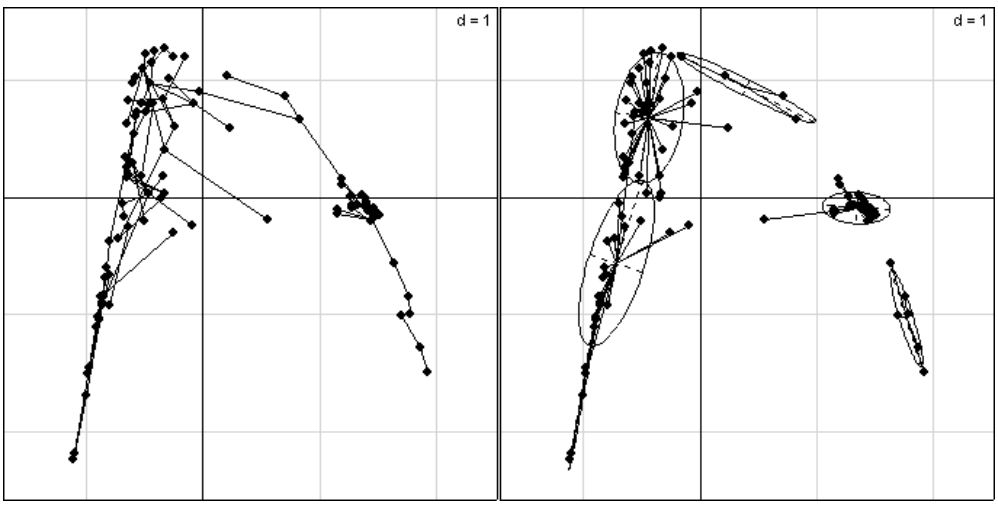
```
hxy=hclust(dxy,"single")
plot(hxy,hang=-1,lab=rep("",200))
mstxy=mstree(dxy)
s.label(xy,neig=mstxy,clab=0)
```



Le nuage est le même qu'au dessus. La CAH du lien simple montre les effets de chaîne. La distance du saut minimum tend à agréger un point à un groupe existant plutôt qu'à donner naissance à un nouveau groupe. La procédure ne donne pas satisfaction si on trouve des intermédiaires. Ces points intermédiaires, présents entre deux *clusters* sont considérés comme du bruit aléatoire : les autres méthodes en diminueront l'importance.

Mais le lien simple est lié à l'arbre de longueur minimal. Ce graphe défini sur l'ensemble des *n* points sur lequel on a une matrice de dissimilarité est sans cycle et connexe (c'est la définition d'un arbre). Deux points quelconques sont reliés par un chemin. Il a *n* - 1 arêtes. Par définition la somme des longueurs des arêtes (c'est-à-dire des distances entre deux points reliés) est minimale. C'est un moyen de voir la distance.

```
data(mafragh)
maf.coa=dudi.coa(mafragh$flo,scan=F)
maf.mst=mstree(dist.dudi(maf.coa),1)
s.label(maf.coa$li,clab=0,cpoi=2,neig=maf.mst,cnei=1)
s.class(maf.coa$li,as.factor(cutree(hclust(dist.dudi(maf.coa)^2,"ward"),7)),clab
=0,cpoi=2)
```

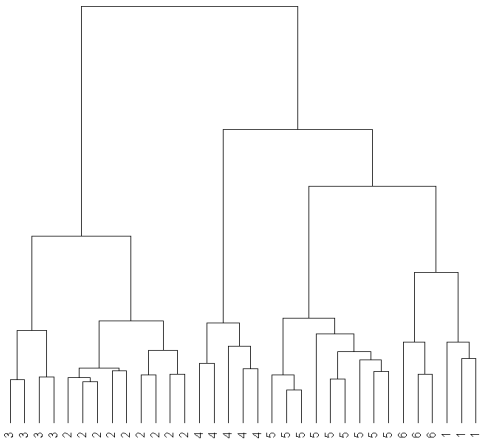


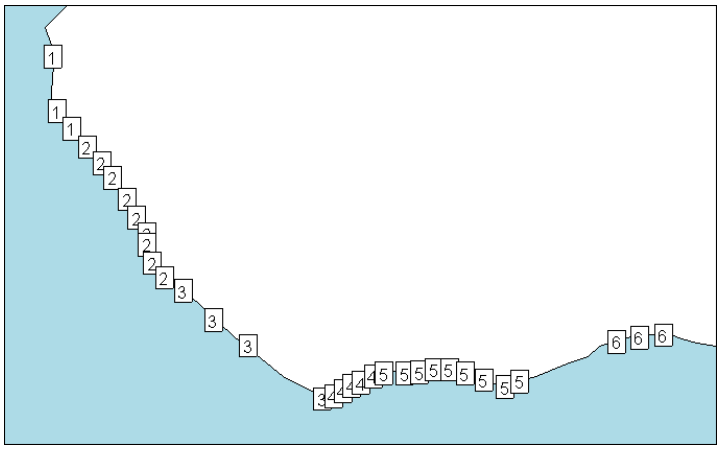
A gauche la recherche des chaînes par l'arbre de longueur minimale. A droite la recherche des groupes compacts avec la CAH de Ward. Sert de support l'ordination par le premier plan de l'AFC. Il est fréquent de rencontrer dans les

tableaux écologiques des mélanges complexes liés aux aléas de l'échantillonnage et au fonctionnement des écosystèmes.

4.4. Utiliser un dendrogramme

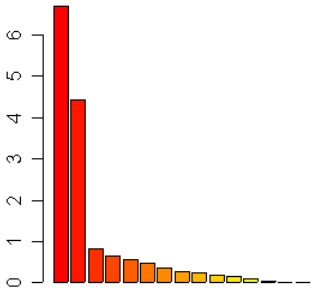
```
data(westafrica)
wa.d=dist.binary(as.data.frame(t(westafrica$tab)),1)
wa.cah=hclust(wa.d^2,"ward")
plot(wa.cah)
plot(wa.cah,lab=as.character(cutree(wa.cah,6)),hang=-1)
```

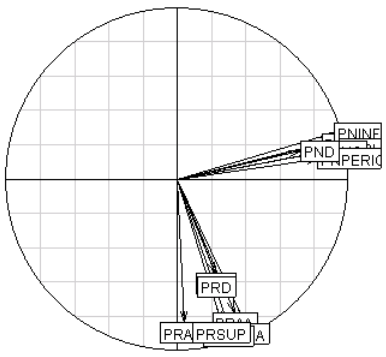




On a vu p.8 une carte donnant la représentation spatiale des embouchures des fleuves étudiés. Peut-on reporter un niveau de partition sur la carte ?

```
data(lascaux)
colo.acp=dudi.pca(lascaux$colo,scannf=FALSE)
par(mfrow=c(1,2))
barplot(colo.acp$eig)
s.corcircle(colo.acp$co,clab=0.75)
```

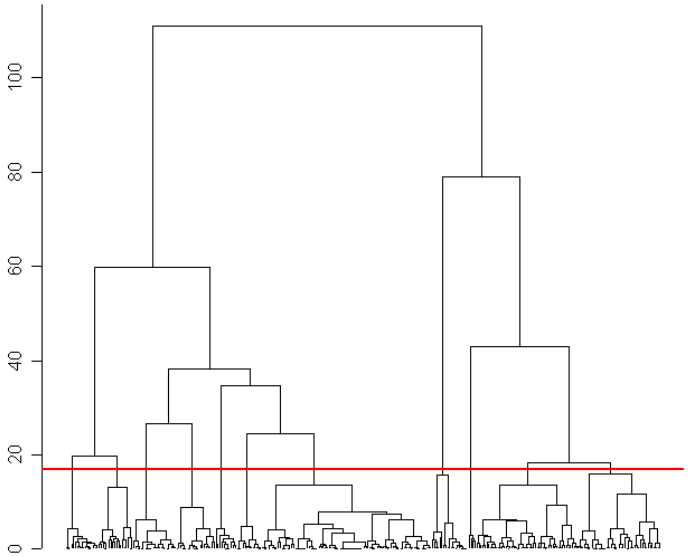





```
anova(lm(colo.acp$l1[,1]~lascaux$gen*lascaux$riv))
Analysis of Variance Table

Response: colo.acp$l1[, 1]
                      Df Sum Sq Mean Sq F value  Pr(>F)
lascaux$gen            6  156.3    26.0   75.23 < 2e-16
lascaux$riv           11   33.7     3.1    8.84 3.2e-13
lascaux$gen:lascaux$riv  42   30.9     0.7    2.13 0.00021
Residuals             246   85.2     0.3
```

On résume par une ordination transparente, on relie le score obtenu : la quantité de points noirs a une composante génétique et environnementale en inter-action.

```
ornem.d=dist.dudi(dudi.acm(lascaux$ornem,scannf=F))
ornem.cah=hclust(ornem.d^2,"ward")
plot(ornem.cah,hang=-1,lab=FALSE)
abline(h=17,lwd=2,col="red")
```

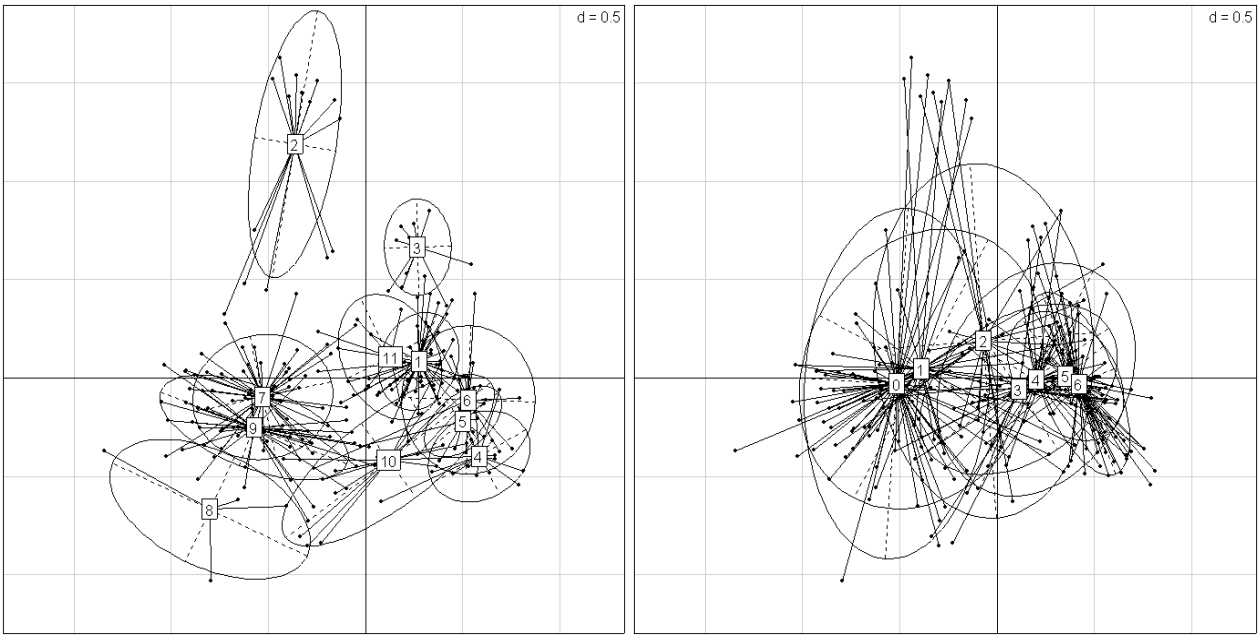



```
ornem.class=cutree(ornem.cah,h=17)
chisq.test(ornem.class,lascaux$gen)
```

```
data:  ornem.class and lascaux$gen
X-squared = 255.7, df = 60, p-value = < 2.2e-16
Warning message:
Chi-squared approximation may be incorrect in: chisq.test(ornem.class, lascaux$gen)
```

Le code génétique (Lascaux, op. cit. p.20) est basé sur 6 marqueurs allozymiques diagnostiques des populations de truites méditerranéenne (sauvages) et atlantiques (domestiques). La variable donne le nombre d'allèles méditerranéens (ancestraux) possédés par le poisson pour les trois systèmes enzymatiques. 0 indique un poisson assimilé à un homozygote atlantique (domestique) et 6 indique un poisson assimilé à un homozygote méditerranéen (sauvage). Le résultat est clair : on a dans les variables environnementales des variables discriminantes du statut génétique.

```
s.class(dudi.acm(lascaux$ornem,scannf=F)$li,as.factor(ornem.class))
s.class(dudi.acm(lascaux$ornem,scannf=F)$li,lascaux$gen)
```



A gauche, une partition obtenue par une CAH sur la distance de l'ACM d'un tableau de variables qualitatives, à droite la variable externe (code génétique). Le fond commun est le plan principal de l'ACM. La convergence entre l'approche ordination et l'approche classification donne du poids au résultat.

```
table(ornem.class,lascaux$gen)
```

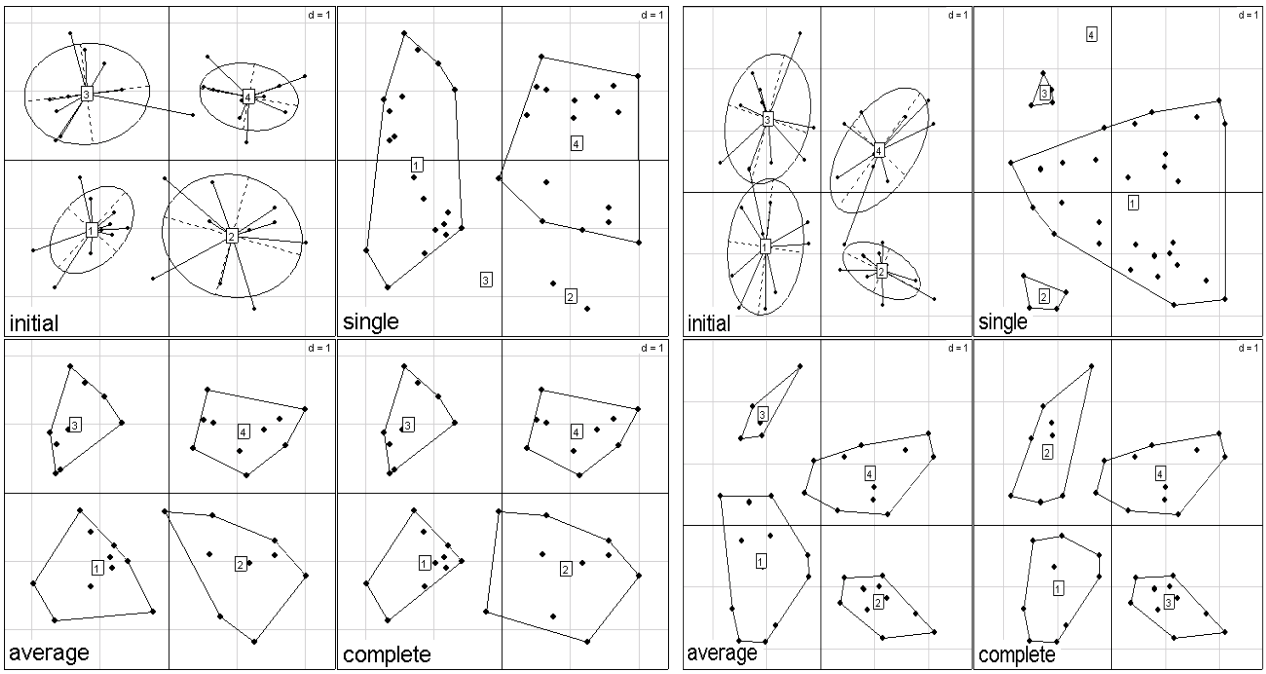
ornem.class	0	1	2	3	4	5	6
1	2	3	2	10	**14**	**19**	**37**
2	**10**	**6**	**3**	0	0	0	0
3	0	1	1	2	**4**	**1**	**4**
4	0	1	1	0	1	**2**	**12**
5	0	0	2	3	0	**6**	**11**
6	1	0	1	2	**3**	**1**	**12**
7	**27**	**15**	**5**	3	2	0	0
8	**4**	0	0	0	0	0	0
9	**23**	**11**	**4**	2	2	1	0
10	5	2	0	2	6	1	2
11	1	2	2	0	0	1	5

Explorer des pratiques de ce type sur les variables méristiques et morphométriques.

4.5. La recherche d'une partition

Les CAH donnent une idée sur la classifiabilité des données. En fait, on peut toujours trouver légitime de partager en paquet un ensmble de points même régulièrement répartis dans l'espace. Le problème est de ne pas faire d'erreurs grossières, lesquelles se voient bien en dimension 2, mais se cachent sans peine en dimension quelconque.

```
library(MASS)
?mvrnorm
```

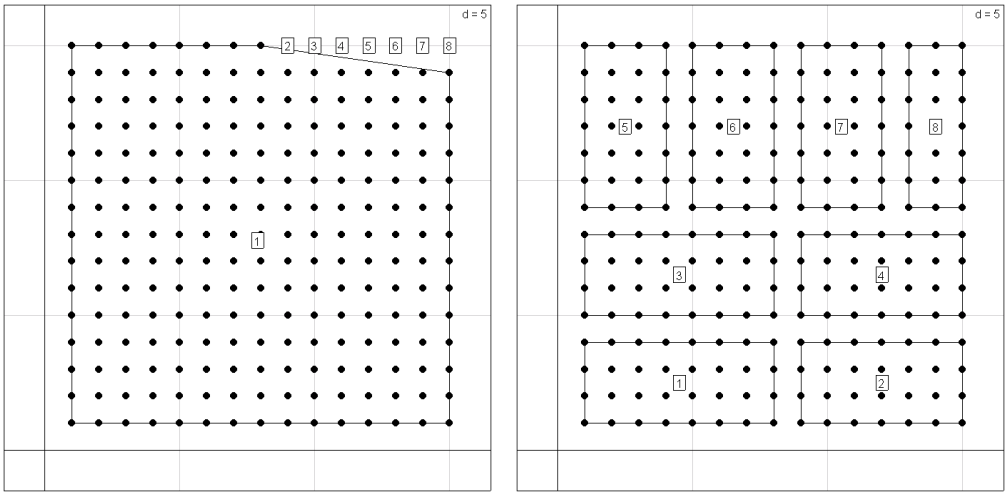


Le lien simple n'a guère d'intérêt pratique, le lien complet trouve au contraire des groupes sphériques, ce qui convient à la simulation. A gauche sd=0.25, à droite sd=0.5.

```
"fc" <- function(sd) {
       x1 = mvrnorm(10, mu= c(-1, -1), Sig=diag(sd, 2))
       x2 = mvrnorm(10, mu= c(1, -1), Sig=diag(sd, 2))
       x3 = mvrnorm(10, mu= c(-1, 1), Sig=diag(sd, 2))
       x4 = mvrnorm(10, mu= c(1, 1), Sig=diag(sd, 2))
       x = rbind(x1,x2,x3,x4)
       init = factor(rep(1:4,rep(10,4)))
       par(mfrow=c(2,2))
```

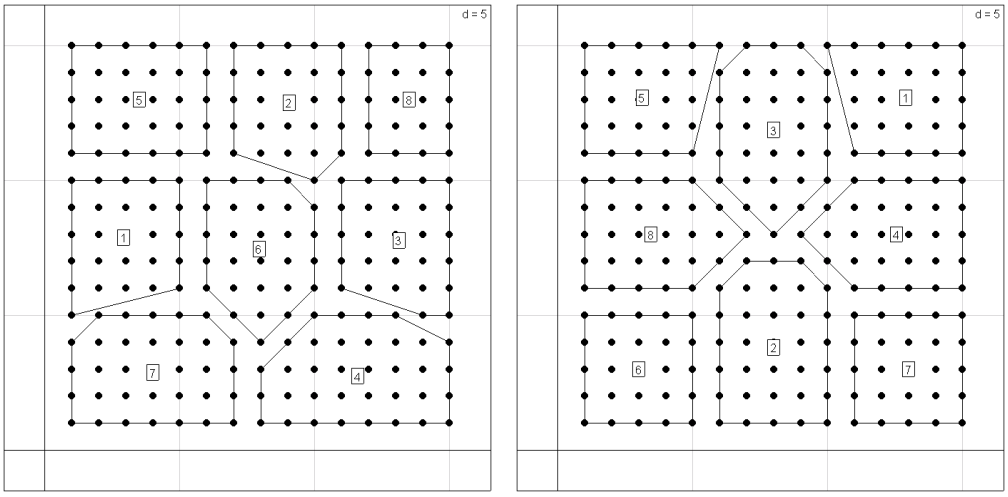
```
        s.class(x,init,sub="initial",csub=2)
        h0 = hclust(dist(x),"single")
        parti = as.factor(cutree(h0,k=4))
        s.chull(x,parti,sub="single",csub=2,opt=1,cpoi=2)
        h0 = hclust(dist(x),"average")
        parti = as.factor(cutree(h0,k=4))
        s.chull(x,parti,sub="average",csub=2,opt=1,cpoi=2)
        h0 = hclust(dist(x),"complete")
        parti = as.factor(cutree(h0,k=4))
        s.chull(x,parti,sub="complete",csub=2,opt=1,cpoi=2)
}
```

```
w=expand.grid(1:15,1:15)

s.label(w,clab=0,cpoi=2)
s.chull(as.data.frame(w),as.factor(cutree(hclust(dist(w)^2,"single"),8)),add.p=T
,opt=1)

s.label(w,clab=0,cpoi=2)
s.chull(as.data.frame(w),as.factor(cutree(hclust(dist(w)^2,"ward"),8)),add.p=T,o
pt=1)
```



On doit pouvoir faire mieux :

```
s.label(w,clab=0,cpoi=2)
s.chull(as.data.frame(w),as.factor(kmeans(w,8)$cluster),add.p=T,opt=1)
```



C'est mieux, mais pas toujours la même chose ! Il s'agit d'une agrégation autour des centres mobiles. L'explication donnée dans 32 p. 149 résume parfaitement la situation :

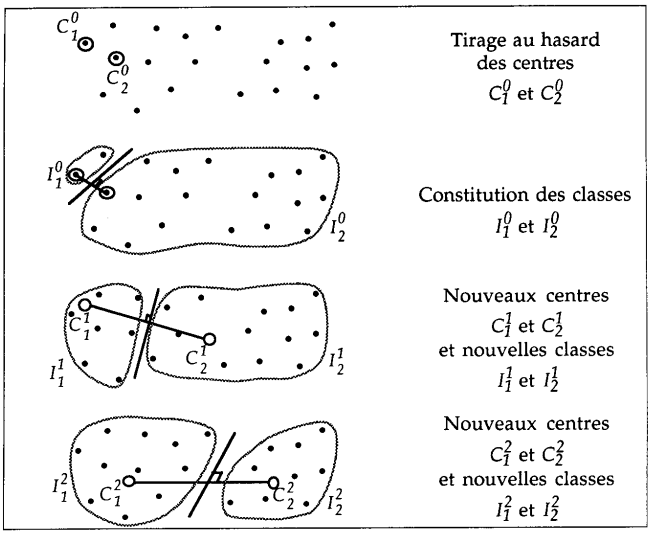
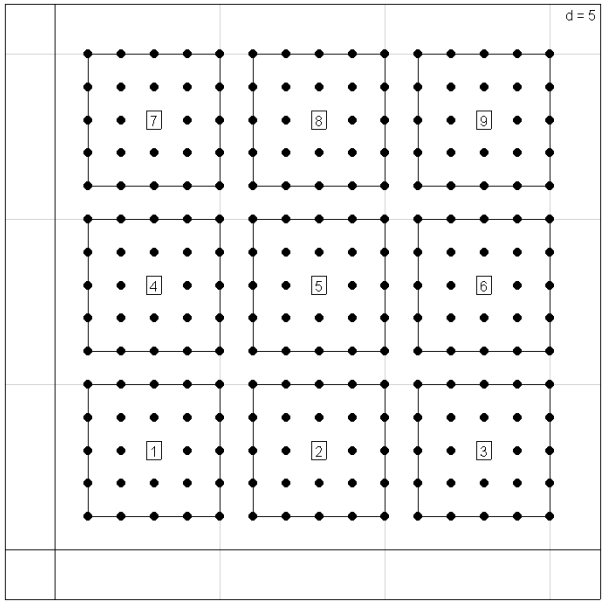


Figure 2.1 - 1
Etapes de l'algorithme

La fonction **38** calcule à chaque étape les centre de gravité des classes puis réaffecte chaque point au centre le plus proche. Elle accepte en entrée soit le nombre de classes (dans ce cas, la première série de centres est tirée au hasard), soit une liste de points qui serviront de centres de départ.

```
s.label(w,clab=0,cpoi=2)
cent <- expand.grid(c(3,8,13),c(3,8,13))
s.chull(as.data.frame(w),as.factor(kmeans(w,cent)$cluster),add.plot=T,opt=1)
```

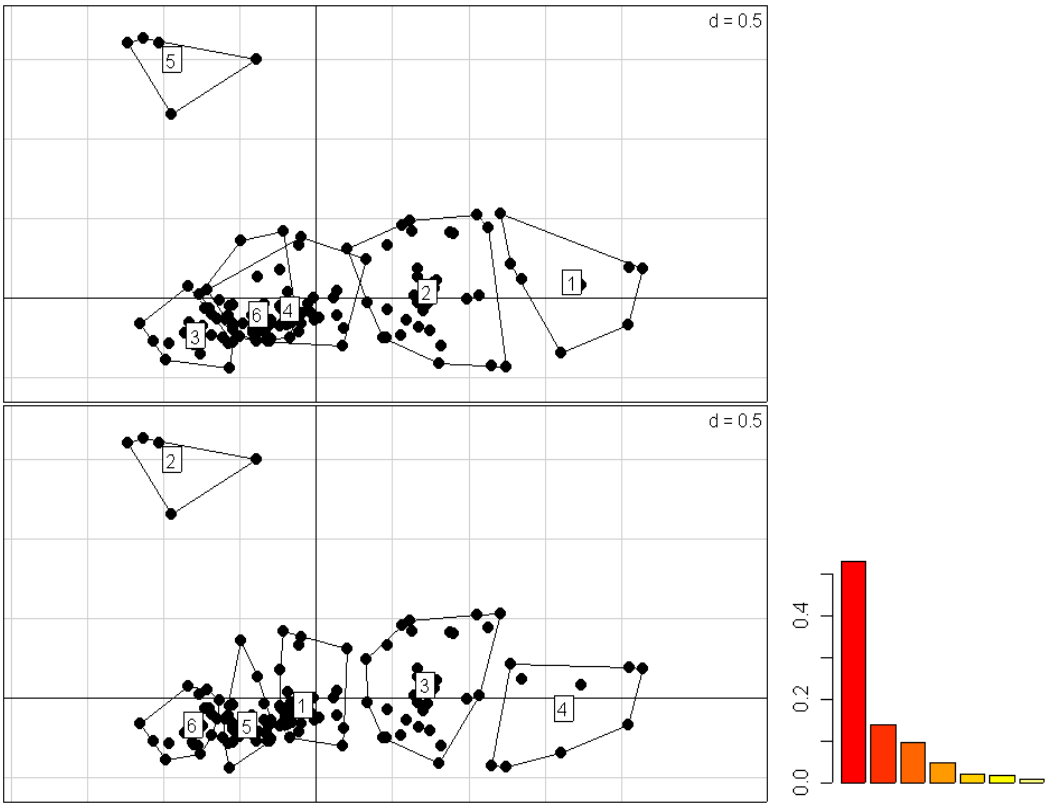


Les sorties sont simples.

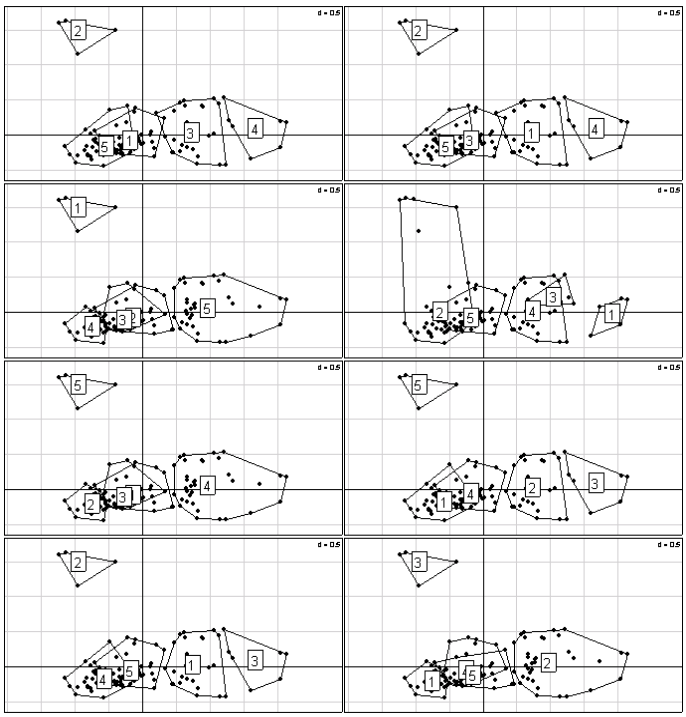
```
data(ecomor)
molo=log(ecomor$morpho)
molo=as.data.frame(t(apply(molo,1,function (x) x-mean(x))))

molo.pca=dudi.pca(molo,scale=F,scan=F)
par(mfrow=c(2,1))
```

38 Hartigan, J.A. and Wong, M.A. (1979). A K-means clustering algorithm. Applied Statistics, 28 , 100-108.

```
s.label(molo.pca$li,clab=0,cpoi=2)
km1=kmeans(molo,6)
s.chull(molo.pca$li,as.factor(km1$cluster),add.p=T,opt=1)
km2=kmeans(molo.pca$li,6)
s.label(molo.pca$li,clab=0,cpoi=2)
s.chull(molo.pca$li,as.factor(km2$cluster),add.p=T,opt=1)
```



Deux logiques très différentes. En haut représentation d'une classification sur les données initiales sur la carte factorielle de l'ordination (PCA). Les contradictions ou les écarts entre les deux peuvent exprimer ce qui se passe dans l'espace par projection sur un plan. En bas, classification sur les coordonnées factorielles représentée sur la carte factorielle. Quand il n'y a que deux facteurs conservés, il ne s'agit que de redondance inutile. Se méfier quand même du tirage aléatoire des centres de départ (ci -dessous).



```
par(mfrow=c(4,2))
for (k in 1:8) {
      s.label(molo.pca$li,clab=0,cpoi=2)
      km=kmeans(molo,5)
      s.chull(molo.pca$li,as.factor(km$cluster),add.p=T,opt=1,clab=2)
}
```

L'écologue prendra sans doute la décision de partir d'un choix motivé d'espèces *emblématiques*. Au total, la réduction d'un tableau à une variable qualitative, si elle n'est jamais simple est souvent un outil de description utile.

```
km
$cluster # le numéro du cluster pour chacun des points
  [1] 3 3 3 3 3 2 2 2 2 2 2 2 2 2 1 1 5 5 5 5 5 5 5 5 5 5 5 1 2 2 2 5 5 5 5 5 5 5 5
 [42] 5 5 5 5 5 1 1 1 4 2 5 2 2 5 2 2 2 4 4 4 2 5 1 5 5 4 1 5 1 4 4 5 2 2 2 2 4 4 2 1 4
 [83] 1 1 1 1 1 4 4 4 4 1 1 1 1 1 1 1 1 4 4 4 4 4 2 2 2 2 2 1 4 4 1 4 4 4 4 2 2 2 4 2
[124] 2 4 4 2 2 2

$centers # les moyennes par classe de chacune des variables
  wingl taill    culml   bilh  billw    tarsl    midtl     weig
1 1.420 1.432 -0.09333 -1.517 -1.096  0.25846  0.02309 -0.42774
2 1.306 1.070 -0.30969 -1.558 -1.280 -0.23411 -0.15958  1.16569
3 1.821 1.354  0.94054 -1.361 -0.955 -0.99575 -0.20333 -0.59966
4 1.433 1.208 -0.15881 -1.506 -1.001  0.04988 -0.05699  0.03120
5 1.356 1.134 -0.39050 -1.034 -1.033 -0.05532 -0.07628  0.09923

$withinss # les sommes de carrés des écarts par classe
[1]  4.5870 16.8711  0.8936  4.6746  6.4440

$size # Les tailles des groupes
[1] 26 37  5 28 33
```

Pour vérifier :

```
tapply(molo$wingl,as.factor(km$cluster),mean)
    1     2     3     4     5
1.420 1.306 1.821 1.433 1.356

for (k in 1:5) {
      w=dist(molo[km$cluster==k,])
      print(sum(w^2)/km$size[k])
}
[1] 4.587
[1] 16.87
[1] 0.8936
[1] 4.675
[1] 6.444
```

Exercice On donne les 7 points A, B, C, D, E, IF et G caractérisés par leurs abscisses :

$$x_A = 0 \quad x_B = 1 \quad x_C = 3 \quad x_D = 6 \quad x_E = 7 \quad x_F = 11 \quad x_G = 16$$

Appliquer à l'ensemble des 7 points les méthodes d'agrégation ascendante par l'inf et par le sup. On tracera les arbres de classification en indiquant les niveaux.

Même question avec les coordonnées :

$$x'_A = 7 \quad x'_B = 8 \quad x'_C = 5 \quad x'_D = 12 \quad x'_E = 11 \quad x'_F = 16 \quad x'_G = 0$$

Donner le nombre de hiérarchies binaires à 1, 2, 3, 4, 5, 6 et 7 éléments.

Pour en savoir beaucoup plus, voir le problème **39** p. 200 et sa solution par P. Cazes.

---

39 Benzécri, J. P., and Coll. 1973. L'analyse des données. I La taxinomie. Dunod, Paris. 615 p.

4.6. Outils graphiques autour de la représentation de l'arbre

```
data(westafrica) # liste de données voir p. 7
westafrica.d=dist.binary(as.data.frame(t(westafrica$tab)),7)
# Distance faunistique indice de Ochiai 40 voir p. 6. Citations :
```

Measures of similitude

- Jaccard similarity = $\frac{|A \cap B|}{|A \cup B|}$
- Ochiai similarity = $\frac{|A \cap B|}{\sqrt{|A| \cdot |B|}}$

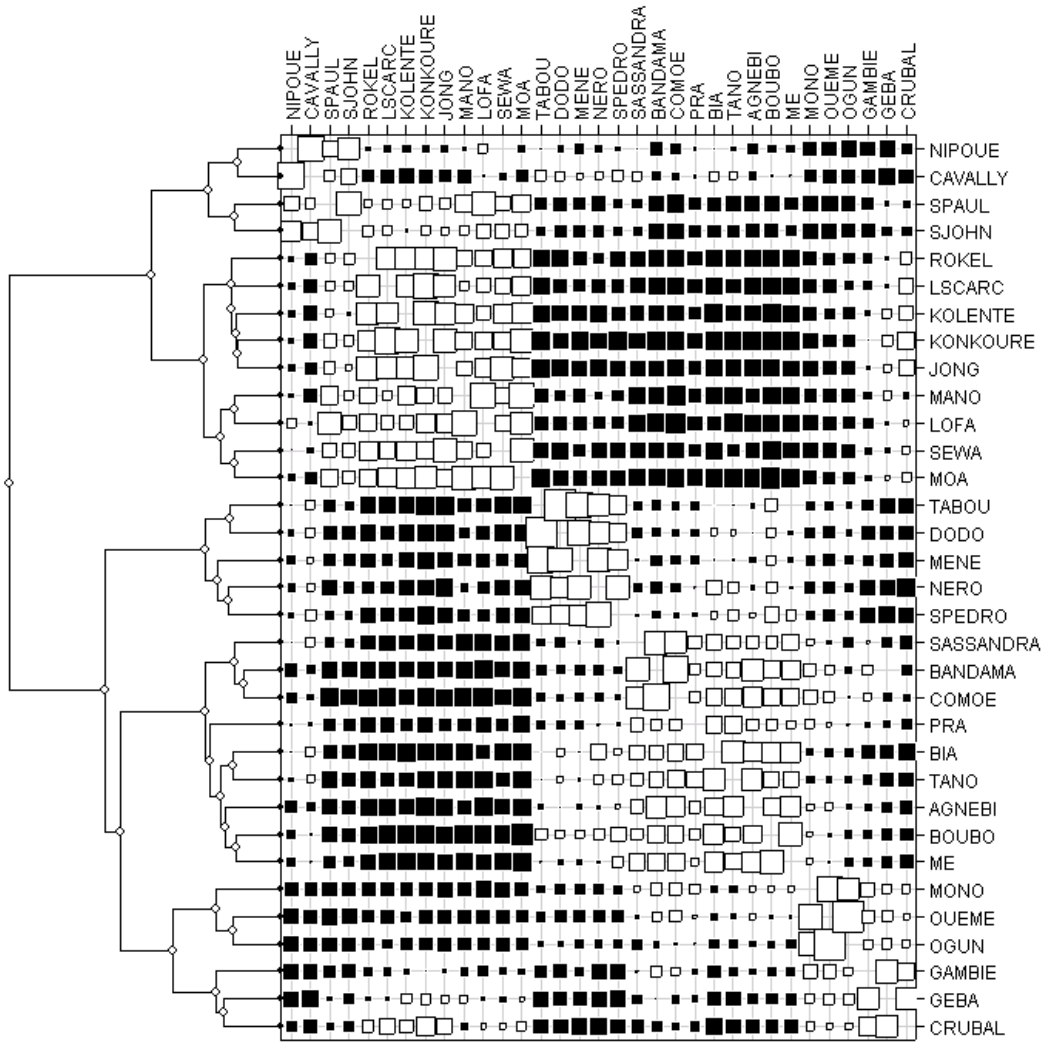
41

Ochiai similarity measure

the binary form of the cosine of the vector of values, which is a measure of pattern similarity. The formula is:

$$\text{OCHIAI}(x,y) = \text{SQRT}\left(\frac{a}{a+b} \cdot \frac{a}{a+c}\right)$$

42



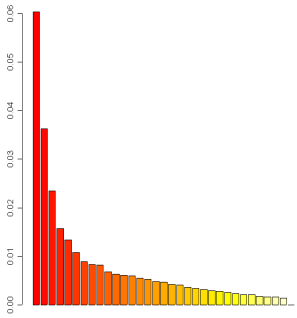
[40] Ochiai, A. 1957. Zoogeographic studies on the soleoid fishes found in Japan and its neighbouring regions. *Bulletin of the Japanese Society of Scientific Fisheries* **22**:526-530.

[41] http://www-poleia.lip6.fr/~rifqi/IFSA03trsp.pdf

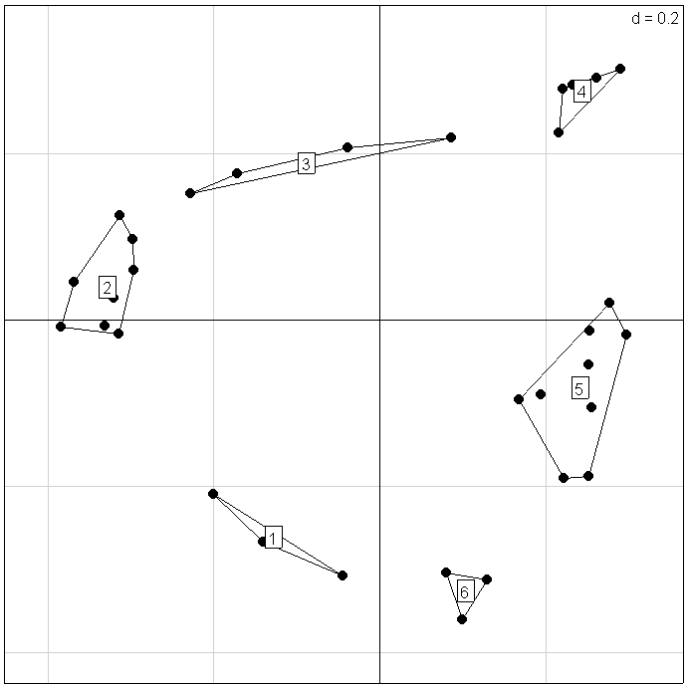
[42] http://www.usc.edu/isd/doc/statistics/pantry/strangestat.forweb.pdf

Pour refaire la figure :

```
westafrica.hc=hclust( westafrica.d,"ward")
# La CAH sur l'inertie intra
westafrica.phy=hclust2phylog(westafrica.hc,TRUE)
# l'objet équivalent dans la classe phylog de la librairie ade4 [43]
w=bicenter.wt(westafrica.d)
w=as.data.frame(w)
# la matrice de distances doublement centrée avec 0 sur la diagonale
w=w[names(westafrica.phy$leaves),]
w=w[,names(westafrica.phy$leaves)]
for(k in 1:nrow(w)) w[k,k]=0
# lignes et colonnes dans l'ordre des feuilles
table.phylog(w,westafrica.phy,f=0.3,clabel.r=0.75,clabel.c=0.75,cleg=0,csi=0.75)

westafrica.pco=dudi.pco(westafrica.d)
Select the number of axes: 3
```

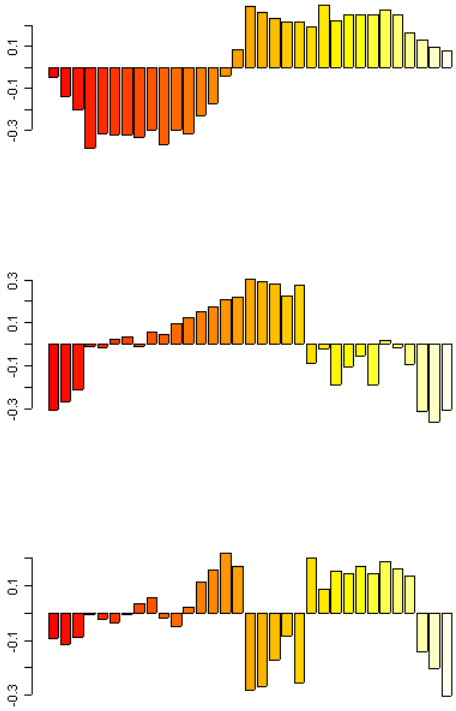



```
s.label(westafrica.pco$li,cpoi=2,clab=0)
s.chull(westafrica.pco$li,as.factor(cutree(westafrica.hc,6)),opt=1,add.p=T)
```



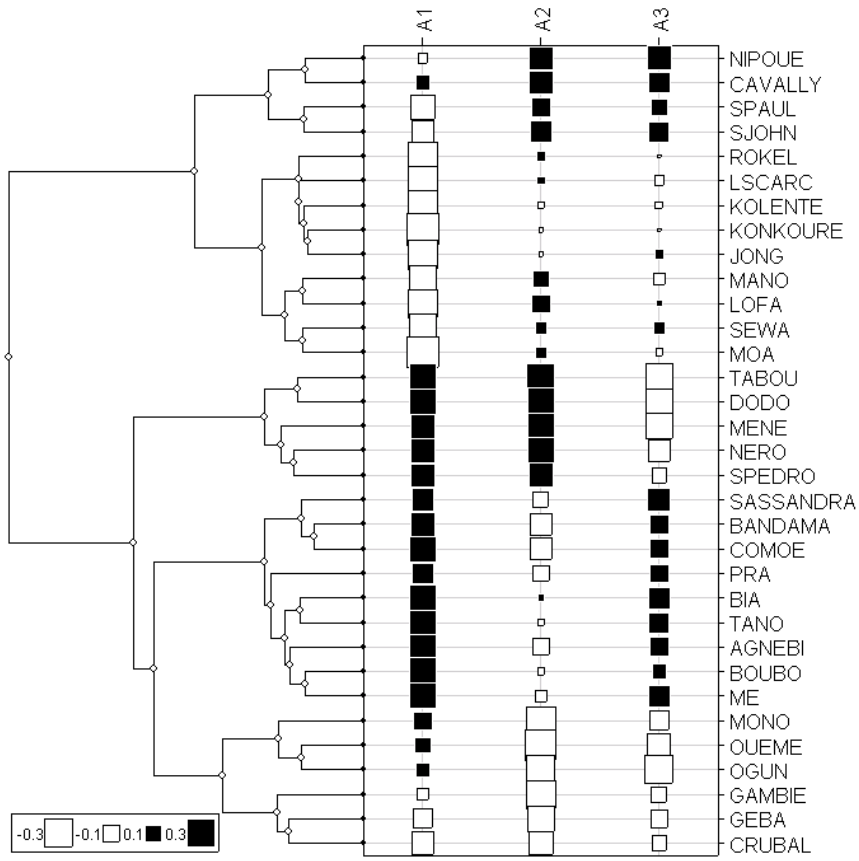
[43] Contributions de S. Ollier à la librairie ade4.

```
par(mfrow=c(3,1))
for(k in 1:3) barplot(westafrica.pco$li[,k])
```



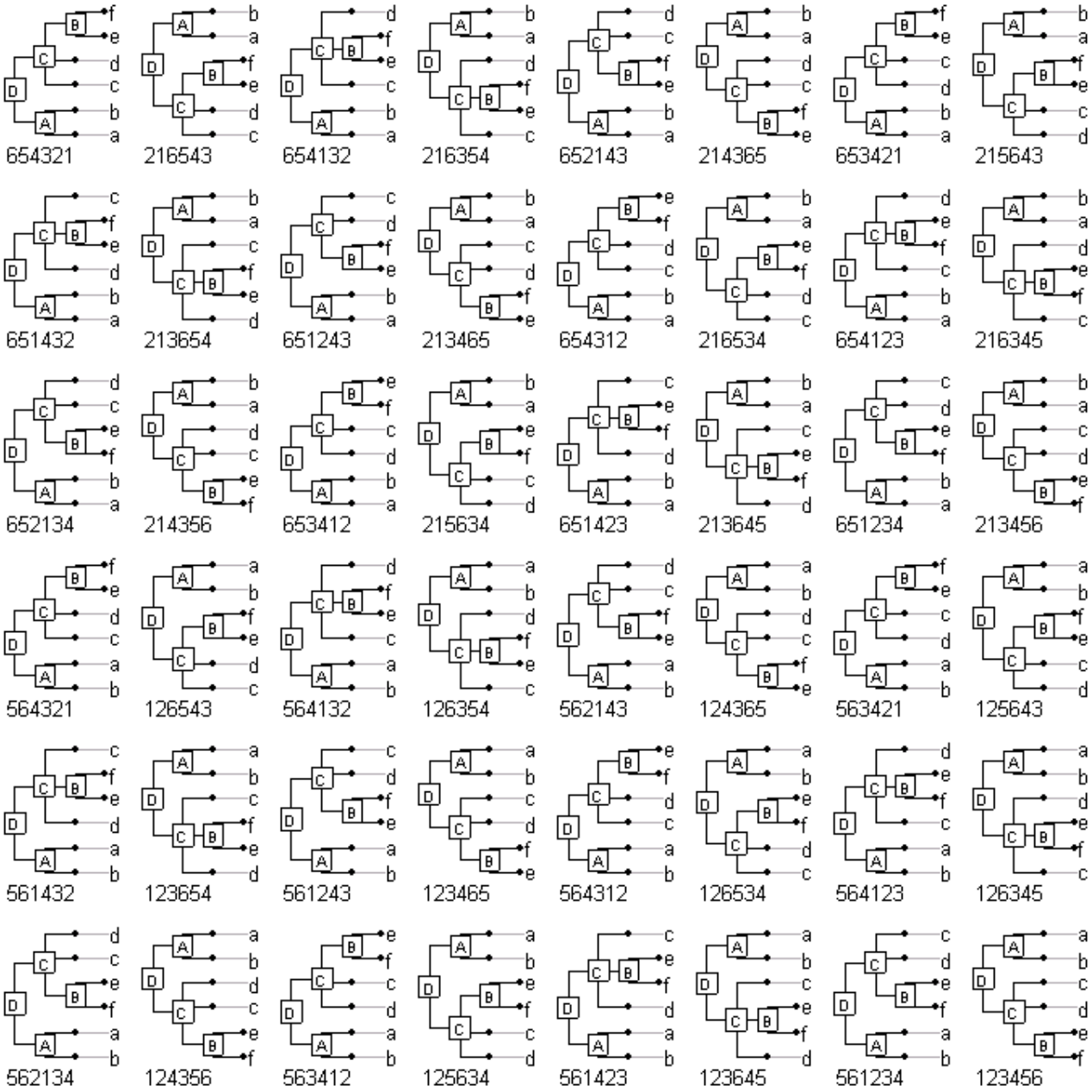
Alors, ordination ou classification ?

```
table.phylog(westafrica.pco$li,westafrica.phy,csi=2)
```



Une décision est souhaitable !

Dernière question : il y a 2^n représentations possibles de l'arbre dans une CAH complètement résolue. Pour une hiérarchie de partition incomplètement résolue combien y en a-t-il ? Un cas d'énumération complète :



Voir la documentation de **`plot.phylog`** *dans ade4.*

Dans le cas de **`westafrica`**, y en a t-il une qui serait complètement compatible avec la disposition spatiale ? Le dendrogramme est une figure au statut très particulier.