

Parcimonie de Fitch-Hartigan pour un arbre donné

- La parcimonie totale est la somme des parcimonies de chaque site ; on raisonne indépendamment sur chacun.
- A chaque nœud x est attaché l'ensemble des valeurs possibles $C(x)$ réalisant la parcimonie optimale $P(x)$
- Soit $g(x)$ et $d(x)$ les fils droit et gauche de x , on a la récurrence :

$$\begin{array}{ll} \text{Si } g(x) \text{ et } d(x), & \text{si } C(g(x)) \cap C(d(x)) \neq \emptyset \\ & \text{alors } C(x) = C(g(x)) \cap C(d(x)) \\ & P(x) = P(g(x)) + P(d(x)) \\ & \text{sinon } C(x) = C(g(x)) \cup C(d(x)) \\ & P(x) = 1 + P(g(x)) + P(d(x)) \end{array}$$

Sinon $P(x) = 0$ et $C(x) = \{\text{valeur de } x\}$

- Le résultat est indépendant de la position de la racine. Le calcul est en $O(ns|\Sigma|)$ (n = nombre de séquences ; s = nombre de sites). Et on peut réaliser les unions et intersections au niveau bit.
- Ce parcours est postorder. Un parcours preorder permet d'assigner aux nœuds une valeur réalisant l'optimum.

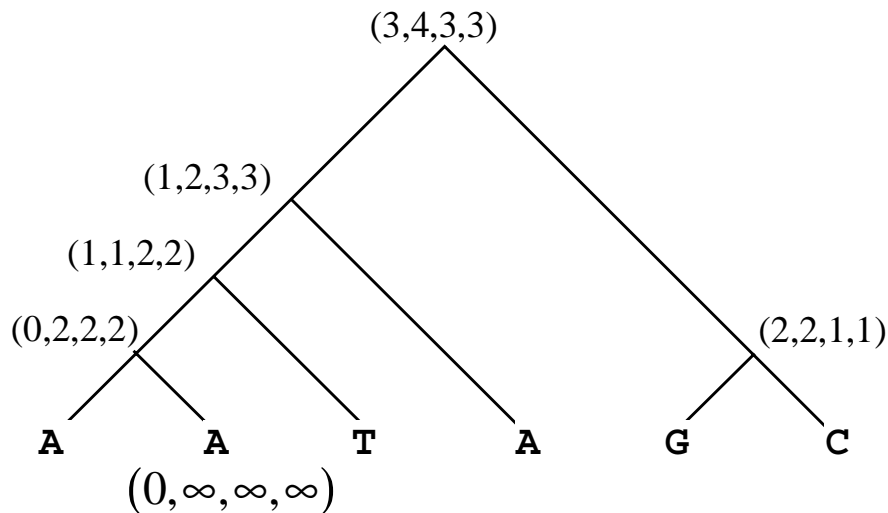
Parcimonie de Sankoff pour un arbre donné

- Le poids des substitutions n'est pas unitaire mais est donné par une matrice de coûts : S_{XY}
- A chaque nœud i est attaché le vecteur (A_i, T_i, G_i, C_i) qui représente le minimum de parcimonie en supposant que le nœud a pour valeur A, T, G, resp. C.

- Soit le nœud 0, père de 1 et 2. On a la formule de récurrence :

$$A_0 = \text{Min}(A_1 + A_2, S_{AT} + A_1 + T_2, S_{AT} + T_1 + S_{AG} + G_2, \dots)$$

$$T_0 = \text{Min}(T_1 + T_2, S_{TA} + A_1 + T_2, 2S_{TA} + A_1 + A_2, \dots)$$



Coût unitaire. La parcimonie est 3.

Le résultat est indépendant de la position de la racine lorsque S est symétrique. Le calcul est en $O(ns|\Sigma|^3)$.

Recherche de tous les meilleurs arbres de duplication simple (arches) dans le cas des minisatellites

- Il y a un seul site
- L'alphabet est constitué des variants
- On dispose d'une matrice de coût de substitution entre variants
- L'algo de distance Elemento&Gascuel (CPM'03) est directement applicable, mais la parcimonie est plus cohérente avec l'alignement
- On va combiner algorithme de Sankoff et EG.

- A chaque interval $[p, q]$ est attaché $(A_{pq}, B_{pq}, C_{pq}, D_{pq}, \dots)$, chaque X_{pq} contient quatre informations :
 - (1) PX_{pq} la valeur de parcimonie optimale pour X
 - (2) MX_{pq} l'entier m^* de $[p, q - 1]$ réalisant cet optimum
 - (3) GX_{pq} le caractère de $[p, m^*]$ réalisant l'optimum
 - (4) DX_{pq} le caractère de $[m^* + 1, q]$ réalisant l'optimum
- On calcule récursivement ces valeurs pour $[p, q]$ de taille croissante :

$$PX_{pq} = \text{Max}_{m \in [p, q-1], Y \in \Sigma, Z \in \Sigma} (PY_{pm} + PZ_{(m+1)q} + S_{XY} + S_{XZ})$$

- Un arbre optimal est composé de sous-arbre optimaux ; la méthode assure d'avoir au moins une solution optimale pour chaque interval.
- On s'attend à avoir plusieurs solutions optimales, pour plusieurs caractères et plusieurs positions de la racine. Les calculs sont pour une part redondants, mais on ne le sait qu'à la fin (?).
- L'algo est en $O(n^3 |\Sigma|^3)$.