Fragmentation de génomes bactériens : deux approches d'optimisation combinatoire

R. Andonov[1,2], D. Lavenier[1], N. Yanev[3] et P. Veber[1]

1. IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France
(randonov,lavenier)@irisa.fr, veberp@ifsic.univ-rennes1.fr

2. Université de Valenciennes, Le Mont Houy 59313 Valenciennes, France

3. Université de Sofia, Bulgarie, choby@math.bas.bg

Mots-clefs : optimisation combinatoire, programmation linéaire, graphes de flot

Problématique Une des possibilités pour étudier la plasticité des génomes bactériens consiste à tronçonner plusieurs souches en *fragments* de taille sensiblement identique, puis à amplifier par PCR [1] ces fragments de manière à obtenir un profil relatif à chaque souche. La dissemblance entre ces profils reflète le degré de divergence entre diverses souches d'une même bactérie. L'intérêt d'un telle méthode est qu'elle évite le séquençage systématique de toutes les souches pour mettre en évidence leurs différences.

La découpe se base sur une souche référence dont on connaît l'intégralité de la séquence, et sur laquelle on repère des points particuliers – les *amorces* – qui délimitent précisément les extrémités des fragments. Il y a deux types d'amorce : les amorces qui débutent et les amorces qui terminent un fragment. Elles sont de taille identique (autour de 25 nucléotides) et sont déterminées à partir de critères tels que leur pourcentage en nucléotides G ou C, leur propriété thermodynamique, l'absence de courtes séquences palindromiques, etc.

Une fois ces amorces repérées, se pose le problème de couvrir le génome en fragments chevauchants. Aussi, étant donné une taille minimum et maximum de fragment, ainsi qu'un intervalle minimum et maximum de chevauchement, il faut trouver une suite de fragments qui tende vers un recouvrement optimal, i.e. une suite de fragments de taille proche. L'espace des solutions dépend à la fois de la taille de la séquence génomique (quelques millions de nucléotides) et du nombre d'amorces (quelques dizaines de milliers); il est suffisamment grand pour interdire une exploration systématique de toutes les solutions et requiert des techniques d'optimisation combinatoire pour délivrer des résultats en un temps raisonnable. Nous proposons deux approches, la première fondée sur la recherche de composantes connexes dans un graphe, et la seconde, basée sur la résolution d'un modèle linéaire MIP (*Mixed Integer Problem*).

Modèle graphe de flot Formalisons cet énoncé : étant donnée une séquence nucléotidique et la position des amorces s'y trouvant, on construit l'ensemble F des *fragments admissibles*, *i.e.* des fragments f tels que :

- f commence sur une amorce de début et s'achève sur une amorce de fin.
- sa longueur $\mathcal{L}(f)$ vérifie $\underline{L} \leq \mathcal{L}(f) \leq \overline{L}$ ou $\underline{L}$ et $\overline{L}$ sont des constantes données.

On munit F d'une relation binaire "est compatible avec" : $f \prec f'$ ssi f et f' sont deux fragments successifs dont le chevauchement est compris entre les constantes $\underline{O}$ et $\overline{O}$. On définit ensuite une fonction de coût $\mathcal{C}$ sur F : $\forall f \in F \quad \mathcal{C}(f) = |\mathcal{L}(f) - L|$ où L est une donnée du problème et représente la longueur idéale d'un fragment.

1. Polymerase Chain Reaction, consulter à ce sujet l'ouvrage de D. Larzul intitulé *La PCR, un procédé de réplication in vitro*, Éts Tec & Doc

On désigne par $F_s \subset F$ (resp. $F_t \subset F$) l'ensemble des fragments qui commencent (resp. s'achèvent) "suffisamment" près du début (resp. de la fin) de la séquence.

Définition Le *graphe de recouvrement* de la séquence est le graphe orienté (S, A) où :

- l'ensemble des sommets est $S = F \cup \{s, t\}$, où s et t sont des sommets distingués, appelés *entrée* et *sortie* du graphe.
- l'ensemble d'arcs correspondant est

$$A = \{(f, f') \in F \times F \ : f \prec f'\} \cup \{(s, f) \in \{s\} \times F \ : f \in F_s\} \cup \{(f, t) \in F \times \{t\} \ : f \in F_t\}$$

Chercher une suite de fragments couvrant la séquence nucléotidique sauf éventuellement aux extrémités de celle-ci revient donc à chercher un chemin de s à t dans le graphe de recouvrement de la séquence. On procède maintenant à une extension de la fonction de coût :

$$\begin{cases} \mathcal{C}(s) = \mathcal{C}(t) = 0 \\ \forall (i,j) \in A \quad \mathcal{C}(i,j) = \max \mathcal{C}(i)\ \mathcal{C}(j) \\ \text{Pour tout chemin } r = sv_0 \dots v_n t \quad \mathcal{C}(r) = \max\limits_{i=0\dots n-1} (v_i, v_{i+1}) \end{cases}$$

Si R est l'ensemble des chemins de s à t, le problème revient à trouver $\min\limits_{r \in R} \mathcal{C}(r)$.

Un algorithme basé sur la recherche de composantes connexes dans un graphe a été proposé pour ce problème. Cet algorithme, de complexité $O(|A| \log_2 T)$, permet d'extraire toutes les solutions optimales au sens où elles ont été définies pour une coût maximal toléré T.

Un modèle linéaire (Mixed Integer Problem, MIP) Soit $G = (S, A)$ un graphe de recouvrement. A chaque arc $(i, j) \in A$ nous pouvons associer une variable booléenne x_{ij}. Le problème s'exprime ainsi comme un problème de flot dans un graphe.

Minimiser ξ selon les contraintes :

$$\sum_{j \in \Gamma^+(i)} x_{ij} - \sum_{j \in \Gamma^-(i)} x_{ji} = 0, \text{ pour } i \notin \{s, t\} \quad \textit{(Loi de Kirchhoff)}$$

$$\sum_{i \in \Gamma^+(s)} x_{si} = 1 \quad \textit{(Flot en entrée)}$$

$$\sum_{i \in \Gamma^-(t)} x_{it} = 1 \quad \textit{(Flot en sortie)}$$

$$\forall (i,j) \in A \quad \mathcal{C}(i,j) x_{ij} \leq \xi \quad \textit{(Linéarisation de la fontion objective)}$$

Conclusion Les deux approches ont été validées. L'algorithme de composantes connexes a été programmé en *Objective* CAML, tandis que le modèle MIP a été résolu en utilisant le logiciel CPLEX d'ILOG. Il n'est donc pas surprenant qu'aujourd'hui l'algorithme dédié se comporte mieux que la réalisation CPLEX. Néanmoins, l'intérêt de cette deuxième approche est plutôt méthodologique : une fois le modèle trouvé, l'effort de programmation est quasi-nul. Ce moyen est probablement moins performant en vitesse, mais permet une réutilisation simple et rapide d'un composant logiciel fiable et très puissant, utile ne serait-ce que pour valider les résultats.