**Projet Annotation – Annotation d’un fragment génomique**

Annoter l'un des deux fragments génomiques ci-joint en utilisant les différentes approches vues en cours. Vous pouvez utiliser, pour la recherche de motifs, les motifs et matrices PWM établit sur *B. subtilis*.

Ce projet comprend deux parties conduisant à deux notes :

- la rédaction d’un rapport décrivant les résultats de votre annotation.

- la réalisation de trois programmes (en fournissant les algorithmes en pseudo-code) permettant de transformer le format de sortie des deux versions de GeneMark (GeneMark et GeneMark.hmm) et de scan\_for\_matches en format GFF.

**Rapport**

Dans votre rapport expliquez la problématique, décrivez votre démarche (pourquoi telle méthode, que vous permet-elle de faire, ses limites), présentez et analysez vos résultats. Il faudra donc réaliser un tableau de synthèse comme fait en TP et expliquez vos choix, comme, par exemple, celui des codons start retenus. Les résultats doivent inclure la prédiction fonctionnelle des produits des gènes identifiés sur le fragment génomique ainsi que celle de leur localisation cellulaire.

La prédiction fonctionnelle sera réalisée ensuite à partir des produits protéiques potentiels déduits de votre annotation. Cette annotation fonctionnelle comprendra la détection de domaines fonctionnels. Le découpage et la traduction des CDS peuvent être réalisés en utilisant la suite logicielle EMBOSS : le programme *extractseq* pour le découpage de votre séquence nucléique et le programme *transeq* pour la traduction du fragment découpé.

**Programmation**

Pour intégrer les résultats, il faudra réaliser plusieurs parsers pour mettre les résultats des programmes dans le format GFF compris par différents logiciels que vous trouverez décrit dans description des parsers : les résultats de GenMark, les résultats de GenMark.hmm et ceux de scan\_for\_matches concernant la prédiction des RBS et donc du codon initiateur, des promoteurs et des terminateurs de transcription. Le format de sortie de scan\_for\_matches étant le même pour n’importe quel signal recherché, un seul programme doit pouvoir permettre de parser les différents résultats de scan\_for\_matches.

**Consignes :** passer le nom des fichiers en argument de manière à ce que l’utilisateur puisse utiliser celui qu’il souhaite (ne pas coder les noms « en dur » dans le programme). De même, pour le parser de scan\_fro\_matches, la feature recherchée (RBS, promoteur, terminateur) doit être passée en argument pour que le même programme puisse être utilisé quel que soit la recherche réalisée.

Ne pas oublier d’afficher un message d’utilisation du programme quand celui-ci est lancé sans argument (cf. le message de scan-for-matches) de manière à ce que l’utilisateur du programme sache les informations à fournir au programme et dans quel ordre.

Comme les programmes à réaliser sont simples, pour vous habituer à utiliser des langages différents, il serait souhaitable que ces parsers soient écrits en Perl en utilisant le petit mémo qui est mis en ligne. La partie la plus importante est la création de l’algorithme, structure logique de votre programme avant son implémentation dans un langage de programmation. Cependant, l’utilisation du langage Python est acceptée.

Même si la sortie de GenMark.hmm est actuellement disponible au format GFF, je vous demanderai de bien vouloir réaliser le programme qui transforme la sortie LST en format GFF, ceci pour vous entraîner à la programmation.

Je vous demande aussi de me fournir les algorithmes en pseudo-code de chacun de vos programmes (cf. ci-dessous rapport). Vous trouverez les règles d’écriture en pseudo-code les plus usuelles par exemple sur le [site](https://s21182c7956f6fcf0.jimcontent.com/download/version/1518791103/module/16524888625/name/regles-pseudocodes.pdf)

Envoyer par courrier électronique :

* un rapport écrit qui doit être réalisé individuellement. Ce rapport doit justifier votre démarche et votre raisonnement, présenter les résultats obtenus à chaque étape et leur synthèse. Il devra intégrer les algorithmes des différents parsers (attention, pas le code, mais le pseudo-code du programme)
* Les fichiers de vos résultats qui ont été utilisé comme entrée de vos programmes (GenMark, scan\_for\_matches)
* Les fichiers de vos programmes (pas dans le rapport mais dans des fichiers séparés pour que je puisse les tester)