# Méthode de Parcimonie

## Procédures

### Algorithme exact

#### *Exemple du Quagga*

Le quagga est un animal éteint en 1883. Il en reste des exemplaires naturalisés dans différents musées. En utilisant des fragments d'os on a pu déterminer la séquence partielle de deux gènes mitochondriaux (cytochrome oxydase  et la NADH Déshydrogénase, 229bp en tout)


**Figure III-b 1. Une espèce disparue: le quagga.**

| | Cyt Ox | | | | NADH Dase | | |
|---|---|---|---|---|---|---|---|
| | **4** | **10** | **67** | **103** | **28** | **58** | **71** |
| **Quagga** | **A** | **C** | **T** | **T** | **C** | **C** | **T** |
| **Z.pl** | **A** | **C** | **T** | **T** | **C** | **C** | **T** |
| **Z.mt** | **A** | **T** | **C** | **T** | **T** | **C** | **C** |
| **Cheval** | **G** | **T** | **C** | **C** | **C** | **T** | **C** |
| **Vache** | **G** | **T** | **C** | **C** | **T** | **T** | **A** |

**Tableau III-b 1. Les positions variables des séquences mitochondriales partielles.**

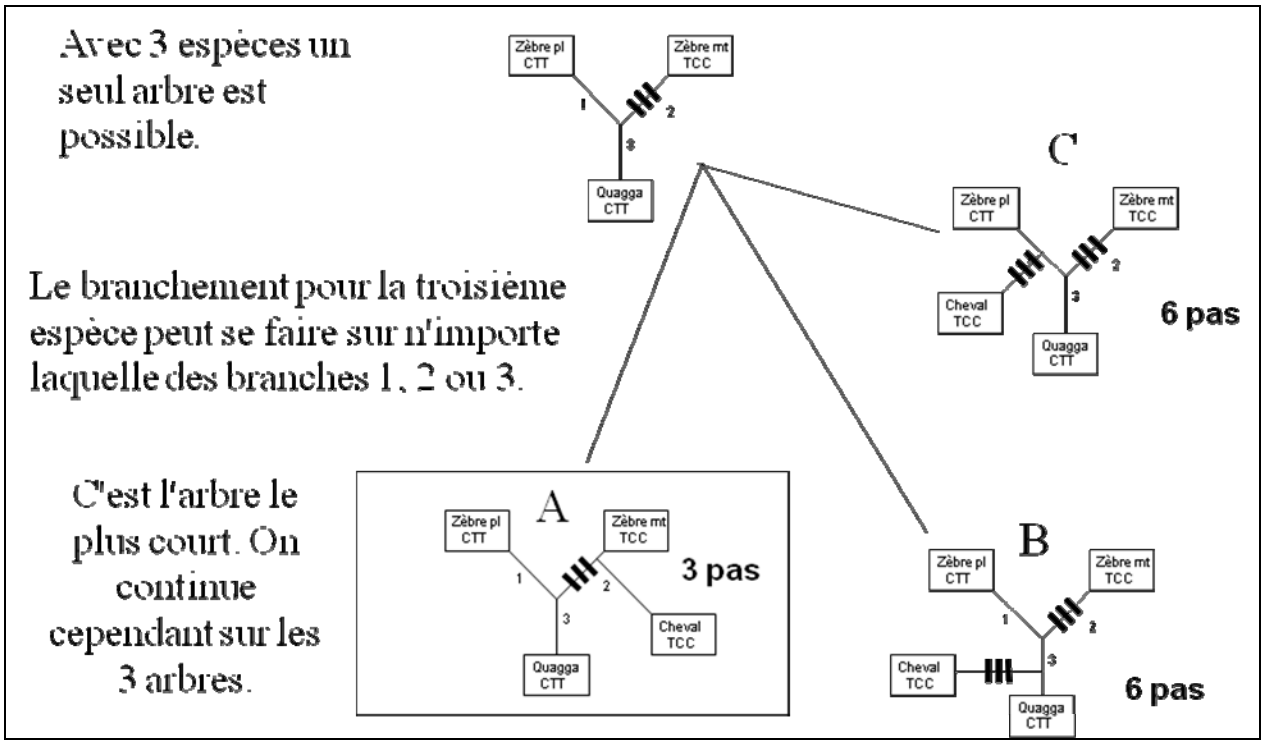


**Figure III-b 2. Pour construire ces arbres on commence par trois taxons puis on en ajoute un (trois possibilités).**

On va calculer tous les arbres possibles pour ces cinq taxons à partir des données moléculaires du tableau 3. On trace d'abord l'arbre étoile pour trois taxons. Il comporte trois branches. Le quatrième taxon peut se placer sur chacune de ces branches. Avec ces quatre taxons il n'y a que trois caractères informatifs. Ajouter la vache augmente le nombre des caractères informatifs à 7. On doit indiquer pour chacun leur état aux nœuds.

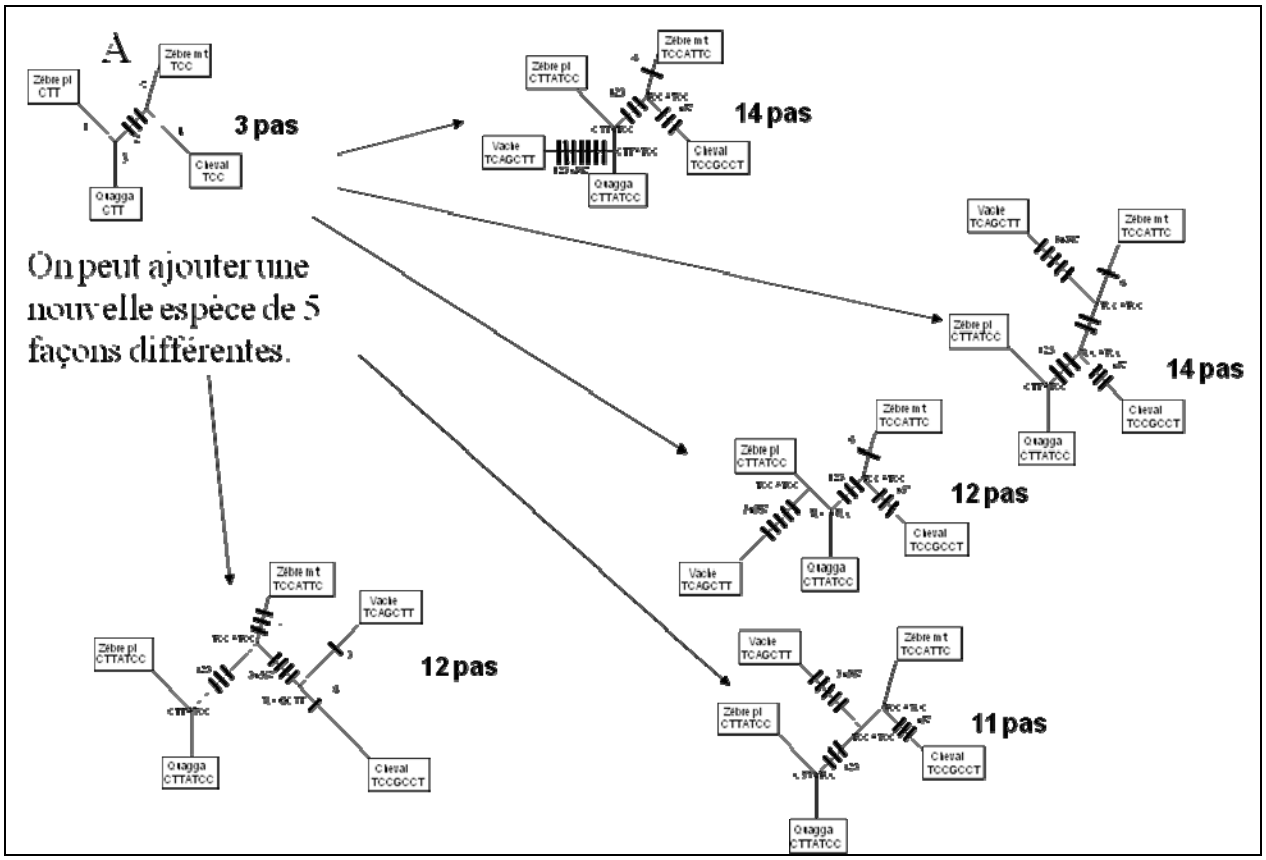


**Figure III-b 3. On ajoute ensuite le cinquième taxon. Pour chacun de ces trois arbres il y a cinq branches possibles soit en tout 15 arbres. Ici les cinq arbres .**

Quinze arbres sont obtenus de longueurs variées.

| **Arbre** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Pas** | **15** | **12** | **15** | **15** | **14** | **15** | **15** | **12** | **15** | **14** | **14** | **14** | **12** | **9** | **11** |

C’est l’arbre 14 qui est le plus parcimonieux.

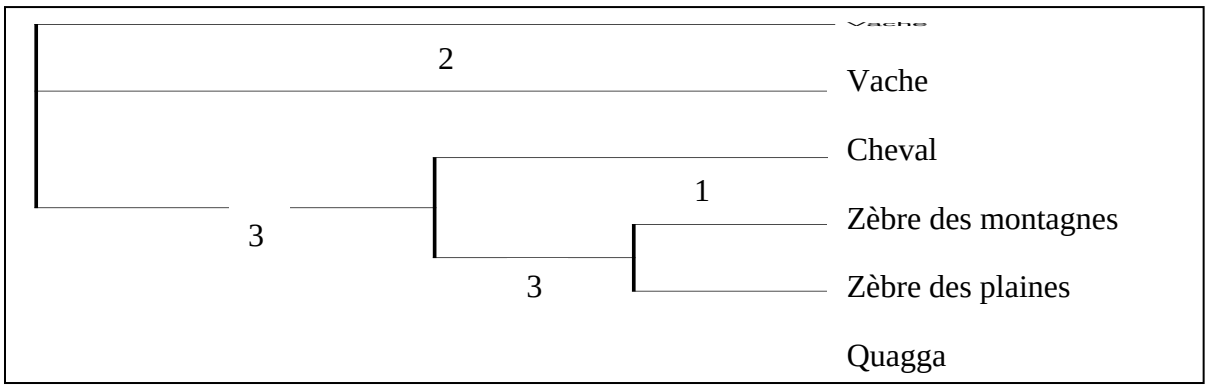


**Figure III-b 4. Arbre le plus parcimonieux (le nombre de pas est indiqué au dessus de chaque branche).**

Cette méthode exhaustive examine tous les arbres possibles en ajoutant les taxons un par un. Le nombre d’arbres possibles augmente rapidement.

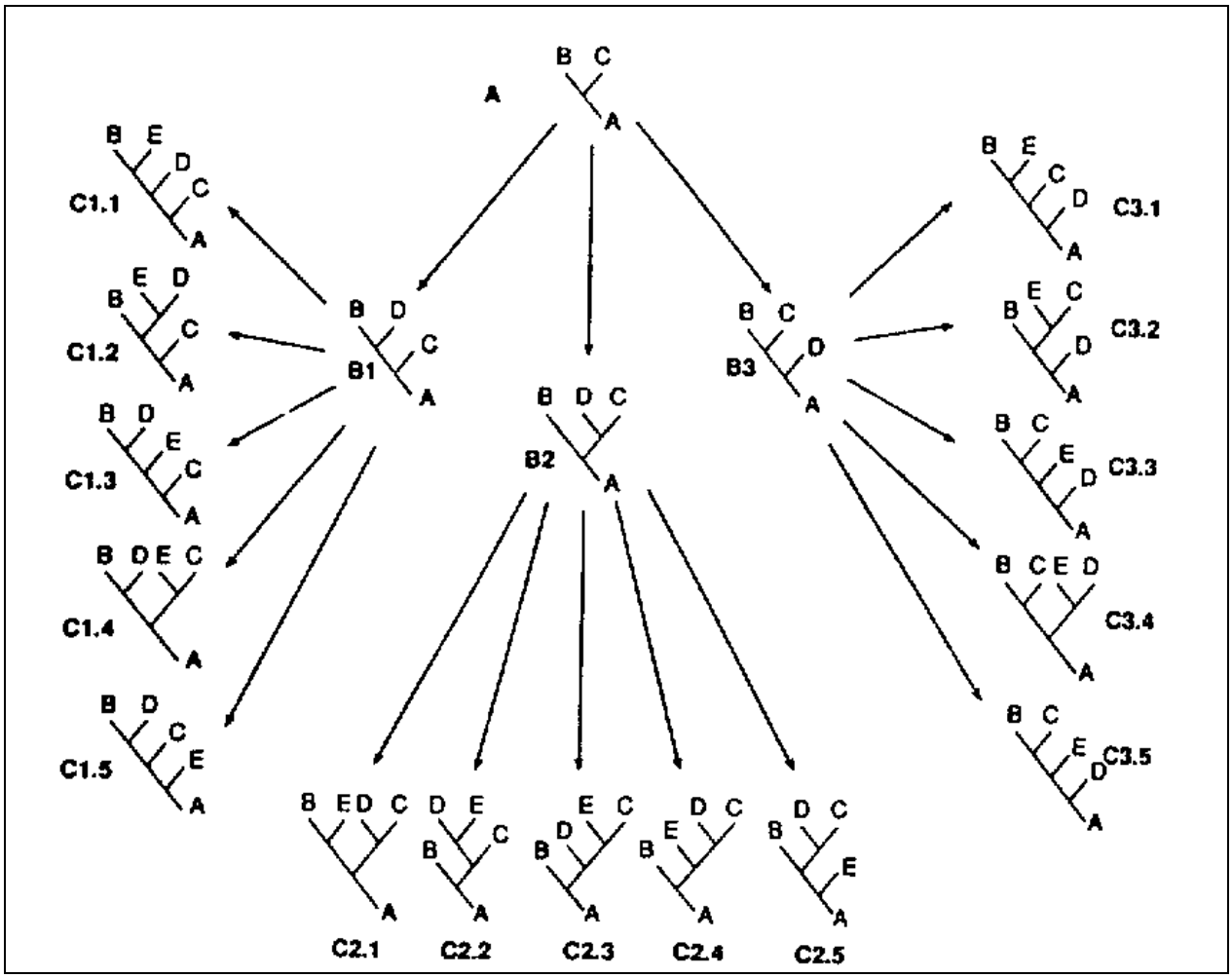


**Figure III-b 5. Méthode exhaustive : après avoir évalué tous les arbres on choisit le ou les plus courts.**

## Algorithme Branch and bound

Pour aller un peu plus vite sans abandonner la recherche exhaustive la procédure du branch and bound commence par évaluer un arbre au hasard. Ensuite, au cours de la recherche exhaustive des différents arbres, elle va abandonner une voie à partir du moment où le début d'arbre présente déjà une longueur supérieure à celle de l'arbre au hasard. lorsqu'au bout de sa recherche dans une voie elle trouve un arbre plus court que l'arbre au hasard du début, elle garde cette nouvelle valeur comme limite supérieure. Néanmoins, l'algorithme exhaustif seul peut donner tous les arbres suboptimaux.

*Un exemple de l'efficacité de cet algorithme : sur un jeu de 11 taxons (34 459 425 arbres binaires en tout)il fallait compter en 1982 55 jours pour trouver tous les arbres possibles. Le branch and bound a permis de réduire ce temps à un peu moins de 5mn.(Testing the theory of descent, Penny, Hendy and Steel (1991) in Phylogenetic analysis of DNA sequences ed Miyamoto and Cracraft).*

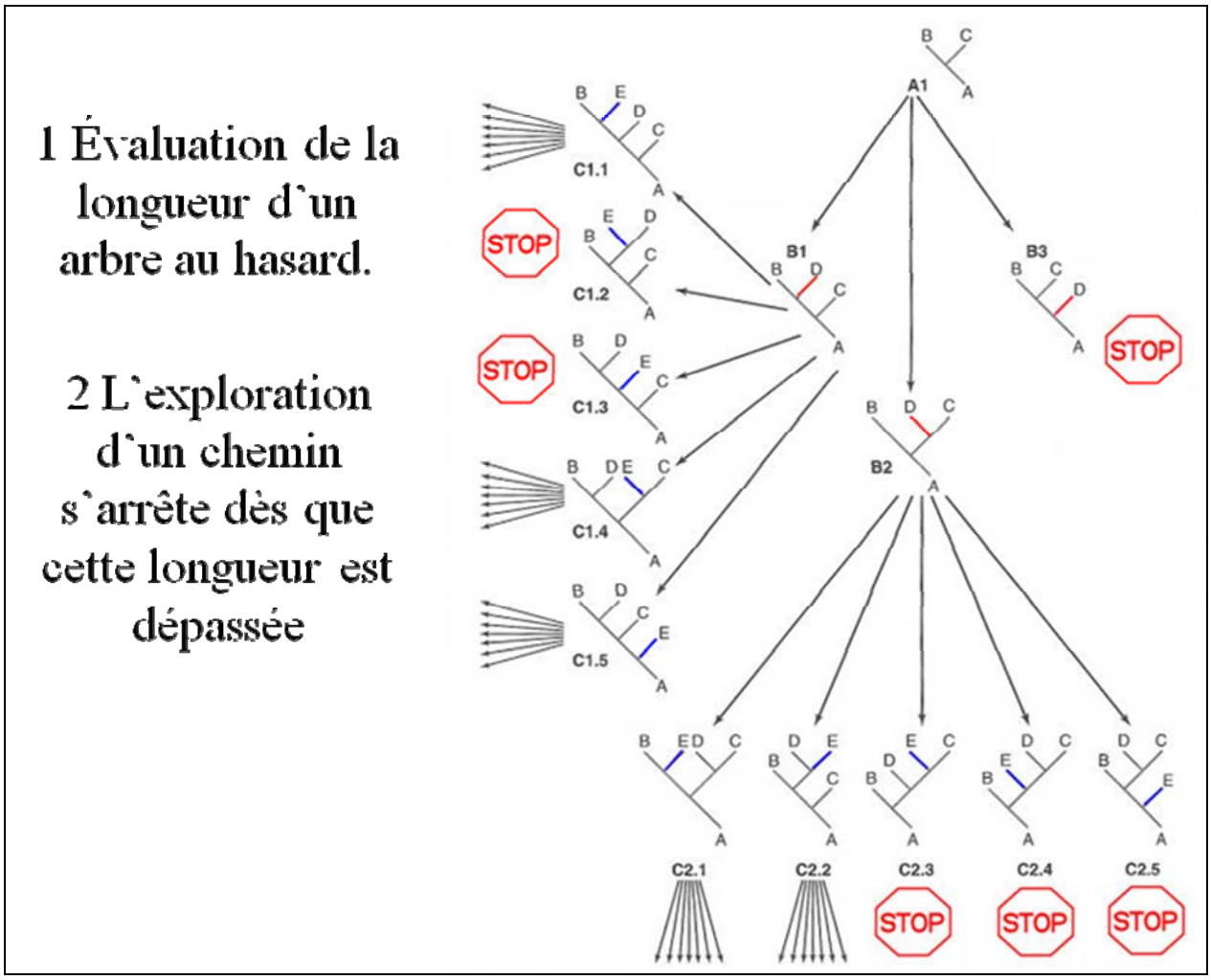


**Figure III-b 6. Méthode Branch and Bound (BB).**

## Algorithme heuristique

Les exemples précédents comportaient très peu de taxons et généraient peu d'arbres mais ce nombre croît avec le nombre de taxons. Dans quelles proportions ?
Pour calculer le nombre d'arbres possibles entre plusieurs taxons on considère la façon de déduire les arbres les uns des autres : pour 3 taxons il n'existe qu'un arbre qui présente 1 nœud, 3 segments externes et pas de segments internes. Le 4$^{e}$ taxon peut être ajouté sur chaque segment de cet arbre : 3 arbres possibles qui ont 2 nœuds,1 segment interne et 4 segments externes. Le cinquième taxon peut être inséré sur chacun des segments (interne ou externe) de chacun des 4 arbres, on obtient ainsi 3*(1+4)=15 arbres. Un arbre à x taxons présente

- x-2 nœuds
- x-3 segments internes
- x segments externes

Il y a Tx arbres possibles. L'insertion du taxon x+1 pourra se faire sur {(x-3)+x}=(2x-3) segments, ce qui générera

Tn=Tn-1 * {2(n-1) – 3} arbres (avec x=n-1)

Par récurrence on montre que

Tn=produits de k=3 à n de l'expression (2k-5)

$$T_n = \prod_{k=3}^{n} (2k-5)$$

En plus ces arbres sont dessinés non racinés. On peut choisir de placer la racine sur n'importe lequel des segments. Le nombre d'arbres racinés est

Tn=produits de k=2 à n de l'expression (2k-3)

$$T_n^{'} = \prod_{k=2}^{n} (2k-3)$$

| Nb Taxa | Nombre d'arbres non racinés | Nombre d'arbres racinés |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 1 | 1 |
| 3 | 1 | 3 |
| 4 | 3 | 15 |
| 5 | 15 | 105 |
| 6 | 105 | 945 |
| 7 | 945 | 10 395 |
| 8 | 10 395 | 135 135 |
| 9 | 135 135 | 2 027 025 |
| 10 | 2 027 025 | 34 459 425 |
| 11 | 34 459 425 | 654 729 075 |
| 12 | 654 729 075 | 13 749 310 575 |
| 13 | 13 749 310 575,00 | 316 234 143 225,00 |
| 14 | 316 234 143 225,00 | 7 905 853 580 625,00 |
| 15 | 7 905 853 580 625,00 | 213 458 046 676 875,00 |
| 16 | 213 458 046 676 875,00 | 6 190 283 353 629 370,00 |
| 17 | 6 190 283 353 629 370,00 | 191 898 783 962 511 000,00 |
| 18 | 191 898 783 962 511 000,00 | 6 332 659 870 762 850 000,00 |
| 19 | 6 332 659 870 762 850 000,00 | 221 643 095 476 700 000 000,00 |
| 20 | 221 643 095 476 700 000 000,00 | 8 200 794 532 637 890 000 000,00 |

**Tableau III-b 2. Le nombre d'arbres croît rapidement avec le nombre de taxa.**

Même si les arbres à retenir avant le choix sont non racinés, leur nombre augmente considérablement au fur et à mesure du nombre de taxons. Il devient très vite impossible de considérer tous les arbres possibles. Il va donc falloir choisir parmi quelques uns construits au hasard (procédure heuristique), des essais seront alors nécessaires pour espérer obtenir quand même un arbre parmi les plus court. On utilise des techniques dites d'exploration de collines (« hill-climbing technics ») pour chercher dans un paysage d'arbres de plus en plus grand. La manière d'arriver à une solution optimale locale peut empêcher d'explorer des arbres plus longs (on violerait le principe de parcimonie) pour arriver à une solution optimale globale.

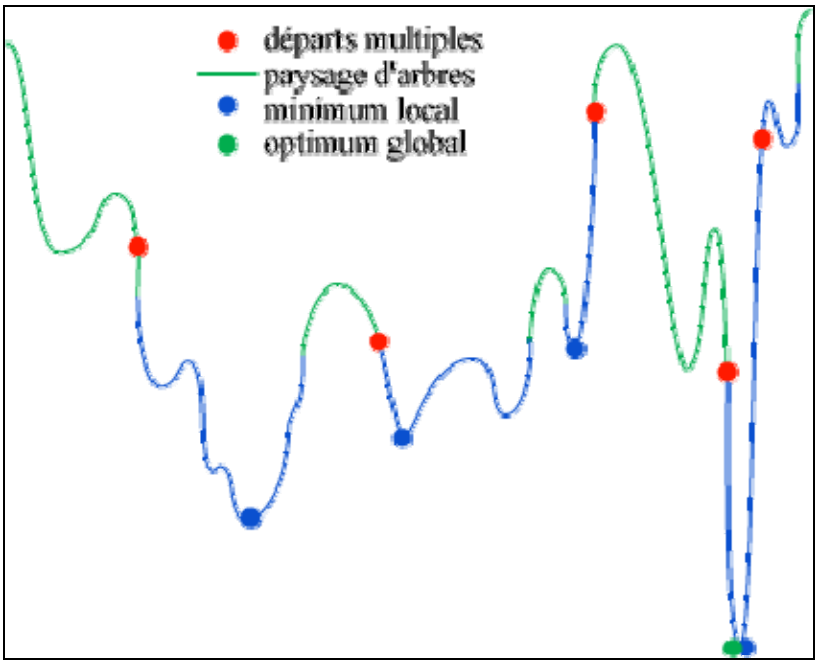


**Figure III-b 7. Plusieurs points de départ indépendants assurent un meilleur balayage du paysage d'arbres.**

Une façon de tourner la difficulté est de recommencer la recherche au hasard d'un point différent. On accroît ainsi la probabilité de trouver l'arbre le plus court de tous, sans pour autant garantir le résultat.

### Algorithme de Wagner (1961)

C'est l'ancêtre des algorithmes heuristiques actuels utilisés par les programmes de parcimonie (dont les algorithmes ne sont pas publiés).
Ces algorithmes connectent les unités évolutives entre elles, en construisant un arbre de telle façon que le nombre total de transformations de caractères soit minimal. L'état des caractères est établi pour chaque nœud en maximisant les synapomorphies et en minimisant les homoplasies. Chaque nœud prend valeur d'UEH caractérisée comme les UE. On dresse tout d'abord la liste des distances Manhattan entre tous les couples de taxons. A et B étant les deux taxons.

$x_{Ah}$ = état du caractère h pour l'UE A. Il y a en tout K caractères.

$$d_{AB} = \sum_{h=1}^{K} \left| x_{Ah} - x_{Bh} \right|$$

- On connecte les UE pour lesquelles cette différence est la plus grande (les plus éloignées).
- Une autre UE est ensuite connectée en un nœud Y. Cette troisième UE est choisie de telle sorte que la distance entre elle et Y soit maximale. Cette distance est estimée :

$$d_{CY} = \frac{1}{2}\left(d_{AC} + d_{BC} - d_{AB}\right)$$

- Les états de caractères de l'UEH Y sont définis de la façon suivante : pour un caractère donné c'est la médiane des états de A, B et C(médiane valeur sur l'histogramme du 50$^{e}$ de la population. C'est cette règle qui assure le minimum de transformations.

$$x_{Yh} = médiane(x_{Ah}, x_{Bh}, x_{Ch})$$

- Les états de caractères étant connus pour Y il est maintenant possible d'agglomérer une nouvelle UE comme précédemment en prenant comme critère celle qui a une distance maximale du second nœud Y'. Cette distance se calcule en fonction des distances entre UE et non avec les distances entre UE et UEH..

$$d_{DY'} = \frac{1}{2}\left(d_{CD} + d_{YD} - d_{YC}\right)$$

$$d_{YD} = \frac{1}{2}\left(d_{AD} + d_{BD} - d_{AB}\right)$$

$$d_{YC} = \frac{1}{2}\left(d_{AC} + d_{BC} - d_{AB}\right)$$

$$d_{YD} = \frac{1}{2}\left[d_{CD} + \frac{1}{2}\left(d_{AD} + d_{BD}\right) - \frac{1}{2}\left(d_{AC} + d_{BC}\right)\right]$$

Cette méthode qui peut rappeler un processus phénétique s'en distingue totalement par l'attribution d'états de caractères aux nœuds. Cependant il n'est pas assuré d'arriver à l'arbre le plus court et il convient d'effectuer des tests supplémentaires pour s'en assurer.

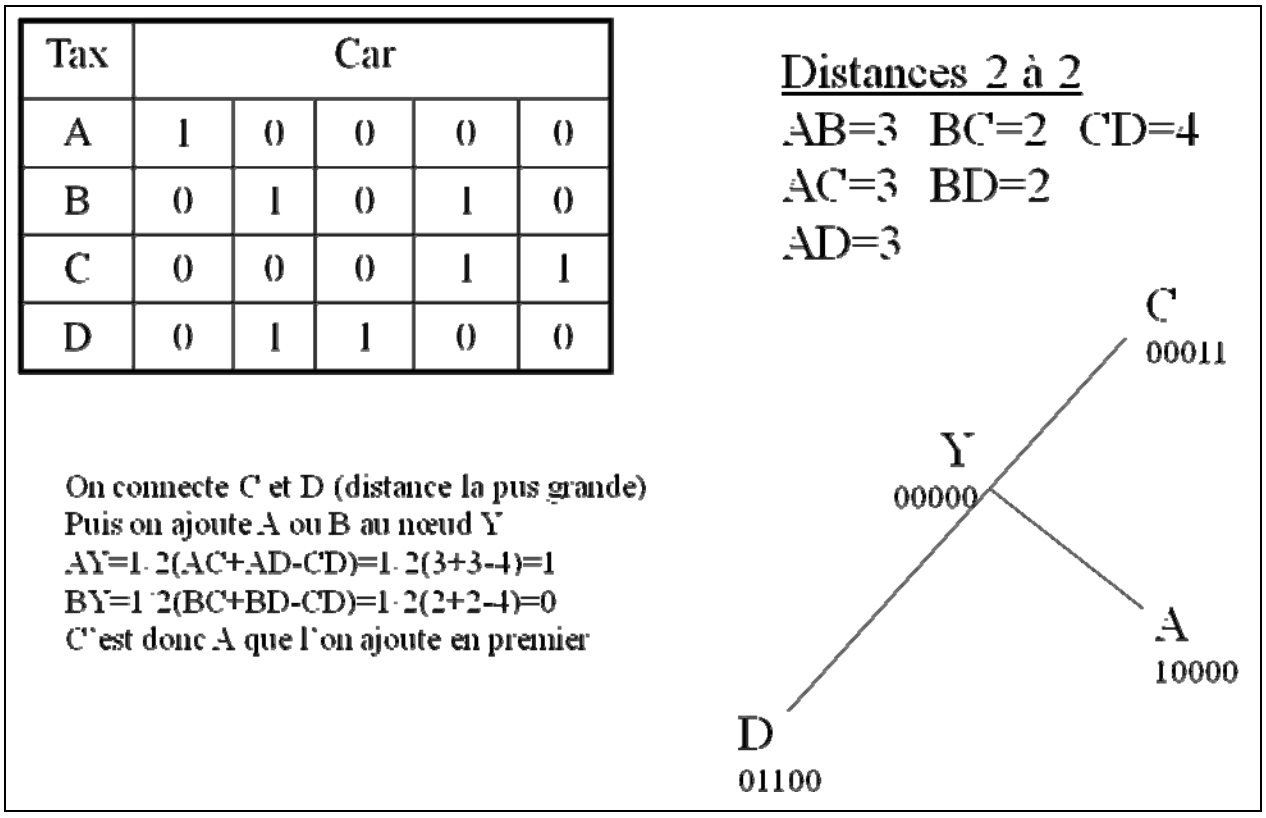


**Figure III- 1. Première étape de l'algorithme de Wagner: on prend les deux taxons les plus éloignés et on ajoute celui qui est le plus éloigné du nœud Y. L'état des caractères est alors défini pour ce nœud.**

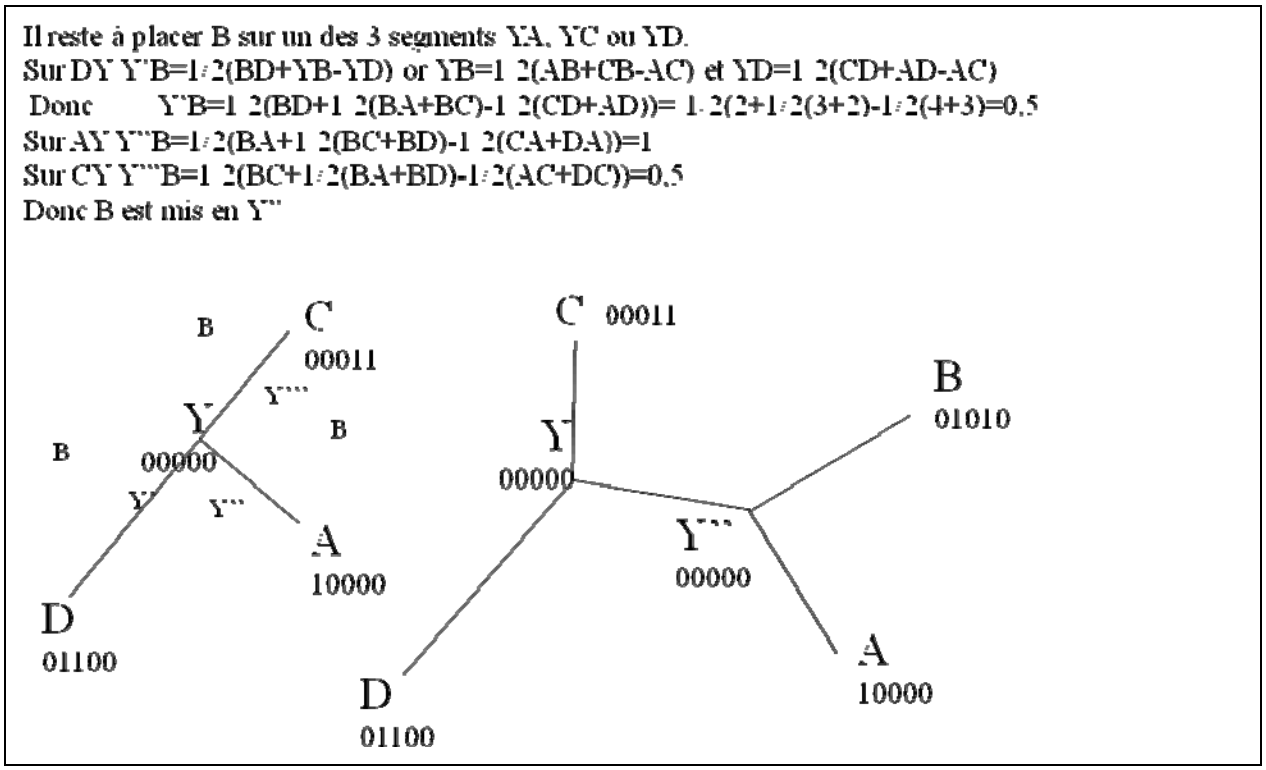


**Figure III-b 8. L'étape suivante de l'algorithme (Farris, 1971).**

**Branch swapping**

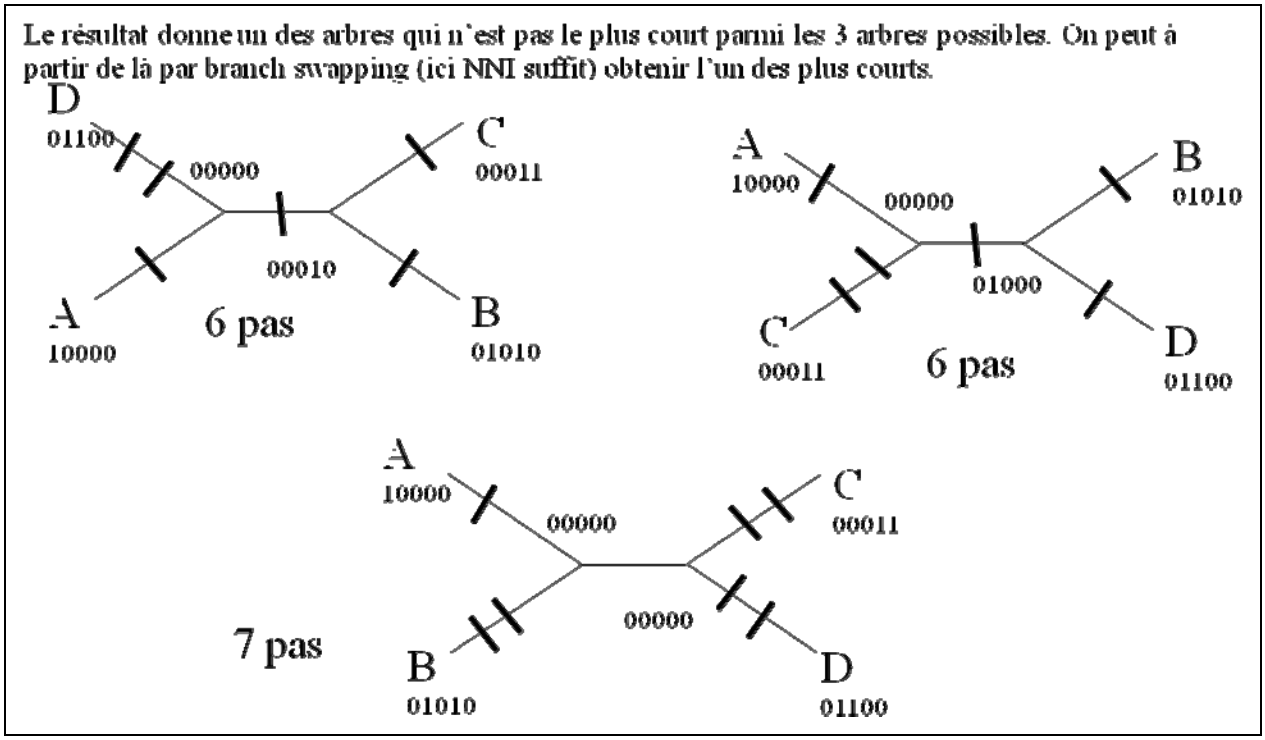


**Figure III-b 9. L'arbre trouvé par l'algorithme (AB),(CD) n'est pas forcément le plus court.**

En général , les algorithmes d'addition pas à pas ne trouvent pas l'arbre le plus court. Cet arbre est seulement parmi les moins longs. Il faut donc tester l'arbre initialement trouvé en lui faisant subir différents réarrangements qui sont appelés branch swapping (échange des branches). Les procédures pour cela peuvent être locales (Nearest Neighbor Interchanges de PAUP), échange du voisin le plus proche entre 4 taxons, global (Subtree Pruning and Regrafting de PAUP) où chaque sous arbre possible est retioré de l'arbre et réinséré à toutes les positions possibles, et le réarrangement pâr bissection et reconnexion (Tree Bissection Reconnexion de PAUP) où un arbre est coupé en deux le long d'une branche, donnant ainsi deux sous arbres qui sont ensuite reconnectés à toutes les branches possibles de l'arbre.

Comme dans le cas de la procédure du Branch and Bound, si l'obtention d'un arbre minimal nécessite le balayage d'un arbre plus coûteux en pas, l'arbre le plus parcimonieux n'est pas trouvé. Pour pallier à ce problème, il faut que les réarrangements s'appliquent également aux arbres non parcimonieux.

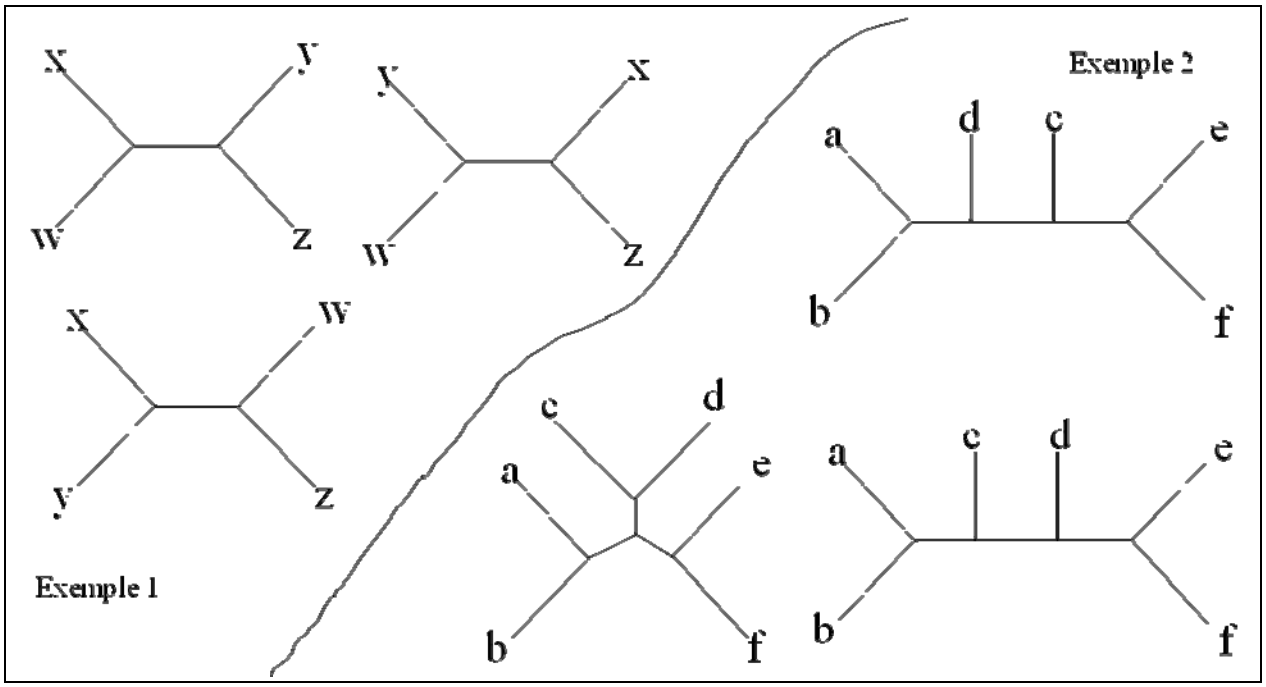


**Figure III-b 10. Branch swapping:réarrangement local(NNI)**

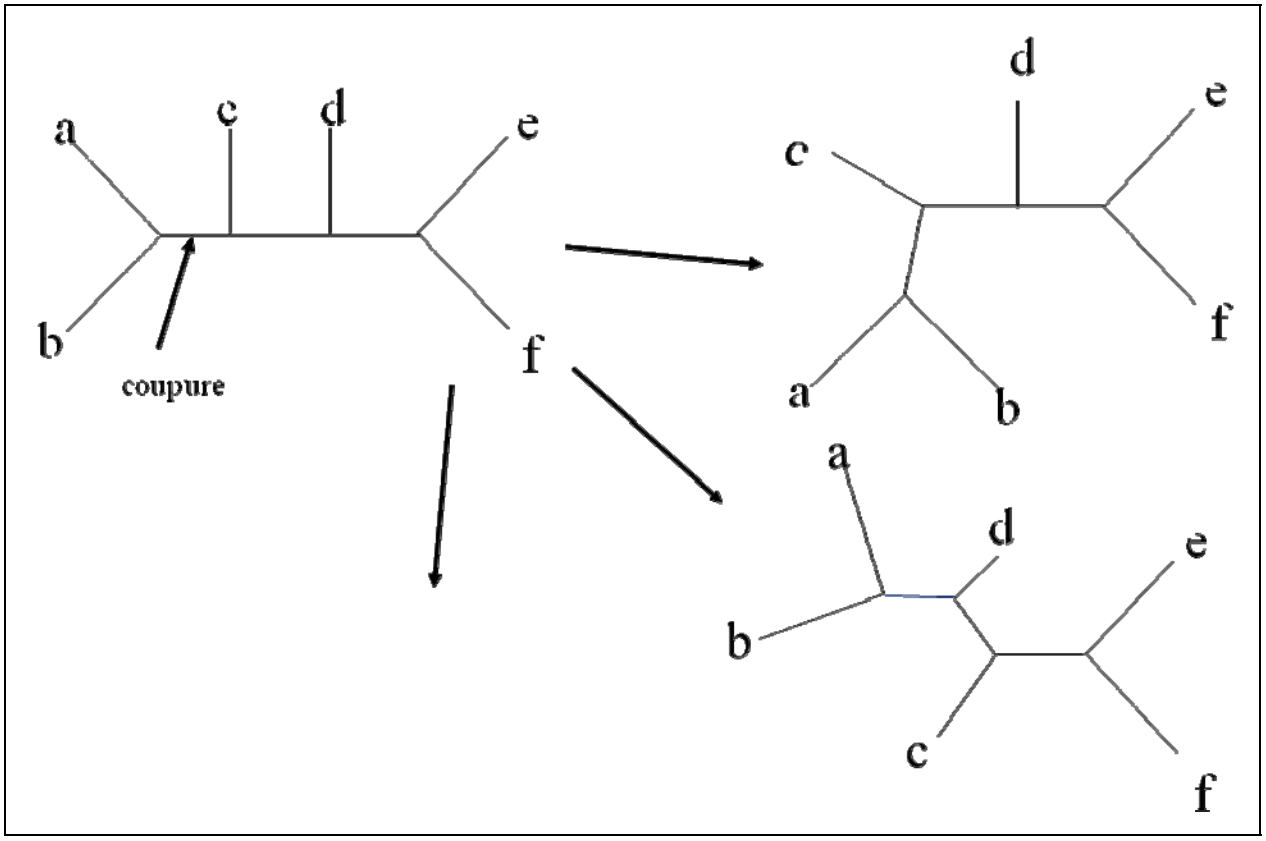


**Figure III-b 11. Branch swapping:réarrangement global(SPR).**

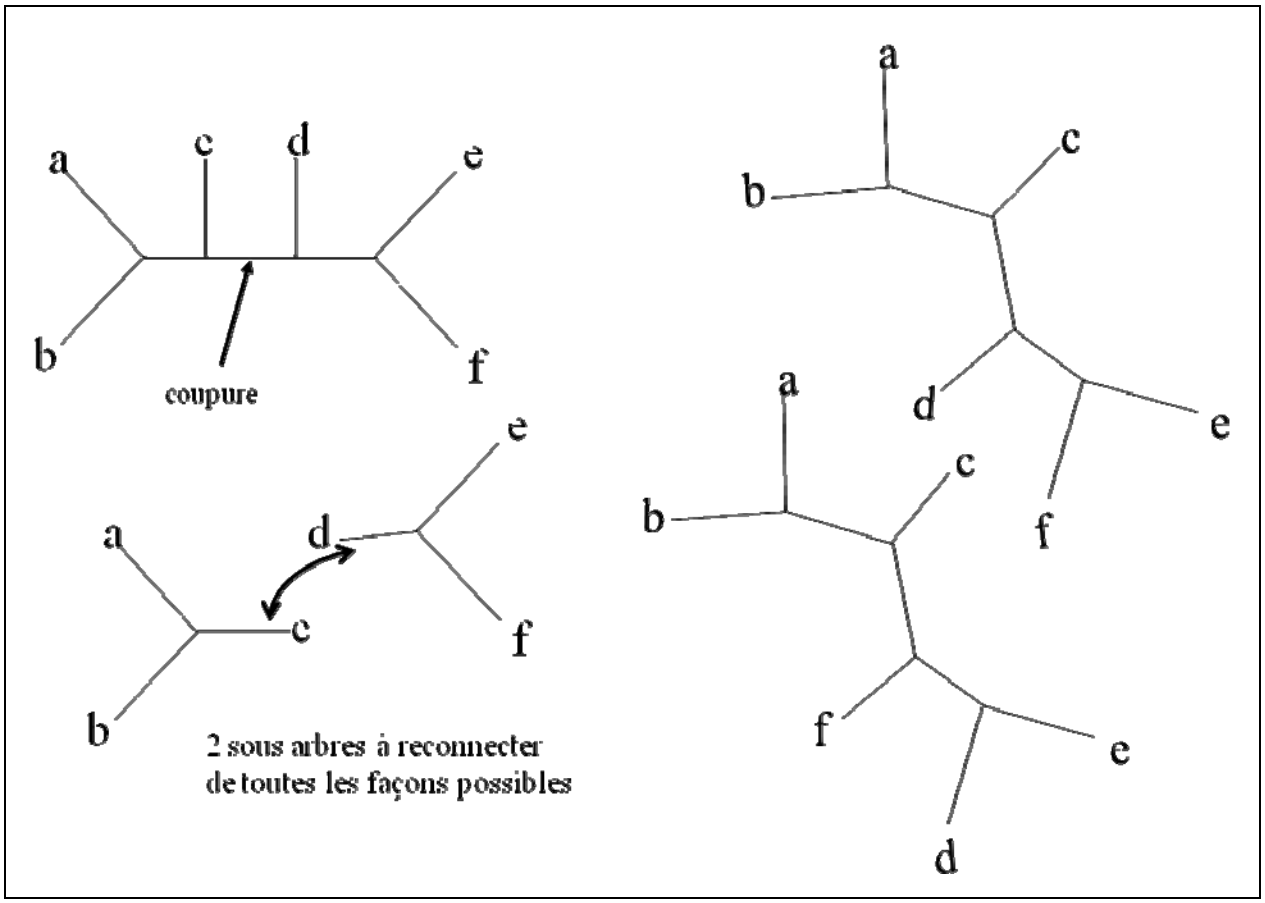


**Figure III-b 12. Branch swapping:réarrangement global(TBR).**

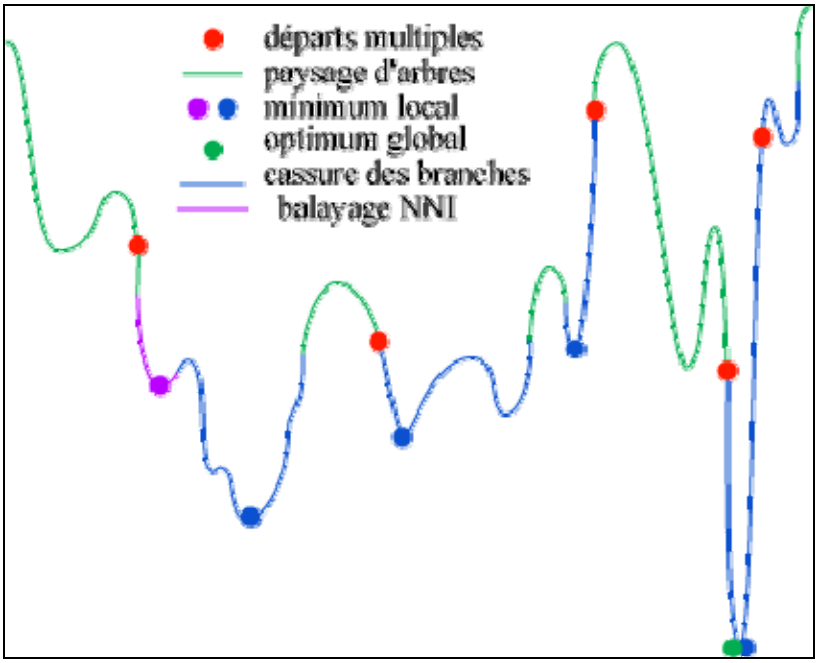


**Figure III-b 13. L'ensemble du paysage d'arbres. En partant de l'arbre représenté par le point rouge, on explore la zone bleue (clair et foncé représente l'espace exploré par deux méthodes différentes.**

## Analyse des résultats

### Retour aux caractères

#### Optimisation

Une fois l'arbre le plus parcimonieux obtenu on peut avec la méthode de parcimonie placer le long des branches les transformations de caractères. Certaines ne se sont produites qu'une fois et n'ont qu'une position possible. D'autres au contraire se rencontrent plusieurs fois et on peut choisir de favoriser les convergences (DELTRAN, :delayed transformations ou les réversions (ACCTRAN, accelerated transformations). Ce choix qui ne modifie pas la topologie de l'arbre ni sa longueur influe néanmoins sur la longueur de chaque branche et donc sur l'histoire évolutive des caractères et des taxons. L'analyse de parcimonie seule ne permet pas de trancher entre les deux modèles. Il faut des arguments extérieurs.

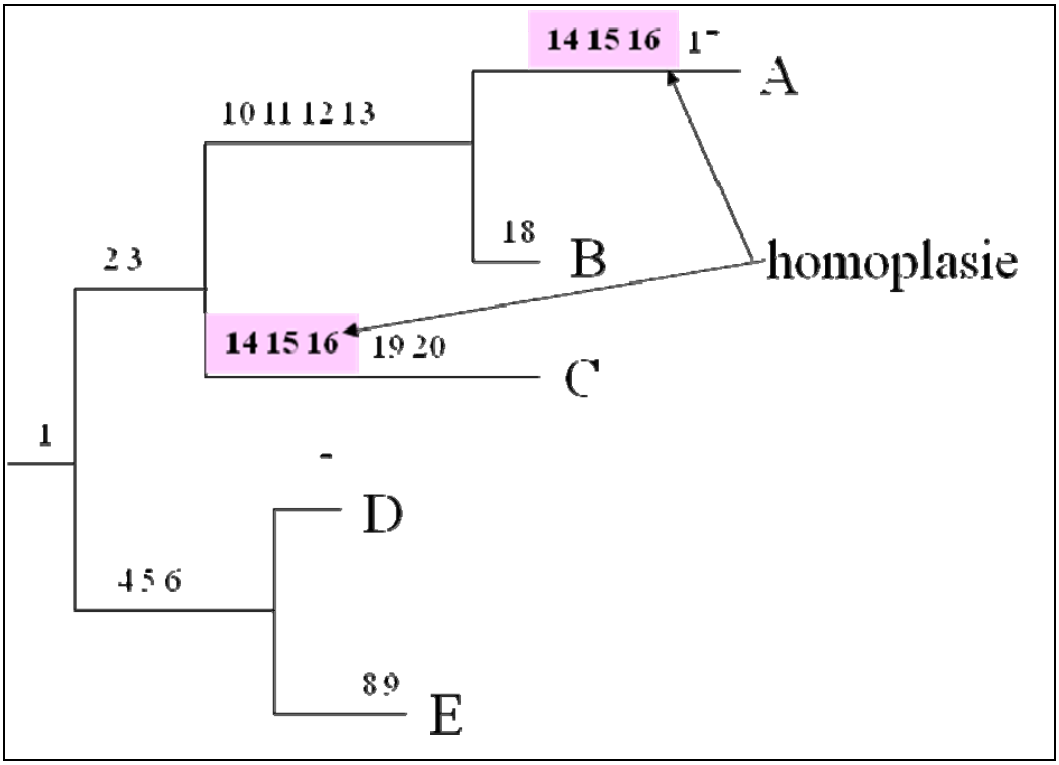


**Figure III-b 14. DELayed TRANsformations : Il y a convergence sur les branches de A et C pour les caractères 14 15 et 16.**

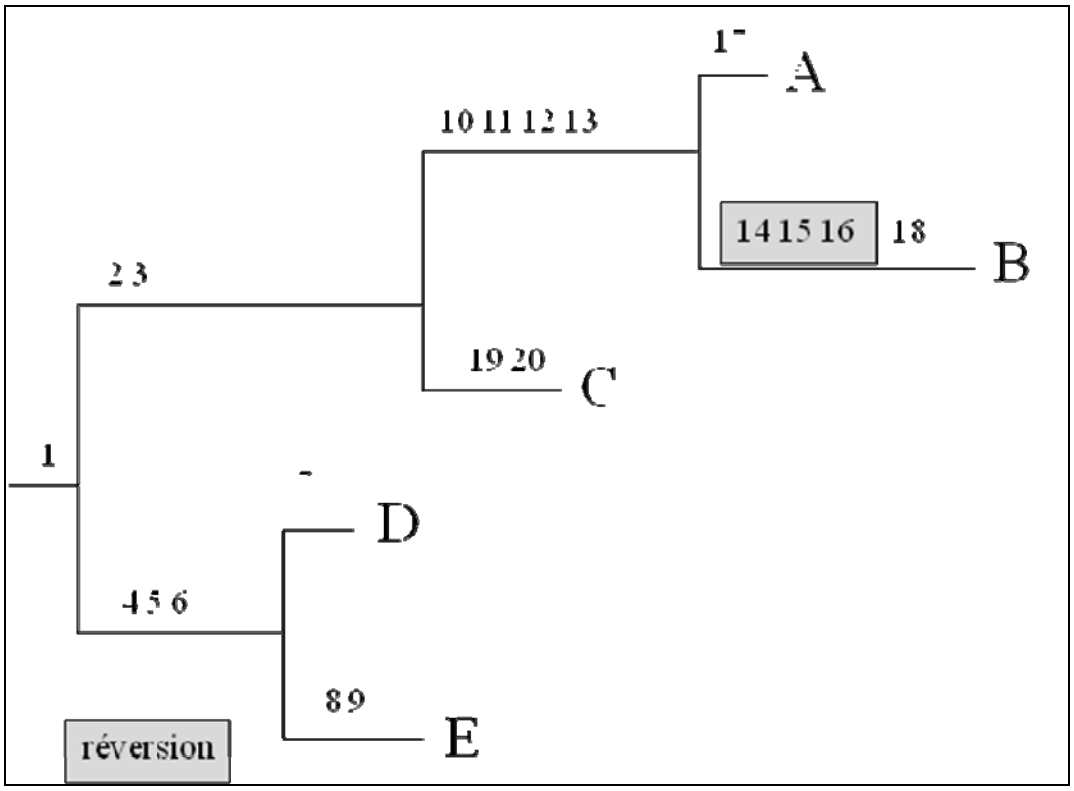


**Figure III-b 15. ACCelerated TRANsformation : La réversion est préférée à la convergence pour rendre compte de l'homoplasie, elle concerne trois caractères: 14, 15 et 16.**

**Pondération**

Je ne vous détaillerai pas ici le codage des états de caractères polymorphiques avec des transformations ordonnées (au moins partiellement) qui s'appliquent plutôt aux caractères morphologiques.

Dans le cas des séquences nucléotidiques on peut tenir compte de différents points.

- Lorsque pour un nucléotide donné on trouve les 4 états possibles distribués au hasard on peut estimer que ce site n'apporte que du bruit et l'éliminer ou lui donner un poids moindre que d'autres qui présenteraient seulement 2 états
- Une position peut ne pas être indépendante d'une autre (cas des régions palindromiques de l'ARN ribosomique par exemple) . Dans ce cas, elle n'apporte pas plus d'information que la première.

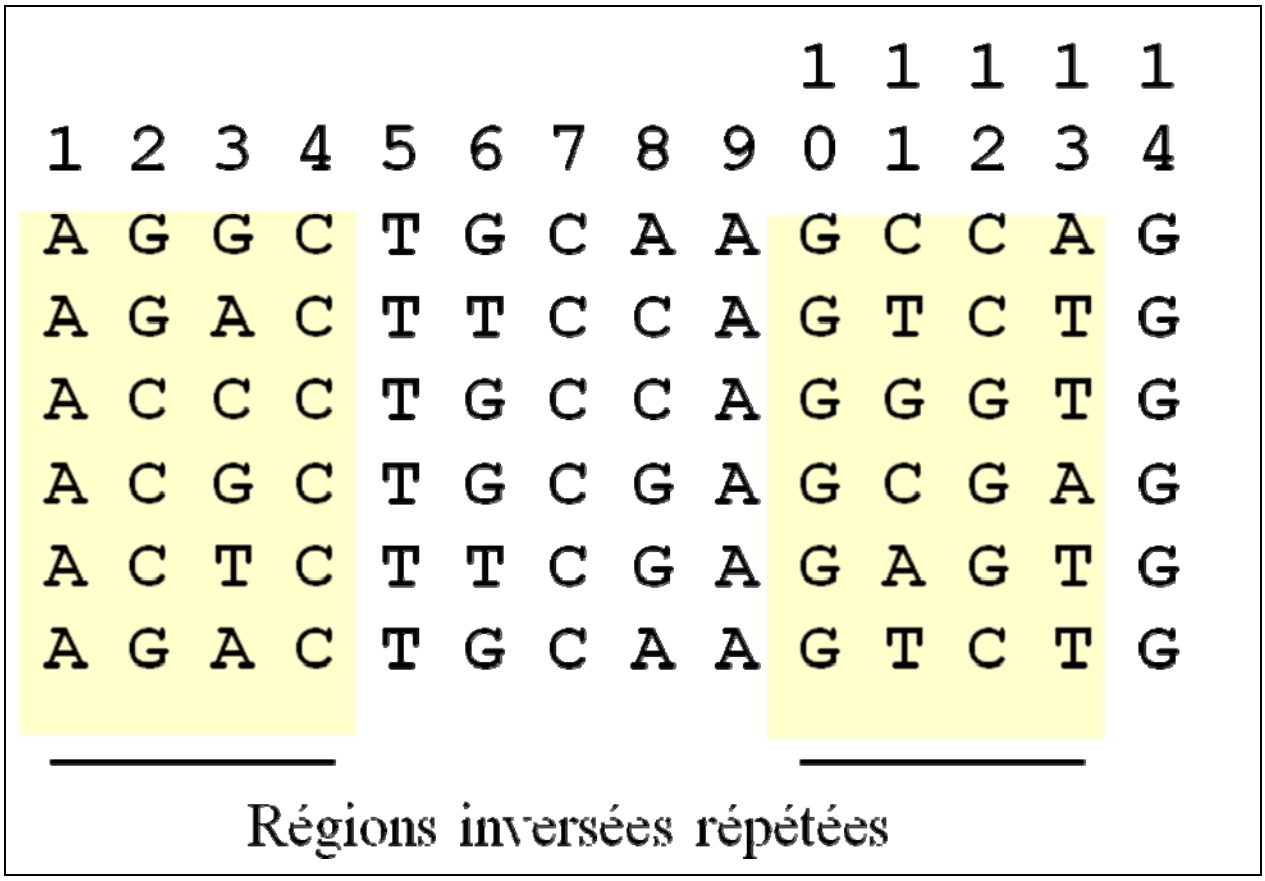


**Figure III-b 16. Les deux régions jaunes ne sont pas indépendantes et peuvent être pondérées moins que les zones blanches.**

- Dans une séquence codante la position du nucléotide dans le triplet a une importance sur sa variabilité et le rôle que va jouer une mutation sur ce nucléotide : dans le gène rbcL pour 39 espèces de gymnospermes on trouve :

| | position 1 | position 2 | position 3 | toutes les positions |
|---|---|---|---|---|
| nombre total de sites | 416 | 413 | 409 | 1238 |
| sites variables | 81 | 44 | 347 | 472 |
| sites informationnels | 52 | 20 | 293 | 365 |

- On peut également décider après le calcul d'un arbre et l'examen des critères (voir ci-dessus) choisir de donner plus de poids aux caractères qui présentent le moins de bruit, celui-ci étant estimé selon les cas par le nombre de pas, le CI, le RI ou le RC.
- On a trois types de transformations : les transitions, les transversions et les in/del. On peut choisir de pondérer en raison inverse de la fréquence de ces différents changements, considérant que le plus souvent observé est le plus facile et celui qui contient le plus de bruit.

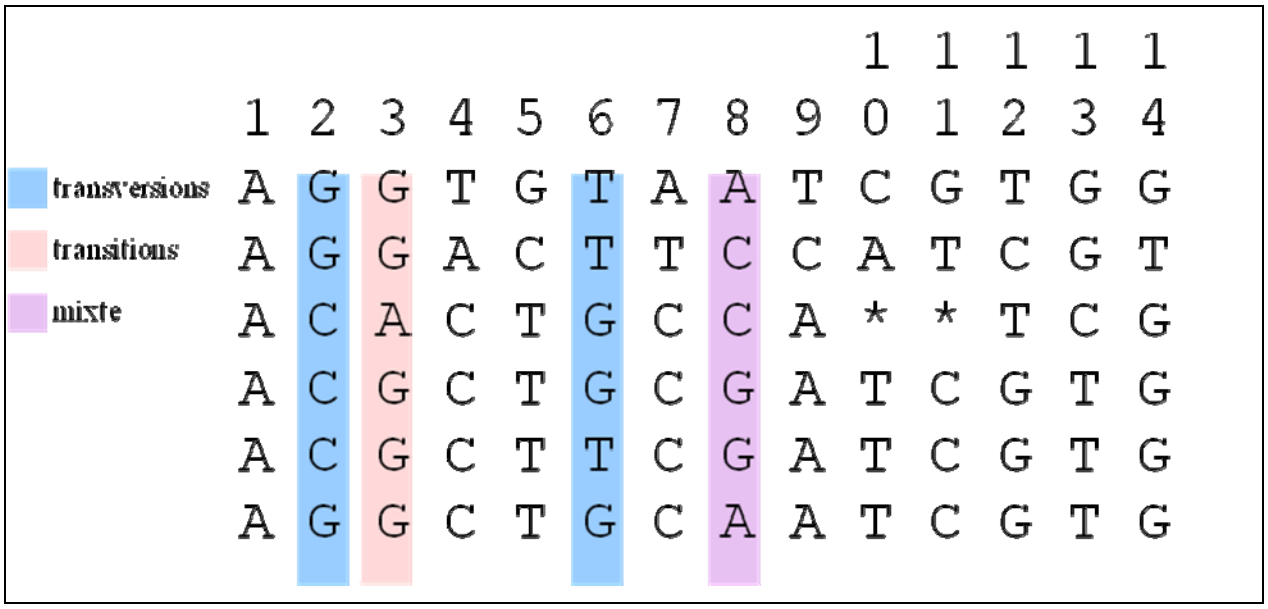


**Figure III-b 17. La pondération différente entre transitions et transversions entraîne des valeurs différentes affectées aux différents changements d'états du caractère 8. Il n'existe plus une valeur unique affectée à ce caractère.**

**Saturation**

Tous les changements n'ayant pas la même vitesse, il est possible que certains types rapides aient tellement tourné que la distribution des états de caractère soit aléatoire alors que pour d'autres types, l'état actuel soit le reflet du temps écoulé depuis la séparation des taxons.

On peut suspecter une telle divergence entre transitions (rapides) et transversions (plus lentes), ou dans une séquence codante entre les positions 1 et 2 et la position 3 du codon, cette dernière entraînant plus souvent des changements synonymes ou pas de changement d'acide aminé.

### Principe de la comparaison

Pour un ensemble de taxons on construit deux tableaux de distance, l'un avec les distances 2 à 2 mesurées uniquement en transition (par exemple) et l'autre avec les distances mesurées uniquement avec les transversions.
Chaque couple est donc caractérisé par deux distances qui déterminent un point dans un plan avec en abscisse lea distance en transition et en ordonnée la distance en transversion.
Dans le plan chaque couple est caractérisé par un point.
Si les dezux distances apportent la même information, on attend que le nuage de points constitue à peu prtès une droite, passant par l'origine ; si de plus la vitesse pour les deux distance est la même, la droite doit avoir une inclinaison à45°.
Au contraire, si une distance est satuyrée et pas l'autre , la première va attendre un maximum alors que la seconde va continuer à augmenter. La courbe sera asymptotique à une parallèle à l'axe représentant les distances non saturées.

| Ve / Si | A | B | C | D |
|---|---|---|---|---|
| A | | 4 | 8 | 16 |
| B | 6 | | 20 | 28 |
| C | 16 | 41 | | 32 |
| D | 27 | 40 | 42 | |

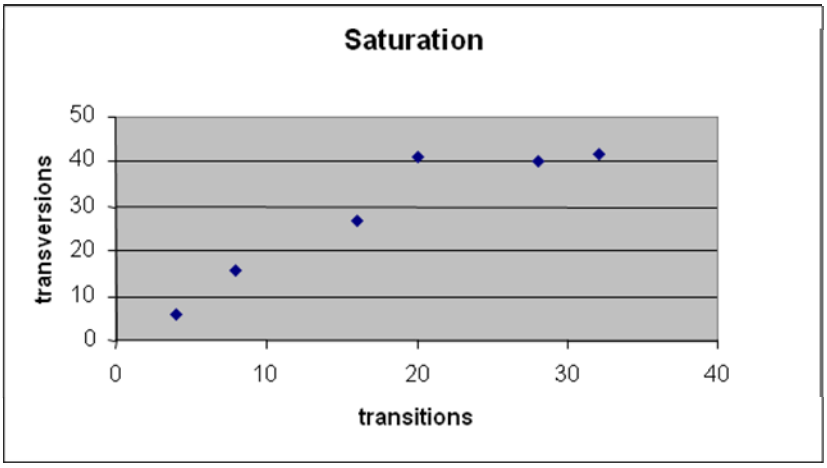


**Figure III-b 18. Saturation: principe. A gauche la matrice comporte les distances en transition dans la partie inférieure et celles en transversion dans la partie supérieure. Dans cet exemple, les transitions sont saturées.**

## Exemples

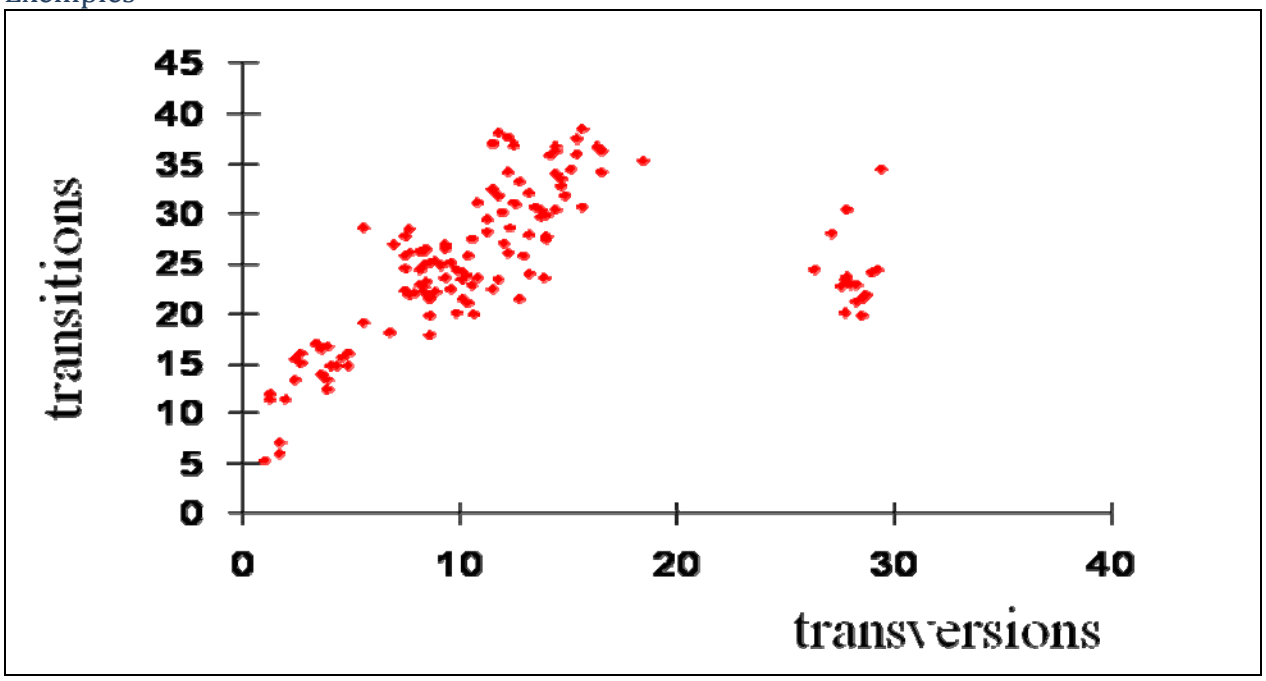


**Figure III-b 19. Comparaison de la vitesse d'évolution en transitions et en transversions pour la position 3 des codons du gène rbcL : il y a saturation.**

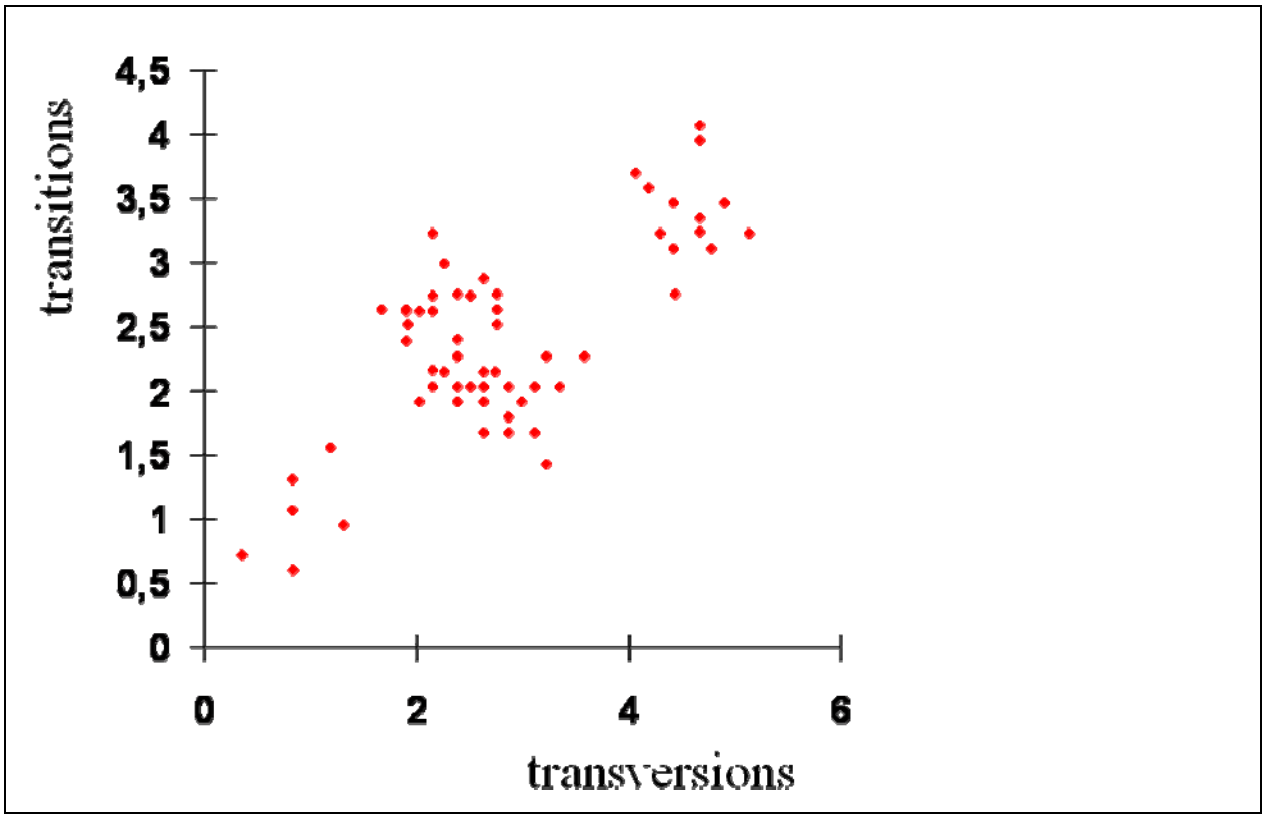


**Figure III-b 20. Comparaison de la vitesse d'évolution en transitions et en transversions pour les positions 1 et 2 des codons du gène RBCL. Ici il n'y a pas de saturation.**

## Arbres de consensus

Il arrive que le programme trouve plusieurs arbres également parcimonieux. Comment choisir ? en l'absence d'arguments externes on peut décider de garder seulement dans l'arbre retenu les nœuds communs à tous les arbres. Ce que l'on appelle l'arbre de consensus. Les critères pour tracer cet arbre varient selon les auteurs.

- arbre de consensus strict ou arbre de Nelson ne garde que les nœuds communs à tous les arbres, les points de conflit sont représentés par des multifurcations. C'est la plus utilisée cependant elle ne rend pas compte que la monophylie (BD) n'est pas observée.
- arbre de consensus d'Adams-2 qui consiste à garder le consensus des regroupements à partir de la racine entre deux ou plus d'arbres. Ici l'arbre 1 donne les groupes (A) et (B,C,D et E) alors que l'arbre 2 donne (A, B et C) et (D et E) . L'arbre consensus va donc donner (A) (B et C) (D et E).
- arbre de consensus majoritaire dans lequel on va garder tous les groupements qui sont rencontrés avec une fréquence au moins égale à un pourcentage choisi (>50%).
- arbre de consensus semi strict garde tous les nœuds qui ne sont contredits par aucun arbre (cette méthode s'applique dans le cas d'arbres présentant des polytomies, sinon cela revient à un consensus strict).

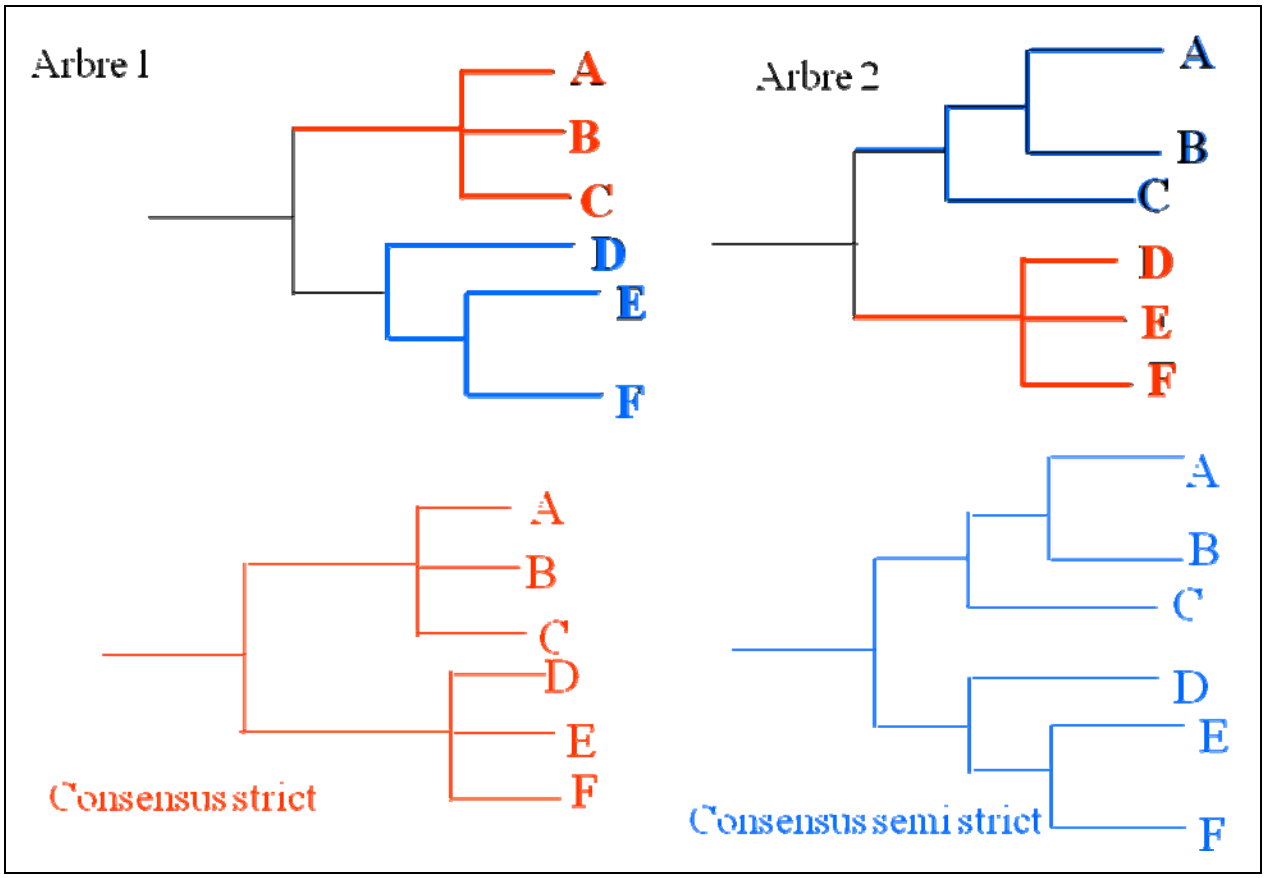


**Figure III-b 21. Arbres de consensus strict ou semistrict, la différence n'existe que lorsqu'on a des polyfurcations.**

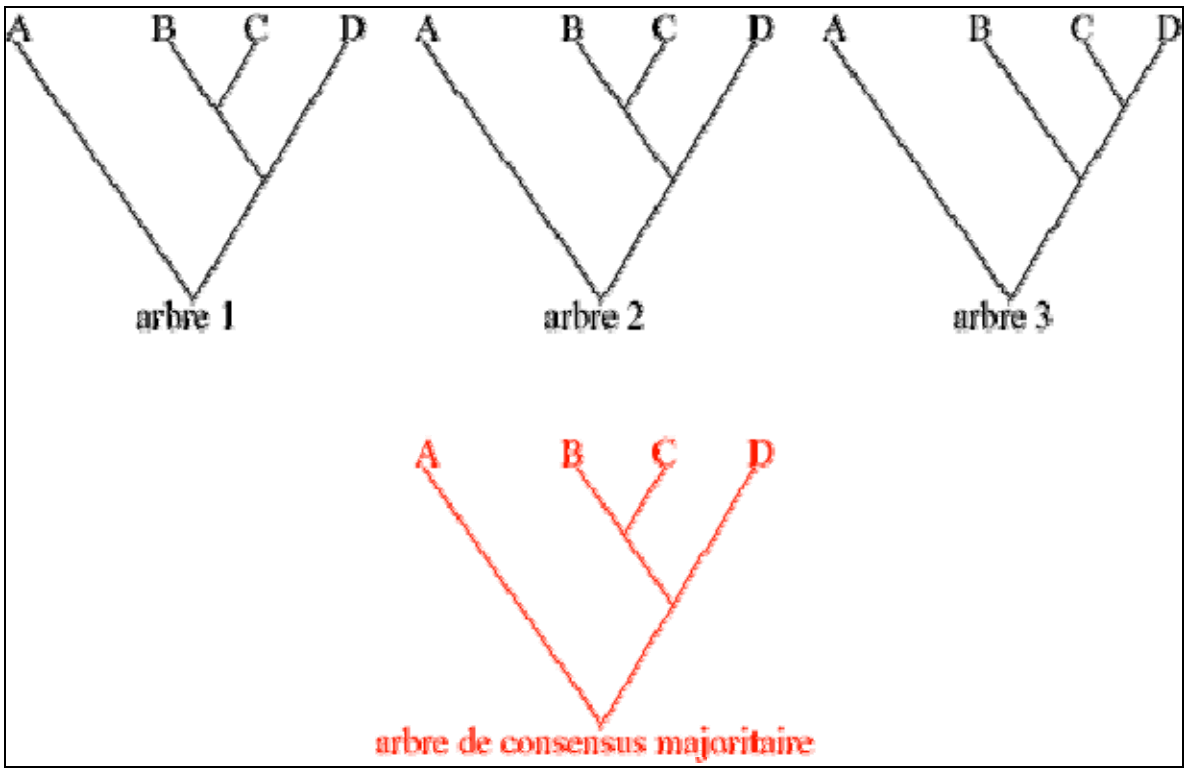


**Figure III-b 22. Arbre de consensus majoritaire.**

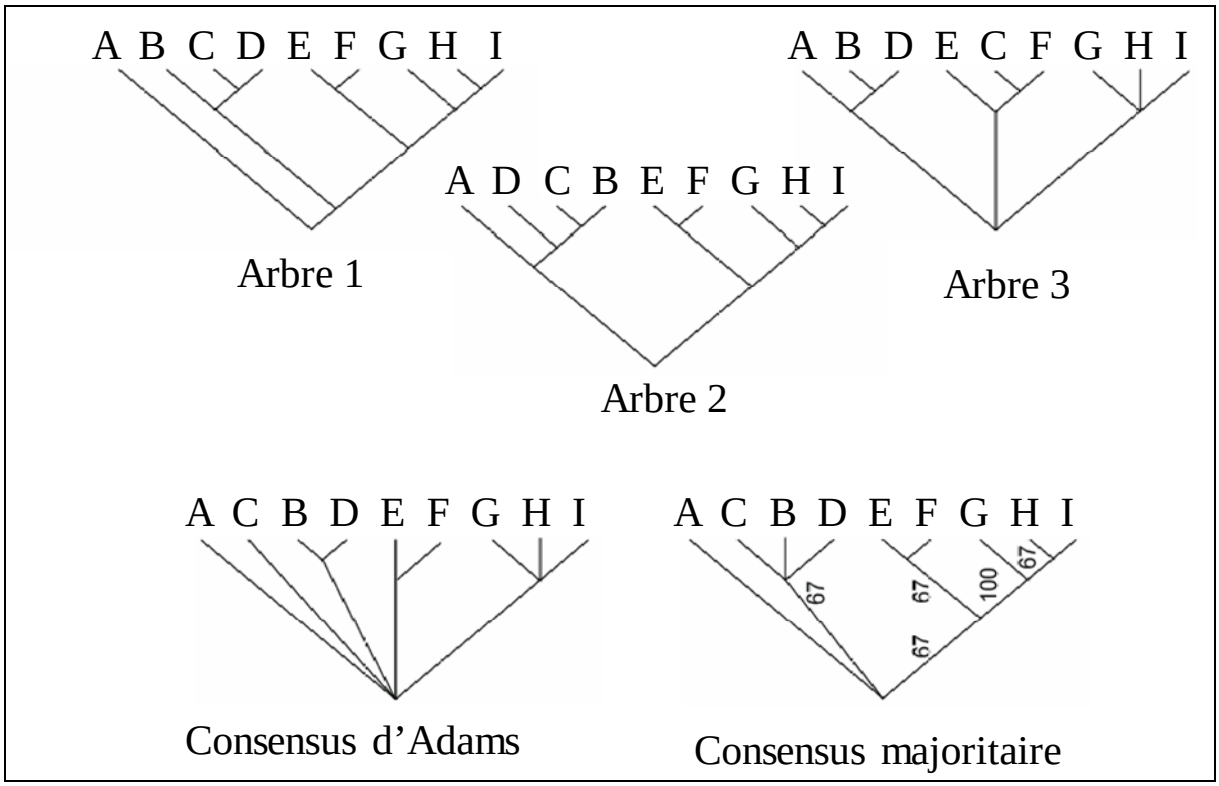


**Figure III-b 23. Comparaison entre consensus majoritaire et consensus d'Adams.**

## Tests de robustesse

### CI et RI mesure de l'homoplasie

Si pour construire un arbre on a utilisé 7 caractères, on peut imaginer un arbre où aucun caractère ne serait homoplasique : il n'y aurait que 7 changements de caractères soit 7 pas c'est la longueur minimum de l'arbre.
Si au contraire on reporte tous les changements sur les branches terminales on aura une longueur maximum de l'arbre ici 15 pas (cette valeur est déduite de la matrice de caractères).
L'arbre le plus court obtenu est de 9 pas.
Par définition son Indice de Consistance est

CI = 7/9 = 0,78
Et son Index de rétention :

RI = 15 – 9 / 15 – 7 = 0,50
Le CI varie de un peu plus de 0 jusqu'à 1 si l'arbre réel est le plus court possible alors que RI varie de 0 à 1.
La limite inférieure du CI n'est pas réellement 0, parfois on calcule un indice de consistance remis à l'échelle (rescaled index ou RC) qui n'est autre que CI diminué de sa valeur la plus petite (m/g) et divisé par son intervalle de variation (1 – m/g) pour qu'il puisse vraiment varier de 0 à 1.

$$RC = \frac{\frac{m}{s} - \frac{m}{g}}{1 - \frac{m}{g}} = \frac{\frac{m}{s} - \frac{m}{g}}{\frac{g-m}{g}} = \left(\frac{m}{s} - \frac{m}{g}\right)\left(\frac{g}{g-m}\right) = \frac{mg}{s(g-m)} - \frac{mg}{g(g-m)} = \frac{mg}{s(g-m)} - \frac{m}{g-m}$$

$$RC = \frac{m}{s}\left(\frac{g}{g-m} - \frac{s}{g-m}\right) = \frac{m}{s}\left(\frac{g-s}{g-m}\right)$$

Le programme donne la longueur réelle de l'arbre ainsi que les longueurs minimum et maximum. Dans le menu *Longueurs et mesures* on peut choisir de voir CI, RI et RC HI n'est autre que le complément à 1 de CI

La valeur du CI est fonction du nombre de taxons contenu dans le jeu de données. Une compilation d'un certain nombre d'arbres publiés à permis à Sanderson et Donoghue de trouver une équation empirique qui exprime cette variation et donne la valeur seuil pour un nombre donné de taxons au-delà de laquelle on peut penser que le CI a une signification (valeur non aléatoire).

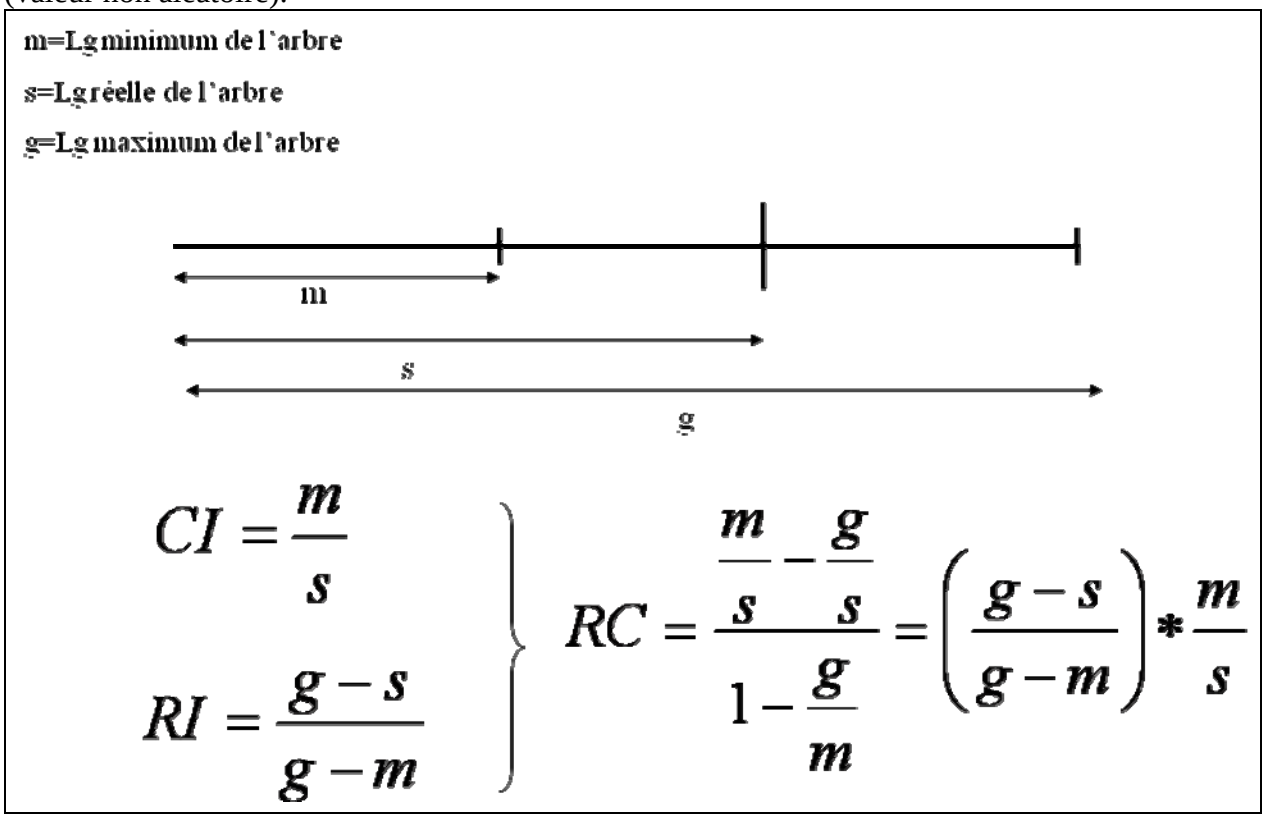


**Figure III-b 24. CI, RI et RC.**

Formule empirique NT: nb de taxa

$$CI = 0{,}90 - 0{,}022NT + 0{,}000213(NT)^2$$

Sanderson, Donoghue (1989)Patterns of variation in levels in levels of homoplasy. Evolution 43 pp1781-95

| Nb. Tax. | CI |
|---|---|
| 15 | 0.6179 |
| 16 | 0.6025 |
| 17 | 0.5876 |
| 18 | 0.5730 |
| 19 | 0.5589 |
| 20 | 0.5452 |
| 21 | 0.5319 |
| 22 | 0.5191 |
| 23 | 0.5067 |
| 24 | 0.4947 |
| 25 | 0.4831 |
| 26 | 0.4720 |
| 27 | 0.4613 |

| Nb. Tax. | CI |
|---|---|
| 28 | 0.4510 |
| 29 | 0.4411 |
| 30 | 0.4317 |
| 31 | 0.4227 |
| 32 | 0.4140 |
| 33 | .0.4060 |
| 34 | 0.3980 |
| 35 | 0.3910 |
| 36 | 0.3840 |
| 37 | 0.3776 |
| 38 | 0.3716 |
| 39 | 0.3660 |
| 40 | 0.3608 |

**Figure III-b 25. Variation de l'indice de consistance significatif en fonction du nombre de taxa.**

**Bootstrap et jackknife**

Ce sont deux méthodes de ré échantillonnage des sites qui vont simuler des matrices nouvelles.

Dans le Bootstrap, on tire autant de sites qu'il y en a dans le lot de séquences initiales, le tirage étant non exhaustif. Dans la procédure du Jackknife le tirage est sans remise et on ne tire qu'un certain pourcentage des sites disponibles. Cela revient par l'une ou l'autre des 2 méthodes à une pondération au hasard des sites.

Pour chacune des matrices nouvelles, on construit l'arbre (avec une méthode de distance ou de parcimonie). Avec 100 tirages on génère 100 matrices différentes et 100 arbres.

A partir de ces 100 arbres on trace le consensus majoritaire : chaque branche est affectée d'un % qui correspond à la fraction des arbres où cette branche a été retrouvée. Plus le % est élevé plus la branche est fréquemment rencontrée et donc plus elle est robuste (robuste ne signifie pas forcément sure).

On peut faire également du Jackknife d'espèces : si le lot d'origine contient n taxons, on va effectuer un nouvel échantillonnage des taxons en n'en prenant que n' (<n). De cette façon on peut détecter si un taxon introduit un biais important dans l'arbre.

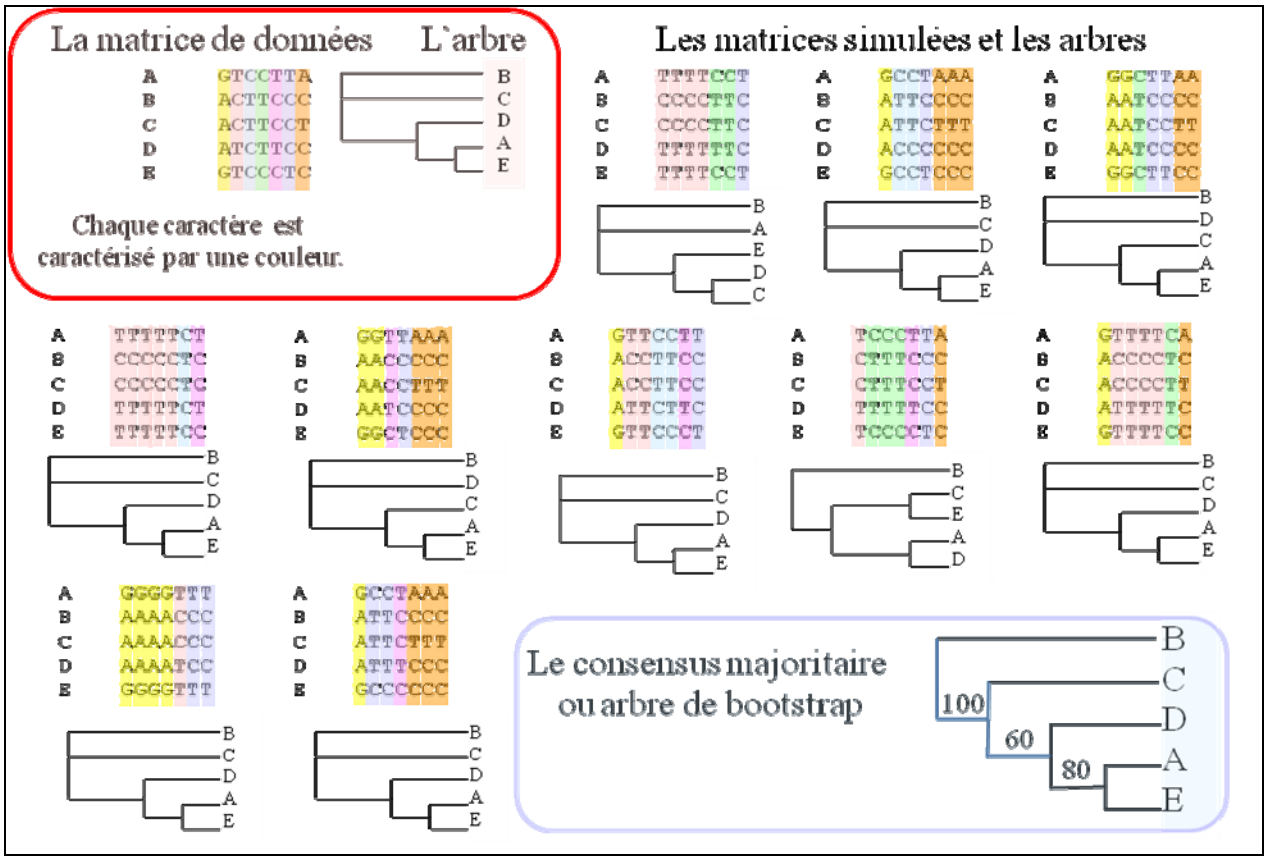


**Figure III-b 26. Le test de bootstrap**

### Indice de Bremer

On cherche de combien de pas il faut rallonger l'arbre pour effacer un nœud. Plus il en faut, plus ce nœud est robuste.
En pratique on cherche tous les arbres les plus parcimonieux qui ont une longueur de n pas. On cherche ensuite tous les arbres qui ont une longueur inférieure ou égale à (n+10) pas. Enfin en utilisant la fonction filtre du menu trees on garde tous les arbres inférieurs ou égaux à (n+1) pas, on en fait le consensus strict que l'on compare avec le consensus strict des arbres de longueur n. Si un (ou plusieurs nœuds ont disparu ils prennent la valeur D=1. On recommence pour toutes les valeurs jusqu'à n+10. Au delà on considère que les nœuds qui restent sont robustes.

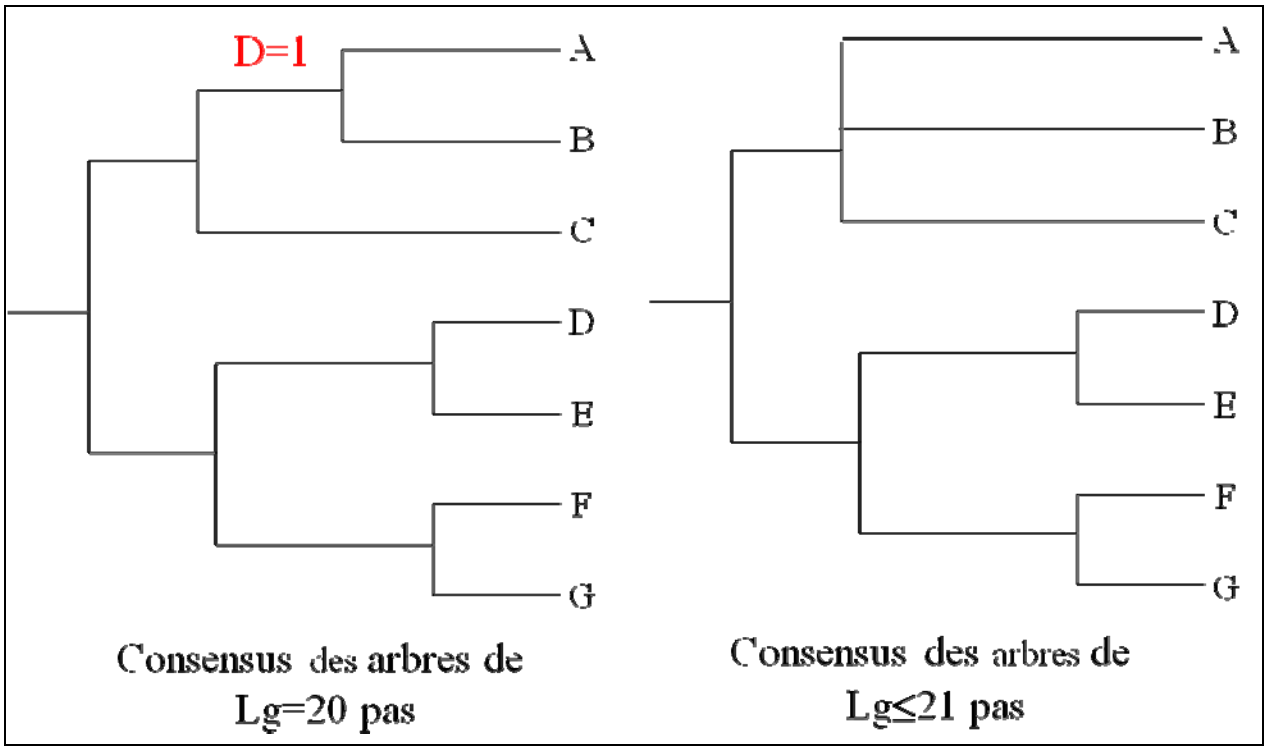


**Figure III-b 27. Calcul de l'indice de Bremer.**

## Congruence des données

Il s'agit de vérifier si plusieurs jeux de données portent le même message phylogénétique ou des messages contradictoires. On parle alors d'inconsistance. Si deux jeux de données sont consistants on peut alors les concaténer dans l'espoir d'un signal phylogénétique plus net et d'un arbre mieux résolu.

1. A partir de chacun des deux jeux de données X et Y (comportant exactement les mêmes taxons) sont cherchés les arbres optimaux dont on calcule la longueur Lx et Ly.
2. Les deux jeux de données (comportant exactement les mêmes taxons) sont réunis. Le ou les arbres les plus parcimonieux sont cherchés et leur longueur déterminée Lz
3. Il faut déterminer si Lx + Ly est significativement différent de Lz (inconsistence) ou pas.
4. Pour ce test statistique on effectue des simulations. Des caractères sont tirés dans cet ensemble, et en parallèle des caractères sont tirés dans un jeu et dans l'autre. On a donc trois séries de tirages (qui représentent autant de simulations) qui vont permettre de calculer pour chaque groupe de trois tirage une valeur de (Lx+Ly-Lz).
5. On va ensuite comparer la différence observée pour les distances vraies à la distribution obtenue sur n (souvent 100) simulations.

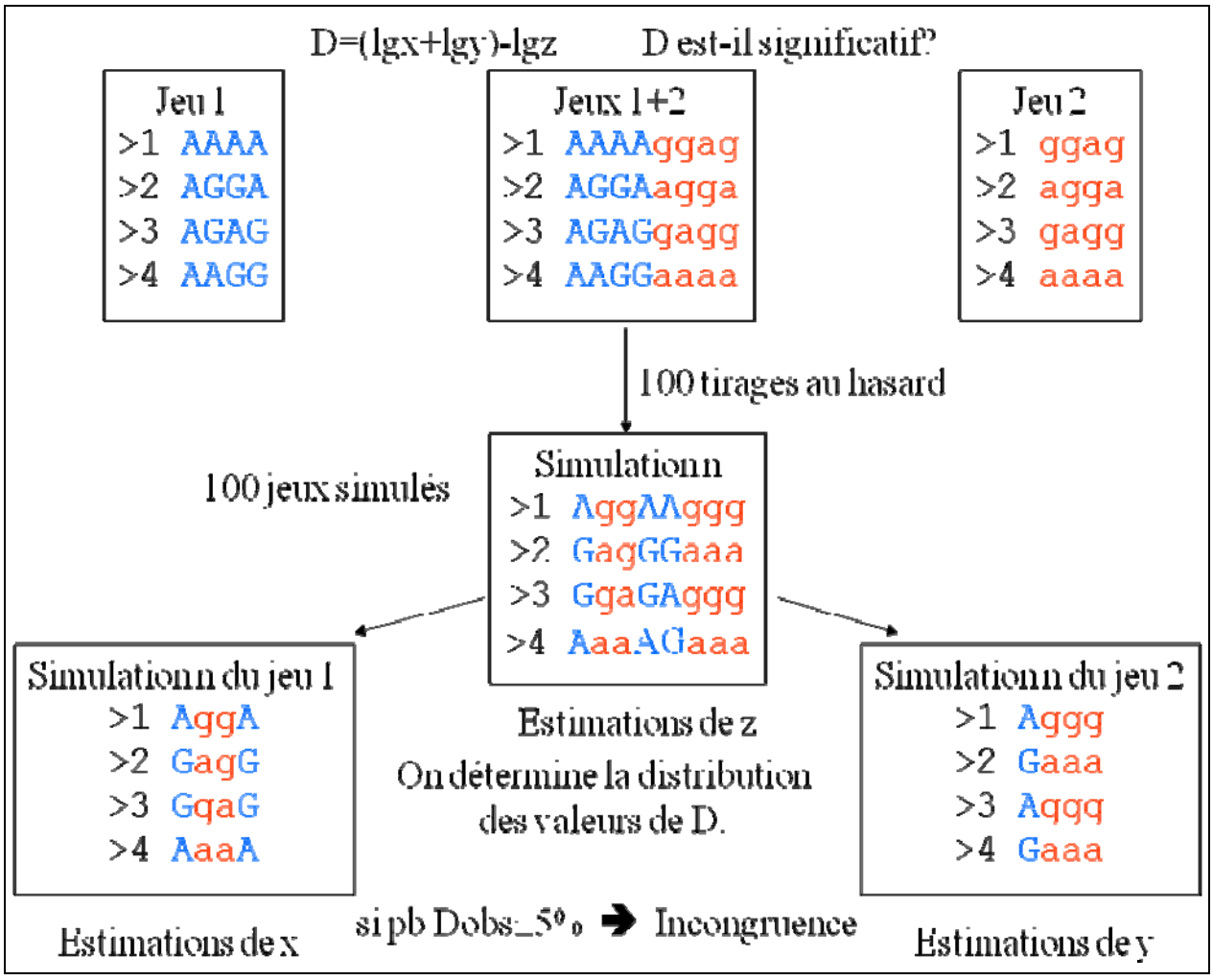


**Figure III-b 28. Réalisation du test de congruence ou test ILD (Incongruence Length Difference)**