TD Assemblage 1 - L3 Gr.C

Josua Philippot

3 avril 2020

Exercice 1

Soit les familles suivantes :
F1 = ACCTGAG
F2 = TGCATTGC
F3 = GCAGACC
F4 = AGCAAT
F5 = CAATG

Exécutons l'algorithme glouton vu en cours sur ces familles :

Etape 1 :

	F1	F2	F3	F4	F5
F1		0	1	2	0
F2	0		2	0	1
F3	3	0		0	1
F4	0	0	0		4
F5	0	2	1	0	

On remarque que le plus gros chevauchement est entre F4 et F5 en effet overlap(F4,F5) = 3, celui-ci fait référence à la séquence 'CAAT' qui apparaît en fin de la chaîne F4 et au début de F5.

On obtient donc la concaténation $F_{4,5}$: AGCAATG

Etape 2 :

	F1	F2	F3	$F_{4,5}$
F1		0	1	2
F2	0		2	0
F3	3	0		0
$F_{4,5}$	0	2	1	

On remarque que le plus gros chevauchement est entre F3 et F1, en effet overlap(F3,F1) = 3, avec la séquence 'ACC' qui apparaît en fin de la chaîne F3 et au début de F1.

On obtient donc par concaténation $F_{3,1}$: GCAGACCTGAG

Etape 3 :

	$F_{3,1}$	F2	$F_{4,5}$
$F_{3,1}$		0	2
F2	2		0
$F_{4,5}$	1	2	

Ici on a deux chevauchement qui nous intéressent car ils ont tout deux une valeur maximale. Comme il faut en choisir qu'un, on va prendre le premier que l'on rencontre en parcourant le tableau, soit overlap($F_{3,1}$, $F_{4,5}$) = 2. Celui-ci concerne la séquence 'AG' qui est présente à la fin de la chaîne $F_{3,1}$ et au début de $F_{4,5}$

On obtient donc par concaténation $F_{3,1,4,5}$: CAATGCATTGCAGACCTGAG

Etape 4 :

	$F_{3,1,4,5}$	F2
$F_{3,1,4,5}$		2
F2	2	

Enfin pour le dernier chevauchement on procède comme précédement et on trouve overlap($F_{3,1,4,5}$,F2) = 2, celui-ci correspond à la séquence 'TG' que l'on retrouve en fin de $F_{3,1,4,5}$ et au début de F2.

On peut donc effectuer la dernière concaténation qui nous donne $F_{3,1,4,5,2}$: GCAGACCT-GAGCAATGCATTGC la chaîne finale, de longueur 22.

Exercice 2

Algorithme 1 : superChaineGlouton

Entrées : **F** = {F1,F2....Fn} un ensemble de chaines.
Sorties : **Sup** une chaîne de caratère contenant la superchaine extraite de F.
Données : **Sup** un ensemble de chaînes, **nbCommun** un entier, **X** et **Y** des chaînes de caractères.

```
début
    Sup ← F ;
    nbCommun ← 0 ;
    X,Y ← ∅ ;
    tant que Sup contient plus d'une chaîne de caractère faire
        pour tout les mots m1 de Sup faire faire
            pour tout les mots m2 de Sup tel que m2 != m1 faire
                si nbCommun == 0 alors
                    X ← m1 ;
                    Y ← m2 ;
                finSi
                temp ← overlap(m1, m2) ;
                si nbCommun < temp alors
                    nbCommun ← temp
                    X ← m1 ;
                    Y ← m2 ;
                finSi
                On supprime m1 et m1 de Sup ;
                On ajoute à Sup la concaténation des chaines (m1 - les nbCommun
                  derniers caractères ) et m2 ;
            finPour
        finPour
    finTq
fin
```

Cet algorithme à une compléxité d'environ $\mathcal{O}(n^3 * x)$ ou n est la taille de l'ensemble F et x la complexité de l'algorithme *overlap()*.

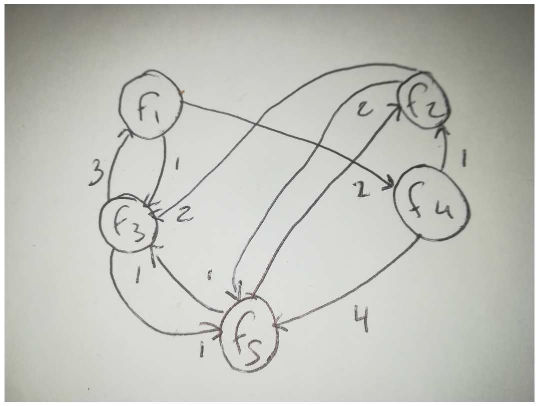

FIGURE 1 – Graphe des chevauchements

Exercice 3

Le graphe de la figure 1 ci-dessus peut être réaliser à partir de l'algorithme suivant :

Algorithme 2 : constructGraphOverlap

Entrées : F = {F1,F2....Fn} un ensemble de chaines.
Sorties : G(V,E) Le graphe des chevauchement.
Données : G(V,E) Le graphe des chevauchement.

début
  Soit **G(V,E)** un graphe avec **V** les sommets et **E** les arêtes.
  V ← **F** ;
  A ← ∅ ;
  pour *tout les mots* ***m1*** *de* ***V faire*** **faire**
    pour *tout les mots* ***m2*** *de* ***V*** *tel que* ***m2*** *!=* ***m1*** **faire**
      temp ← *overlap*($\mathrm{m}_1, \mathrm{m}_2$) ;
      si ***temp*** $>$ *0* **alors**
        A ← **A** ∪ L'arête qui part de $\mathbf{m}_1$ vers $\mathbf{m}_2$ avec un poid de **temp** ;
      finSi
    finPour
  finPour
fin

Cet algo possède une complexité d'environ $\mathcal{O}(n^2 * x)$ avec n le nombre de chaîne de F et x la complexité de l'algorithme *overlap()*.

Exercice 4

Algorithme 3 : cycleHamiltionienGlouton

Entrées : G(V,E) un graphe de chevauchement.
Sorties : H un chemin hamiltonien (Un ensemble d'arêtes).
Données : H un enemble d'arête. **O** un sommet.

début
 H ← ∅
 tant que ***E***≠ ∅ **faire**
  temp ← L'arête de **E** de poid maximal ;
  /*Si on a plusieurs arêtes de poid maximal on prend la première qui arrive*/
  O ← Le sommet à l'origine de l'arête stockée en **temp** ;
  H ← **H** ∪ **temp** ;
  E ← **E** privé des arêtes qui ont pour origine **O** ;
 finTq
 retourner H
fin

Cet algorithme a une complexité d'environ $\mathcal{O}(n * x)$ avec x la complexité de pour l'action qui permet de prendre "L'arête de **E** de poid maximal"

On remarque sur la figure 2 ci-dessous, que l'on retrouve la superchaîne de l'exercice 1 si on suit le cycle hamiltonien.

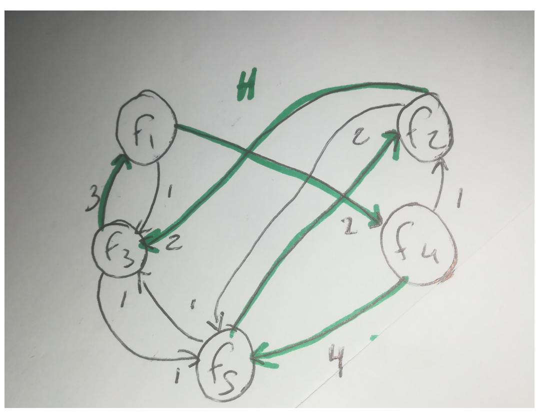

Figure 2 – H le chemin hamiltonien