Analyse statistique de génomes

LI323 - Statistique et informatique

Oct. 2014

Résumé

L'objectif de ce projet est de mettre au point des stratégies d'annotation automatique de génomes à l'aide de méthodes statistiques. Un génome est composé de différents éléments qui sont "lus" par la cellule pour lui permettre de réagir à différentes conditions. Nous verrons comment nous pouvons par l'analyse statistique de la séquence génomique (un texte) localiser ou détecter ces éléments.

Rendu du projet : Le projet peut être fait en binôme. Vous devrez envoyer par mail à `hugues.richard@upmc.fr` :

— un compte rendu (word, open office, pdf) de quelques pages (max. 5) sur vos réponses au questions pour le projet.
— le code du projet dûment commenté (en java).

1 Remarques préliminaires

1.1 Rappels de biologie

Les objets que nous étudions sont des génomes d'organismes vivant. En première approximation, un génome peut être vu comme une chaîne de caractère écrite dans un alphabet à 4 lettres (`A`, `C`, `G` ou `T`). Depuis le début des années 90, il est devenu de plus en plus simple de **séquencer** un génome (le séquençage d'un génome humain coûte à l'heure actuelle un peu moins de 5000$). Cependant on ne peut pas comprendre, simplement à partir de la séquence génomique, comment cette information est utilisée par la cellule (un peu comme avoir a disposition un manuel d'instructions écrit dans une langue inconnue). On doit donc **décrypter** le code qui permettrait de comprendre le rôle des différents éléments qui pourraient exister le long de ce génome [1].

Comme première approximation, on peut distinguer deux types d'éléments d'intérêt le long des génomes :

— Les **gènes**, des séquences qui sont **transcrites** en ARN messager. L'ARNm est ensuite **traduit** en protéine, qui sont les molécules qui font la grande partie de l'activité moléculaire. La traduction d'ARNm en protéine est faite à l'aide du code génétique qui à chaque groupe de 3 lettres (appelé codon) fait correspondre un acide aminé (alphabet à 20 lettres). La protéine commence dans quasiment tous les cas par un codon `ATG`. D'autres codons ne codent pas pour des acides aminés mais signifient la fin de la chaîne de protéines, il sont appelés codons stop [2]. Dans le cas des organismes procaryotes, les gènes sont définis le long d'une séquence contiguë. La longueur d'un gène peut aller de 50 à quelques milliers de nucléotides (*e.g.* de lettres).
— Les **séquences promoteurs**, il s'agit de séquences généralement situées avant les gènes, et qui servent d'interrupteurs pour décider si on transcrit le gène. L'activation est faite en pratique par une protéine (un facteur de transcription) qui se fixe à l'ADN sur un segment relativement court. Ces facteurs de transcription reconnaissent des mots de 6 à 10 lettres bien particuliers.

Dans la suite nous ne considérerons que des exemples sur des organismes procaryotes (des bactéries sans noyaux), donc les génomes sont plus compactes et plus simples pour une première analyse. Un procaryote possède en général un seul chromosome circulaire.

1.2 Données

Les génomes sont décrits comme des fichiers texte simple au format FASTA, par exemple :

1. La séquence génomique peut être lue dans les deux directions (de la gauche vers la droite ou de la droite vers la gauche). Pour la suite de ce projet, nous nous concentrerons uniquement sur la direction de la gauche vers la droite
2. les codons `TAA`, `TAG` ou `TGA`

```
>chromosome 1
ATCATGTCTAGCTGAGTTTGNNNGACGGTCGGCCTTTGACAAGACAGGTGTAGCCATCTTAATGCAATGT
CTTGCAAGACTGTGTGCGCTGAGTTCGAGACAATCACCGGGAGACGAGACTATCCAATTGCGCCAATTGC
TGGCGACAGCGCTTGCCCAGGCCTTTATATCGGAAAACGCGATGCTGAAAGAAAGCGGAGGAGTTCCAAT
...
```

Chaque séquence est décrite par un en-tête (commençant par le caractère `>` et suivi d'une description) et la séquence est écrite ensuite sur plusieurs lignes (en général on écrit des lignes de 60 caractères). Notez qu'un fichier FASTA peut contenir plusieurs séquences (par exemple des chromosomes, des gènes...). La séquence est écrite dans l'alphabet à quatre lettres `A`, `C`, `G` ou `T` auquel on ajoute le caractère `N` pour les bases indéterminées.

Nous utiliserons aussi un format pour décrire les modèles d'apparitions de lettres ou groupes de lettres dans le génome, en tabulant en face de chaque mot, la probabilité de ce mot d'apparaître à un endroit dans la séquence.

Vous aurez à disposition à l'adresse suivante www.lcqb.upmc.fr/hrichard/LI323/ plusieurs informations :

— des fichiers fasta de différents génomes.
— Un fichier avec les gènes du même génome (partie 2)
— des fichiers de séquences régulatrices (partie 3)

2 Détection de gènes

2.1 Un modèle simple de génomes

Pour caractériser par un modèle probabiliste simple un génome, nous allons utiliser dans un premier temps le modèle de langue utilisé dans le premier projet. Nous supposons que la fréquence d'apparition des 4 lettres `A`, `C`, `G` et `T` dépend du génome, et que la probabilité pour une séquence $\mathbf{s}$ connaissant le modèle de génome g s'écrit :

$$p(\mathbf{s} \mid g) = \prod_{i=1}^{|\mathbf{s}|} p(s_i \mid g)$$

1. Appliquer le modèle de langue vu pour le projet 1 pour estimer un modèle à partir d'un génome donné. Ici, chaque organisme a en quelque sorte son génome écrit dans une "langue" spécifique. Ecrire la fonction correspondante d'estimation à partir d'un fichier FASTA
2. Adapter la méthode du projet 1 qui permette de prédire que quelle espèce un segment d'ADN donné vient à partir de différent modèles de génomes.
3. Le modèle utilisé ne se base que sur les fréquences de 4 lettres. Toujours en vous inspirant du projet 1, proposez un modèle de génome qui pourrait être plus spécifique. Quelles pourraient être les limitations ?

2.2 Prédiction de gènes

Pour maintenant détecter dans un génome les régions qui correspondent à des gènes, nous allons prendre en compte le **code génétique**, *i.e.* le fait que ces séquences sont écrites dans un alphabet à codons, où les mots sont écrits par blocs de trois. Le modèle probabiliste correspondant calcule donc pour une séquence $\mathbf{s}$ de longueur ℓ (ℓ doit être divisible par trois)

$$p(\mathbf{s} \mid gene) = \prod_{i=0}^{\ell/3-1} P(s_{3i+1}s_{3i+2}s_{3i+3} \mid gene)$$

où $s_{3i+1}s_{3i+2}s_{3i+3}$ est le i$^{\text{ème}}$ codon du gène. Notez que dans ce modèle les codons stop auront une probabilité nulle.

1. Ecrire une fonction qui, à partir de plusieurs séquences de gènes, estime le modèle d'apparition de codons.
2. Ecrire une fonction qui calcule la probabilité d'une séquence d'être un généré par le modèle de gène.
3. Ecrire une fonction qui décide si un fragment de séquence est ou non un gène. Attention, comme les codons sont composés de trois lettres, vous devrez à chaque fois tester trois phases de lecture différentes et négliger les séquences qui pourraient apparaitre après un codon stop.
4. On veut maintenant annoter les gènes le long du génome. Pour faire cela dans une première phase, on extrait va utiliser une fenêtre d'une certaine longueur ($d = 100$ nt par exemple), et la classer comme gène ou non gène. Ecrire la fonction correspondante.

5. Comment se comporte la qualité des résultats en fonction de la longueur de la fenêtre ? Quelle limitation voyez vous à cette méthode ? Comment pourrait-on imaginer une méthode plus performante ?

3 Annotation de régions promoteurs

Dans la seconde étape, on veut pouvoir détecter les mots qui correspondent à des régions de régulation. On part du principe que si un mot est "important" pour la cellule, il apparaitra plus souvent que les autres mots de la même longueur.

Par contre, on veut se prémunir contre les effets induits par la composition du génome. Pour l'analyse on va avoir besoin de plusieurs ingrédients :

— compter tous les mots d'une certaine taille dans un génome
— Si on observe un mot $\mathbf{w}$, $n_{\mathbf{w}}$ fois dans une séquence de longueur ℓ, quelle est la probabilité d'observer au moins ce nombre d'occurrences, ie $p(N_{\mathbf{w}} \geq n_{\mathbf{w}})$ ($N_{\mathbf{w}}$ est la variable aléatoire du comptage de $\mathbf{w}$ dans une séquence de longueur ℓ) ?
— On veut évaluer la validité de nos calcul de probabilités empiriques à l'aide de simulations.

3.1 Simulation de séquences aléatoires

1. Ecrire une fonction pour simuler une séquence de longueur donnée x fois.
   (a) en utilisant un modèle où toutes les lettres ont la même probabilité (équiprobable)
   (b) avec un modèle où $p_A = p_T = 0.4$ et $p_G = p_C = 0.1$
2. Faire une étude par simulation de la probabilité empirique pour un comptage de mot donné n pour les mots `ATCTG`, `ATATAT` ou `AAAAA` pour les deux modèles donnés précédemment. Rappel : la probabilité empirique $p_{\text{emp}}(N \geq n)$ se calcule simplement comme :

$$p_{\text{emp}}(N \geq n) = \sum_{i=1}^{nsim} \mathbf{1}_{\{n^{(i)} \geq n\}}$$

   où $n^{(i)}$ est le comptage de la i$^{\text{ème}}$ simulation.
3. Comment peut-on calculer un intervalle de confiance pour cette probabilité empirique ?

3.2 Probabilités de mots

1. Calculer la probabilité d'avoir un mot w qui apparait dans une séquence aléatoire. On fera l'hypothèse que le mot w ne peut pas se recouvrir lui même, et donc que différentes occurrences du mot sont indépendantes et qu'on peut utiliser une loi binomiale.
2. comparer la probabilité d'occurrence calculée pour les mots `ATCTG` `ATATAT` ou `AAAAA` avec la probabilité empirique. Que remarquez vous ?
3. Ecrire une fonction pour compter tous les mots d'une longueur donnée dans un texte. Attention ! Ces mots peuvent se recouvrir entre eux.
4. Appliquez ce modèle pour analyser le groupe de séquences promoteurs données.