Projet 3 : Analyse de séquences génomiques

Résumé

L'objectif de ce projet est d'utiliser des méthodes d'analyse statistiques pour extraire de l'information des données génétiques. Un génome peut être vu en première approximation comme un suite de lettre (une sorte de livre), composé de différents éléments qui sont "lus" par la cellule pour lui permettre d'accomplir ses tâches. Nous verrons comment nous pouvons par l'analyse statistique de la séquence génomique (un texte) localiser ou détecter ces éléments.

Rendu du projet :

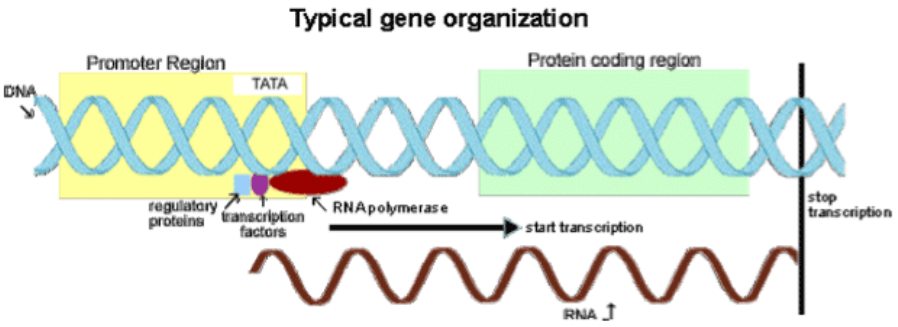
- un compte rendu (format pdf) de quelques pages (max. 5) sur vos réponses au questions pour le projet. Les questions qui attendent une réponse sont marquées avec des étoiles suivant la longueur des réponses attendues (* : quelques lignes, ** : les réponses demandent plus de développement).
- le code du projet dûment commenté.

Note : Ce projet aborde un certain nombre de notions de biologies qui sont vues plus en détails dans le module 3I019 (Introduction à la bioinformatique). Pour faciliter notre évaluation, veuillez indiquer sur votre rendu si vous êtes inscrit à ce module.

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 1000$). Cependant on ne peut pas comprendre, simplement à partir de la séquence génomique, comment cette information est utilisée par la cellule. C'est un peu comme avoir a disposition un manuel d'instructions écrit dans une langue inconnue. On doit donc **décrypter** un code pour comprendre le rôle des différents éléments qui sont présents le long de ce génome [1].

Dans la suite nous ne considérerons que des exemples sur des organismes unicellulaire, dont les génomes sont plus compactes et dont l'analyse statistique est plus simple. Comme première approximation, on peut distinguer deux types d'éléments d'intérêt le long des génomes, les **gènes** et les **séquences promoteurs**. L'organisation de ces éléments le long de la séquence est rappelé ci dessous (pour plus d'information vous pouvez aller sur wikipedia et lire quelques pages de biologie moléculaire).



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 Préliminaires : données et lecture des fichiers

Un génome est stocké dans un fichier texte simple avec le format qu'on appelle FASTA. Un fichier typique commence par exemple ainsi :

```
> S. cerevisae
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 les séquences des gènes d'un génome, plusieurs chromosomes, ...). 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[2]. On ne tiendra pas compte des bases avec le caractère `N` dans la suite. Afin de manipuler les séquences vous avez à disposition deux fonctions qui permettent de recoder la séquence (`read_fasta(fichier)`) et de compter les occurrences de lettres (`nucleotide_count(liste_entiers)`)

- La fonction `read_fasta(fichier)` qui lit un fichier fasta et renvoie une liste de liste d'entier avec le code suivant `A` :0,..., `T` :3, et la valeur -1 pour toute autre lettre. Par exemple si le fichier "`seqs.fasta`" contient les séquences "`GATTACAACT`" et "`TANTNATA`" cela renverra la liste `[[2,0,3,3,0,1,0,0,1,3], [3,0,-1,3,-1,0,3,0]]`
- La fonction `nucleotide_count(liste_entiers)` compte les occurrences des quatre lettres dans une liste d'entiers. La fonction renvoie un tuple de taille quatre avec le nombre de `A`, `C`, `G`, et `T`, respectivement (on ignore les autres lettres). Par exemple, elle renverra le tuple `(4, 2, 1, 3)` pour la séquence "`GATTACAACT`" précédente.

Le but de cette partie préliminaire est de faire une première analyse des séquences avec un modèle de génération simple qui prend en compte les fréquences de lettres (et donc suppose que les lettres apparaissent successivement et de manière indépendante). Il s'agit donc de calculer la probabilité d'une séquence avec ce modèle.

1. Les données de séquences sont disponible sur la page du projet. Télécharger l'archive avec l'ensemble des fichiers et le décompresser dans votre répertoire personnel. Regardez la taille des fichier, le nombre de séquences dans chaque fichier, et le nombre de nucléotides pour vous faire une première impression. Combien de nucléotides compte les chromosomes de *S. cerevisae* qui sont donnés ?
2. La fonction `nucleotide_frequency(liste_entiers)` renvoie les fréquences d'apparition de chaque lettre dans un tuple. Ceci correspond à un modèle aléatoire de génomes où les occurrences successives des lettres sont indépendantes et identiquement distribuées (comme les tirages d'un dé à quatre face). Appliquer cette fonction pour estimer les fréquences des lettres sur le génome de *S. cerevisae*.
3. Ecrire une fonction `logproba(liste_entiers, m)` qui calcule la log-probabilité d'une séquence étant donné les fréquences des lettres `m`. Par exemple avec un modèle `m=(0.2,0.3,0.1,0.4)` la fonction appliquée à la séquence "`CAT`" renvoie $\log(0.3 * 0.2 * 0.4) = \log(0.3) + \log(0.2) + \log(0.4) \approx -3.7297$
   (Note : On demande de calculer des log-probabilités car les probabilités décroissent exponentiellement vite avec la longueur de la séquence, ce qui amène rapidement à des erreurs d'arrondi). Pour utiliser la fonction log, vous devez importer le module math.
4. faire une version optimisée de `logprobafast` qui prend en premier paramètre le résultat de `compte_lettres`.

2. il peut y avoir d'autres bases suivant le niveau d'indétermination : R= A ou G, Y = C ou T, etc.

3 Annotation des régions promoteurs

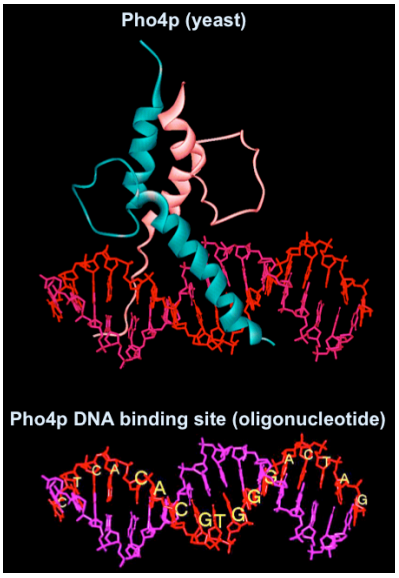


FIGURE 1: Structure protéique tridimensionnelle du facteur de transcription Pho4p attaché à la protéine (haut) et séquence du site de fixation représenté sur la molécule d'ADN.

Gene	Site Name	Sequence	Affinity
PHO5	UASp2	---aCtCaCA**CACGTGGG**ACTAGC-	high
PHO84	Site D	---TTTCCA**GCACGTGGG**GCGGA--	high
PHO81	UAS	----TTATG**GCACGTGC**GAATAA--	high
PHO8	Proximal	GTGATCGCT**GCACGTGGC**CCGA---	high
group 1	consensus	---------**gCACGTGgg**-------	high
PHO5	UASp1	--TAAATTA**GCACGTTTT**CGC----	medium
PHO84	Site E	----AATA**CGCACGTTTT**TAATCTA	medium
group 2	consensus	--------**cgCACGTTTt**-------	medium
Degenerate consensus		---------**GCACGTKKk**-------	high-med

FIGURE 2: Exemple de séquences de fixation du facteur de transcription PHO sur deux groupes de gènes. Ces séquences sont situées en amont de la séquence du gène, en général entre 200 et 500 nucléotides avant le début de la transcription.

Les séquences promoteurs sont des séquences généralement situées avant les gènes. La fixation d'une protéine (appelée facteur de transcription) sur une séquence d'ADN particulière en amont du gène sera responsable de sa transcription. On peut donc voir ces séquences de fixation comme des "interrupteurs" qui, quand il sont activés, lancent la production d'un gène et de la protéine correspondante suivant l'état de la cellule. En terme de séquence, ces facteurs de transcription se fixent donc sur des mots de 6 à 10 lettres en général. Par exemple, pour le facteur de transcription PHO (figure 1), le motif pour certain des gènes qu'il régule a été déterminé à l'aide d'une série d'expériences biologiques spécifiques. (figure 2).

3.1 Description Empirique, préliminaires

Détecter les sites de fixations avec des expériences biologiques prend beaucoup de temps et est coûteux, on veut donc pouvoir avoir des méthodes qui permettent de les connaître directement avec l'analyse de la séquence. Par contre on peut facilement détecter un ensemble de gènes qui sont activés au même moment. La question ensuite est de trouver, en amont du gène, les motifs commun qui seront responsable de l'activation. Pour pouvoir prédire automatiquement les mots qui pourraient correspondre à des sites de fixation des facteurs de transcription, nous allons utiliser le fait que ceux ci devraient être "inattendus" eu égard à la composition en nucléotide. On part du principe que si un mot est "important" pour la cellule, il apparaitra relativement plus souvent que les autres mots de la même longueur et plus que ce qu'on attendrait au hasard. On va analyser les séquences dans les fichiers `regulatory_seq_PHO.fasta`, `regulatory_seqs_GAL.fasta` et `regulatory_seqs_MET.fasta`.
Pour simplifier l'analyse, on considère qu'on **concatène** l'ensemble des séquences de chaque fichier. On travaille donc uniquement avec une liste d'entier à chaque fois.

1. Compter tous les mots d'une taille k dans un génome. On va ranger tous ces mots dans un tableau de taille 4^k, rangés en ordre lexicographique : `AA..A`= 0, ..., `TT..T`= $4^k - 1$.
   (a) Ecrire une fonction `code(m, k)` qui renvoie pour un mot `m` de taille k son indice dans le tableau ordonné lexicographiquement. Aide : l'indice correspond à l'écriture du mot en base 4 :
   `TAC`$= 3 \times 4^2 + 0 \times 4^1 + 1 \times 4^0 = 49$ (on a codé `T`=3, `A`=0 et `C` = 1)
   (b) Ecrire la fonction inverse qui, connaissant un indice i et la longueur du mot k renvoie la séquence de longueur k correspondante. Aide : vous pouvez utiliser les fonctions de la division euclidienne "`//`" et "`%`" (`10//3 = 3 et 10%3 = 1`).
   (c) Ecrire la fonction qui compte le nombre d'occurrences pour tous les mots de taille k dans une séquence d'ADN. On comptera les occurences chevauchantes, par exemple pour $k = 2$ la séquence `ATCAT` a comme comptages `AT`=2, `CA`=1, `TC`=1.
2. *Si on connait les fréquences des lettres dans le génome, quel est le nombre attendu d'occurrences pour un mot **w** dans une séquence de longueur ℓ ? Ecrire une fonction `comptage_attendu` qui prend en paramètres les fréquences des nucléotides, k, et la longueur du génome ℓ et renvoie les comptages attendus pour tous les mots de longueur k.
3. **Afficher avec un graphique 2D le nombre attendu d'occurrences sur l'axe des abcisses et le nombre observé sur l'axe des ordonnées pour tout les mots de longueur k pour les séquences PHO, GAL et MET. On testera avec $k = 2, 4, 6, 8$.
   Pour évaluer si un mot a un comptage attendu différent de son comptage observé vous pouvez les triez par rapport à l'enrichissement relatif (i.e. la droite $y = x$ sur le graphique 2D).

3.2 Simulation de séquences aléatoires

On veut évaluer la validité de nos calcul de probabilités empiriques à l'aide de simulations. On va simuler un grand nombre de séquences aléatoire et comparer la probabilité obtenue avec la valeur observée.

1. Ecrire une fonction `simule_sequence(lg, m)` qui génère une séquence aléatoire de longueur `lg` d'une composition donnée `m` (proportion de A, C, G et T). Vous pouvez repartir du code utilisé pendant la première séance de TME.
2. Avec la fonction de simulation de séquences, simuler un plusieurs séquences (par exemple 1000) pour comparer le comptage attendu et le comptage observé.
3. *On veut maintenant estimer la probabilité d'observer un mot un certain nombre n de fois dans une séquence de longueur ℓ. Utilisez les simulations pour estimer la probabilité empirique des mots `ATCTGC`, `ATATAT`, `TTTAAA` ou `AAAAAA`. Rappel : la probabilité empirique $p_{emp}(N \geq n)$ se calcule simplement comme la proportion de simulations où le mot est apparu au moins n fois.
4. **Dessiner des histogrammes de la distribution du comptage des mots. Remarquez vous des différences en fonction du mot ? Par exemple `ATATAT` et `TTTAAA` on la même probabilité d'apparaître, ont-ils la même distribution ?
5. *Comment peut-on calculer un intervalle de confiance pour cette probabilité empirique ?

3.3 Modèles de dinucléotides et trinucléotides

Une des limitations de la méthode proposée précédemment est qu'elle ne peut pas prendre en compte le fait que certaines combinaisons de nucléotides ont plus de chance d'apparaître que d'autres. On peut tenir compte de ces effets en utilisant un modèle de séquence aléatoire qui prend en compte les fréquences de dinucléotides chevauchant.

En plus de la table des nucléotides, on estime une table à 4 lignes et 4 colonnes M où l'élément $M(i,j)$ est la probabilité de la lettre j sachant qu'on est sur la lettre i.
Avec ce modèle, on génère une séquence de la manière suivante :

- On génère la première lettre suivant la table des fréquences de nucléotides (par exemple `G`)
- On génère ensuite chaque lettre suivante en fonction de la lettre courante avec la matrice M (par exemple la troisième ligne de M pour la seconde position).

1. Est ce que ce modèle correspond à une chaîne de Markov ? De quel ordre ? A quoi correspond la probabilité stationnaire ?
2. Ecrire une fonction qui estime M à partir des comptages des mots de longueur 2.
3. Ecrire une fonction qui simule une séquence de longueur ℓ avec le modèle de dinucléotides.
4. Calculez la probabilité d'apparition d'un mot à une position donnée (rappel : pour les fréquences de nucléotides, c'est le produit des probabilités des lettres).
5. *Quel est le nombre attendu d'occurrences pour un mot de longueur k avec le modèle de dinucléotides ?
6. *Comparez les comptages du nombre d'occurrences entre le modèle de nucléotides et de dinucléotides.
7. **En déduire la distribution du nombre d'occurrences quand les mots ne se chevauchent pas, l'appliquer aux séquences de régulation. Est ce que cela marche mieux dans le cas de PHO ?

3.4 Probabilités de mots

1. * On va calculer une approximation de la probabilité d'avoir un mot w qui apparait n fois dans une séquence aléatoire ($0 \leq n \leq \ell - k + 1$). On regarde ici des mots w qui ne sont pas chevauchant avec eux même (par exemple w=`ATCT` est ok, mais pas w=`CCTCC`). On fait l'hypothèse en première approximation que dans ce cas, les différentes positions d'occurrences du mot sont indépendantes et qu'on peut utiliser une loi binomiale pour le comptage. Donnez les paramètres de cette loi binomiale.
2. *Quelle approximation peut être faite avec une loi de Poisson ?
3. **Comparer la distribution de probabilité calculée pour les mots `ATCTGC`, `ATATAT`, `TTTAAA` ou `AAAAAA` avec l'histogramme de la probabilité empirique. Que remarquez vous ?
4. Ecrire une fonction qui, à partir de la liste des comptages des mots de taille k et de la longueur de la séquence, calcul leur probabilité d'occurrence $P(N_w \geq n_w)$ avec la formule analytique.
5. **Pour l'exemple de PHO, donnez les mots qui apparaissent significativement plus qu'attendu pour $k = 2, 4, 6$. Arrivez vous à reconstruire les séquences des sites de fixation ?