Le séquençage du génome humain

Thomas Schiex

Tangente

Bibliothèque Tangente
L'aventure mathématique

Mathématiques & biologie

L'organisation du vivant

EDITIONS POLE
HS n° 42
ISSN 0987-0806

par Thomas Schiex

Le séquençage du génome humain

En séquençant le génome d'un être vivant, le biologiste espère accéder directement au « programme principal » qui s'exécute dans chaque cellule. Il s'agit d'un enjeu scientifique aux implications considérables. Pour y répondre, des outils mathématiques et informatiques sophistiqués sont nécessaires au séquençage du génome humain.

L'information génétique de tous les organismes vivants, stockée dans les chromosomes, est écrite sur de l'ADN, un polymère constitué d'une séquence de molécules élémentaires appelées nucléotides ou bases : A (pour adénine), T (pour thymine), G (pour guanine) et C (pour cytosine). La succession de ces lettres forme un grand livre dans un volume microscopique. La séquence du génome humain, finement embobinée dans vingt-trois chromosomes, représente un total de plus de trois milliards de caractères, pour une longueur totale d'environ un mètre. En termes de quantité d'information, chaque base observée, parmi les quatre possibles, apporte deux bits d'informations. Un génome humain représente donc un total d'environ 800 Mo (mégaoctets) d'information par cellule, environ 600 000 pages écrites ou un CD complet. En lisant cette séquence, le biologiste espère accéder directement au « programme principal » qui s'exécute dans chaque cellule.

Des chaînes de Markov cachées au *shotgun*

Avant de s'atteler à la lecture du génome humain, les biochimistes se sont attaqués à des génomes de virus, très compacts. Après un séquençage de l'ARN (une molécule intermédiaire) du génome d'un virus de 3 569 bases, en Belgique entre 1972 et 1976, le biochimiste britannique Frederick Sanger et son équipe séquencent l'ADN d'un autre virus en 1977. La technique employée par Fred Sanger et ses collègues, simple à utiliser, est rapidement devenue une méthode de choix. Sa principale limitation est qu'elle est incapable de lire une séquence d'ADN qui dépasse quelques centaines de bases. C'est cette technologie qui a été utilisée pour le génome humain, pris en charge dans le cadre du Human Genome Sequencing Project (HGSP) à partir de 1990. Né aux États-Unis, ce projet de treize ans a impliqué en sus le

Royaume-Uni, la France, l'Allemagne, le Japon, la Chine et l'Inde. Tout d'abord, de nombreuses informations importantes sur le génome humain ont été accumulées, en particulier une carte génétique fine, positionnant un ensemble de balises sur le génome. La construction de telles cartes s'appuie de façon importante sur des techniques probabilistes comme les chaînes de Markov cachées.

Pour faire face à la limitation sur la taille maximale des lectures possible, l'approche par *shotgun* consiste à couper un ensemble de copies d'un chromosome en fragments aléatoires dont la longueur est contrôlable et dont on va lire une extrémité (parfois les deux ; on parle alors de *mate pairs*). Cet ensemble non ordonné de lectures (ou *reads* en anglais), dont la position est inconnue *a priori*, forme un gigantesque puzzle qu'il va falloir *assembler* (voir sur la figure ci-contre) en comparant les différentes lectures entre elles. Pour l'ensemble du génome humain dans sa version de 2003, on estime que chaque base a été lue environ dix fois. Ce sont des dizaines de millions de lectures qu'il faut comparer entre elles et assembler pour résoudre ce puzzle ! En dehors d'un simple problème de taille, cet assemblage est rendu difficile par le fait que le séquençage est un processus imparfait, qui commet des erreurs de lecture.

Lors de la comparaison de lectures, pouvant contenir des erreurs, on doit pouvoir tolérer une correspondance imparfaite entre les deux séquences comparées. Ce problème de comparaison de séquences, appelé *alignement*, a été résolu par Needleman et Wunsch en 1972. Aligner deux séquences consiste à trouver un ensemble de modifications élémentaires (remplacement d'un caractère par un autre, effacement ou insertion d'un caractère dans une séquence) le plus simple possible permettant de rendre les deux séquences identiques. Chaque type d'opération reçoit un coût associé, et on cherche alors un ensemble de modifications dont la somme des coûts est minimum.

```
TAGTCGAGGCTTTAGATCCGATGAGGCTTTAGAGACAG
 AGTCGAG CTTTAGA  CGATGAG CTTTAGA
  GTCGAGG TTTAGATC  ATGAGGC    GAGACAG
     GAGGCTC   ATCCGAT AGGCTTT GAGACAG
 AGTCGAG    TAGATCC ATGAGGC  TAGAGAA
TAGTCGA  CTTTAGA CCGATGA   TTAGAGA
    CGAGGCT  AGATCC  TGAGGCT  AGAGACA
TAGTCGA GCTTTAG TCCGATG  GCTCTAG
   TCGACGC     GATCCGA GAGGCTT AGAGACA
TAGTCGA    TTAGATC GATGAGG TTTAGAG
  GTCGAGG TCAGAT   ATGAGGC  TAGAGAC
     AGGCTT   ATCCGAT AGGCTTT GAGACAG
 AGTCGAG   TTAGATT  ATGAGGC   AGAGACA
       GGCTTTA  TCCGATG    TTTAGAG
    CGAGGCT TAGATCC  TGAGGCT   GAGACAG
 AGTCGAG  TTTAGATC  ATGAGGC TTAGAGA
     GAGGCTT  GATCCGA GAGGCTT  GAGACAG
```

Une séquence et un ensemble de lectures aléatoires (les erreurs sont indiquées en rouge).

Le nombre d'alignements possibles entre deux séquences de tailles n et m est énorme. Il y a en effet

$$\frac{(m+n-k)!}{k!(m-k)!(n-k)!}$$

alignements contenant k correspondances parfaites. (Il s'agit du nombre de façons de partitionner un ensemble de : $k + (n-k) + (m-k) = m+n-k$ éléments en k correspondances, $m-k$ effacements sur une séquence et $n-k$ effacements sur l'autre. Au total, on a donc :

$$\sum_{k=1}^{\min(m;n)} \frac{(m+n-k)!}{k!(m-k)!(n-k)!}$$

alignements possibles. Pour deux séquences de dix caractères, cela représente 8 097 453 alignements. Pour des séquences plus longues, le chiffre devient rapidement astronomique et le temps nécessaire pour explorer tous les alignements possibles entre deux séquences devient très vite rédhibitoire (dépassant rapidement le milliard d'années, même sur une machine moderne). La solution à ce problème d'optimisation combinatoire est offerte par la

```
C T A G G T A C
|   .   |   |   .   |
C T - G G T T C
1 2 3 4 5 6 7 8
```

Les deux séquences peuvent être alignées au prix d'un effacement–insertion (position 3), d'une substitution (position 7), et de six correspondances parfaites (ou *match*), représentées par les barres verticales).

méthode de programmation dynamique, dont le domaine d'application dépasse ce problème d'alignement (voir *Mathématiques discrètes et Combinatoire*, Bibliothèque Tangente 39, 2010).

Le problème de recherche d'un alignement optimal peut se réduire à la recherche d'un plus court chemin dans un graphe bien construit. En positionnant une séquence à l'horizontale et une à la verticale, on place un sommet entre chaque intersection de deux caractères. Le graphe est construit afin que chaque chemin dans ce graphe représente un alignement. Si un début d'alignement a déjà été construit depuis le début des séquences jusqu'aux positions i et j, on peut étendre cet alignement :

1• par deux caractères mis en correspondance. S'ils sont identiques, on paye un coût faible (par exemple 0), sinon un coût plus fort (par exemple 4). On suit une arête diagonale pour se retrouver en $(i+1, j+1)$;

2• par effacement d'un caractère sur la séquence horizontale : on paye un coût intermédiaire (disons 3) et on suit une arête horizontale pour se retrouver en $(i, j+1)$;

3• par effacement d'un caractère sur la séquence verticale : on paye le même coût intermédiaire et on suit une arête verticale pour se retrouver en $(i+1, j)$.

Un tel graphe d'alignement est illustré dans la figure ci-contre (en haut). Le meilleur alignement correspond alors à un plus court chemin allant du coin supérieur gauche (en bleu) au coin inférieur droit (en vert).

Pour calculer le coût d'un chemin optimal et identifier ce chemin, on associe à chaque position de la grille (i, j) le coût $C(i, j)$ du meilleur alignement allant du sommet bleu au sommet (i, j). En utilisant le principe de la programmation dynamique, on va utiliser la solution de problèmes plus simples, et déjà résolus. Initialement, seul $C(0, 0)$ est connu et égal à 0. Il suffit alors

La programmation dynamique

Introduite dans les années 1940 par Richard Bellman, la programmation dynamique permet de résoudre des problèmes d'optimisation pouvant se décomposer en problèmes plus simples, de façon récursive. La recherche d'un plus court chemin entre deux sommets a et b dans un graphe en est un exemple : pour tout sommet intermédiaire c sur un plus court chemin de a à b, le chemin de c à b est forcément un plus court chemin qui ne passe pas par a, sinon cela signifierait qu'il existe un raccourci que le chemin le plus court pourrait emprunter, et il ne serait pas optimal. On pourra considérer par exemple pour c l'ensemble des voisins de a et se ramener ainsi à un ensemble de problèmes plus simples qui deviendront *in fine* élémentaires (plus court chemin entre deux sommets voisins). En mémorisant le coût associé à chacun de ces sous-problèmes, on évite les calculs redondants et on obtient un algorithme efficace.

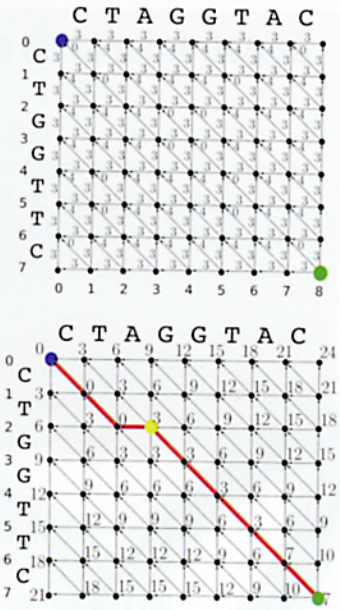
d'organiser ses calculs (ligne par ligne, de gauche à droite par exemple) pour remplir la matrice de programmation dynamique. Le sommet (0, 1) ne peut être atteint que depuis le sommet (0, 0), la longueur du chemin associé est de $C(0,0)+3=3$. De façon générale, le coût d'un plus court chemin atteignant (i,j) dépend des coûts des sommets $(i-1,\ j-1)$, $(i,\ j-1)$ et $(i-1,\ j)$ qui permettent de l'atteindre respectivement par une correspondance (parfaite ou non) ou par un effacement sur l'une ou l'autre des deux séquences. Il satisfait donc la relation de récurrence suivante :

$$C(i,j)=\min\begin{cases} C(i-1,j-1)+0 & \text{(deux caractères identiques),}\\ C(i-1,j-1)+4 & \text{(deux caractères différents),}\\ C(i,j-1)+3 & \text{(effacement horizontal),}\\ C(i-1,j)+3 & \text{(effacement vertical).}\end{cases}$$

Ainsi, le sommet jaune (2, 3) peut être atteint par une correspondance imparfaite de coût 4 ($C(1,2)+4=7$), ou par un effacement sur l'une ou l'autre séquence ($C(2,2)+3=3$ et $C(1,3)+3=9$). On peut donc atteindre (2, 3) par un chemin de longueur 3 seulement, arrivant depuis (2, 2). Ce que l'on mémorise pour poursuivre le calcul jusqu'à (7, 8). On obtient alors le coût, 7, du plus court chemin. Pour retrouver ce chemin, il suffit de suivre à l'envers l'origine du coût de chaque sommet, jusqu'à (0, 0). Ce calcul est réalisé ainsi en un temps proportionnel au produit des longueurs des séquences, bien loin du nombre de chemins–alignements possibles !

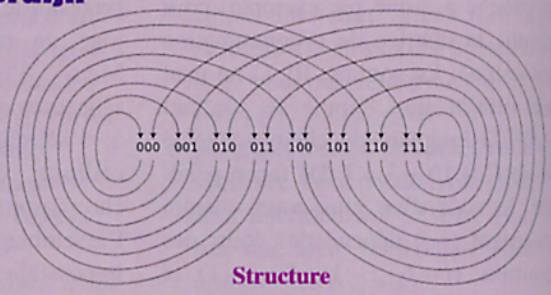
Assemblage

En comparant les lectures entre elles par alignement, il devient possible de savoir si deux pièces du puzzle peuvent s'assembler (plus ou moins bien) et cela fournit une information essentielle pour pouvoir reconstruire la séquence du génome. La modélisation probabiliste des erreurs y joue un rôle important. Mais l'existence de séquences qui apparaissent à l'identique en deux endroits distincts d'un chromosome peut rendre l'assemblage du puzzle impossible : plusieurs pièces issues d'endroits différents s'assemblent au même endroit. Pour résoudre ce problème, on construit des collections de fragments d'ADN de taille contrôlée dont on séquence les deux extrémités. Si l'un de ces fragments enjambe la séquence répétée (ses extrémités sont situées de part et d'autre de cette séquence), il permet de poursuivre la construction de la séquence en s'affranchissant de la répétition. De fait, l'ensemble des ambiguïtés qui doivent être résolues durant l'assemblage rend ce processus très complexe. Pour en limiter la difficulté, il est possible de fractionner l'ADN d'un chromosome en un ensemble de segments assez longs (de 100 à 150 000 bases) qui se chevauchent et recouvrent l'ensemble du chromosome (on parle de *chemin couvrant minimum*). Pour pouvoir être conservé et dupliqué, chacun de ces fragments est inséré dans un chromosome de bactérie et appelé pour cette raison BAC (*bacterial artificial chromosome*). Cette approche a été utilisée par le consortium public HGSP en charge du séquençage du génome humain, permettant de répartir le travail entre les différents pays impliqués. En parallèle, un projet concurrent, mené par l'entreprise privée Celera, a utilisé une approche sans BAC (*shotgun* pur), mais il est difficile de savoir si elle aurait abouti sans disposer



Des structures du graphe de Bruijn

Le graphe de Bruijn, défini par l'ensemble des k-mères possibles, peut se représenter sur un plan de façon à ressembler à des objets mathématiques bien connus, comme l'attracteur de Lorenz.



Structure du graphe de Bruijn issue d'un ensemble de lectures.

Des 3-mères définis sur un alphabet binaire.

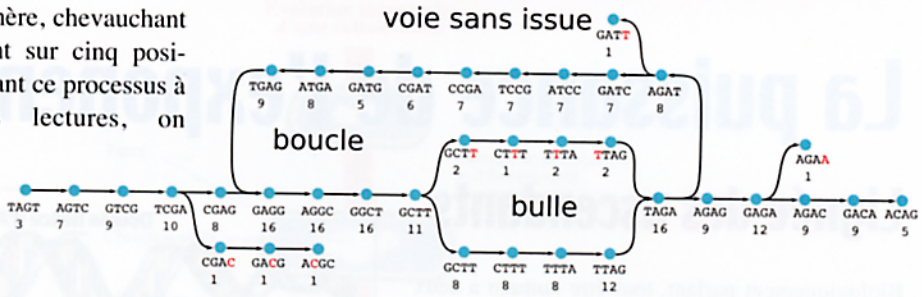
des données publiques du HGSP. En 2003, date à laquelle la séquence du génome humain a été officiellement livrée par le HGSP, la séquence couvrait 95 % du génome avec un taux d'erreur estimé à 0,01 % (une erreur attendue toutes les 10 000 bases). L'essentiel des 5 % manquants est formé d'hétérochromatine, une partie du génome condensée, souvent très répétitive, peu accessible et considérée comme peu active. Il reste en sus de ces régions de nombreux « trous » de courte taille (*gaps*) liés à des difficultés de séquençage locales. Le coût du séquençage d'une base au cours du projet a continuellement baissé, depuis les 0,25 $ prévus initialement pour aboutir à un coût moyen de 0,09 $ par base. Ce n'était que le début d'un mouvement de grande ampleur.

Depuis les années 2000, les technologies de séquençage ont énormément évolué. Les quelques trente milliards de bases séquencées par le HGSP en treize ans sont à comparer aux 600 milliards de bases séquencées en une étape (*run*) par un séquenceur récent. Cette capacité accrue a un prix : les lectures réalisées sont sensiblement plus courtes (typiquement cent bases pour la plus récente et répandue des technologies). Les problèmes d'assemblage soulevés sont donc d'une nouvelle nature. Depuis plusieurs années, une nouvelle famille d'algorithmes d'assemblage a ainsi vu le jour. Elle permet d'éviter les comparaisons deux à deux de chaque lecture, qui deviennent insurmontables lorsque l'on manipule $n = 3 \times 10^9$ lectures, nécessitant n^2 comparaisons. Ces algorithmes s'appuient sur les chevauchements entre k-mères (mots de longueurs k) apparaissant dans les lectures, représentés dans un graphe spécifique, appelé *graphe de Bruijn* (du nom de Nicolaas Govert de Bruijn, mathématicien néerlandais né en 1918). Un arc relie un k-mère à un autre si ces deux k-mères se chevauchent parfaitement sur $k - 1$ positions :

```
TGCGATGAGT
TGCGAT→GCGATG→CGATGA→GATGAG→ATGAGT
```

La séquence en haut de cette figure se représente via le graphe de Bruijn situé en dessous. Chaque sommet corres-

pond ici à un 6-mère, chevauchant le 6-mère suivant sur cinq positions. En appliquant ce processus à l'ensemble des lectures, on construit un graphe d'une très grande taille. Sa représentation en mémoire peut nécessiter des machines inhabituelles (équipées de 1 To, soit 1 024 Go, de mémoire vive). L'intérêt essentiel de cette approche est que le temps de construction du graphe croît linéairement avec le nombre de lectures. On peut en effet transformer un *k*-mère en un nombre entier en associant un numéro aux quatre bases possibles (A = 0, T = 1, G = 2, C = 3 par exemple). Le *k*-mère TGCGAT vaut ainsi 123 201 en base 4, soit 1 761 en base 10, ce qui peut permettre de le retrouver (s'il avait déjà été rencontré) ou de l'insérer dans le graphe sinon. La table qui fait la correspondance entre le code d'un k-mère et sa position est appelée *table de hachage*. Ce calcul est rapide (en temps constant pour *k* fixé). Chaque sommet de ce graphe est pondéré par le nombre de fois où il apparaît dans les lectures. Un chemin dans ce graphe correspond alors à un assemblage possible. Dans les premiers outils d'assemblage, c'est un chemin eulérien – passant une fois par chaque sommet – qui était recherché. Mais cette approche n'a pas rencontré beaucoup de succès en pratique. Les erreurs de séquençage et les répétitions rendent cette recherche de chemin difficile. Une erreur en extrémité de lecture fait ainsi apparaître une voie sans issue. Située à l'intérieur d'une lecture, elle crée une bulle, alors qu'une répétition créera une boucle. L'assemblage consiste alors à faire des choix en se



débarrassant de ces structures parasites. Les poids associés aux sommets permettent d'identifier de façon heuristique les régions répétées. Cette information et la distance entre *mate pairs* permet de traiter les problèmes des boucles générées par les répétitions. Le séquençage et l'assemblage de génomes microbiens est maintenant possible en moins d'une journée sur une machine moderne. Le traitement direct de génomes de grande taille reste encore difficile. Plus ardu encore est le décodage de cette séquence qui permettrait une fine compréhension du fonctionnement cellulaire. La séquence d'ADN ne raconte pas tout car bien d'autres mécanismes sont en œuvre dans la cellule.

T.S.

RÉFÉRENCES

- **Sense from sequence reads: methods for alignment and assembly.** Paul Flicek et Erwan Birney, *Nature Methods*, 2009.
- **A general method applicable to the search for similarities in the amino acid sequence of two proteins.** Saul Needleman et Christian Wunsch, *Journal of Molecular Biology*, 1970.

Tangente

Mathématiques & biologie

L'organisation du vivant

- Les animaux mathématiciens
- Dynamique des populations
- Biostatistiques et génétique
- Le cerveau

Les mathématiques : outil fondamental des sciences du vivant ? C'est ce que suggèrent de nombreuses constatations : la présence dans la nature de structures mathématiques complexes (spirales, fractales, symétrie dans la nature...), les capacités cognitives des chimpanzés, les alvéoles des abeilles ou la géométrie complexe des toiles d'araignées...

Des modèles théoriques, validés par l'expérience, permettent d'appréhender de nombreux phénomènes biologiques complexes, tels les équilibres entre proies et prédateurs ou l'étude du comportement collectif des oiseaux et bancs de poissons.

Enfin, le séquençage des génomes et le fonctionnement du cerveau semblent eux aussi suggérer que les mathématiques, loin d'être le propre de l'homme, seraient en fait le propre du vivant.

9 782848 841199

Prix : 19,80 €

EDITIONS POLE