Recherche des gènes et des erreurs de séquençage dans les génomes bactériens GC-riches (et autres…)

Thomas Schiex, Patricia Thébault, Daniel Kahn

INRA, Toulouse

Résumé

Les génomes procaryotes GC-riches tels que ceux de *Ralstonia solanacerum* ou *Rhizobium meliloti*, posent des problèmes spécifiques d'identification de gènes : il semble que l'absence de codon STOP dans une phase donnée sur une longueur importante (par exemple 400 nucléotides) ne soit absolument pas caractéristique de l'existence d'un gène de protéine. On peut conforter cette hypothèse en générant des séquences "codantes" aléatoires au moyen de modèles de Markov estimés sur des séquences codantes de différents organismes. On observe alors que les sections codantes de génomes GC-riches tendent naturellement à produire des séquences sans STOP dans des phases différentes de la phase codante.

Cette caractéristique ne permet pas d'exploiter la notion de "long ORF" (*Open Reading Frame* ou phase ouverte de lecture) aussi simplement que pour les génomes bactériens moins GC-riches et rend délicate l'utilisation de logiciels réputés tels que Glimmer [5]. En reprenant les briques de base de Glimmer (modèles de Markov interpolés, calcul d'une énergie d'hybridation pour l'évaluation des START), nous avons construit un algorithme, incarné dans le logiciel FrameD, pour localiser les gènes et les erreurs de séquençage (frameshifts) dans les génomes bactériens GC-riches.

1 Génomes GC-riches et phases ouvertes de lecture

Nous avons fait face à deux génomes GC-riches : *Ralstonia solanacerum* avec un GC% de l'ordre de 67% et *Rhizobium meliloti* avec un GC% de l'ordre de 62%. Nous avons représenté ci-dessous, et dans les 6 phases de lecture possibles, les sous-séquences maximales, démarrant par un START (ATG, GTG ou TTG), faisant plus de 300 nucléotides de long et ne contenant pas de STOP en phase pour une section typique du génome de *Rhizobium meliloti*. On observe un nombre important de superpositions de 2, 3 voire 4 ORF de longueur de plus de 300 nucléotides.

Or, il semble généralement admis que l'existence d'une phase ouverte de lecture de longueur importante est le signe évident de l'occurrence d'un gène de protéine. Cela semble inexact dans le cas de nos deux génomes. Une première explication possible réside dans le fait que les codons STOP (TAA, TAG, TGA) sont GC-pauvres. Une explication complémentaire réside dans le fait que le code génétique est essentiellement redondant dans le troisième nucléotide des codons. La GC-richesse conjointement à la contrainte de codage entraîne une utilisation plus forte encore de GC en troisième position des codons (eg. sur *Ralstonia solanacerum*, on dépasse 86% de GC). L'appauvrissement en STOP sur les phases opposées devient alors sensible, et ce particulièrement

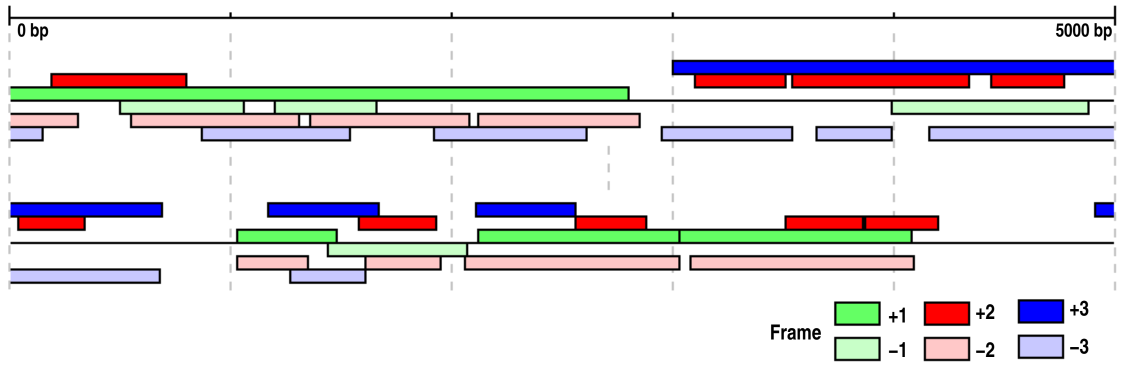


FIG. 1 – Visualisation des ORF de plus de 300 pb. sur *Rhizobium meliloti*

dans la phase où cette troisième position directe correspond à la première position d'un codon sur le brin complément (tous les codons STOP commençant par T).

Afin de conforter ces hypothèses à moindre coût, nous avons construit pour chacun de nos 2 génomes ainsi que pour un génome plus classique dans son taux de GC (*Bacillus subtilis*, GC% de l'ordre de 44%) deux modèles probabilistes :

- un modèle de Markov d'ordre 0 (M0), censé caractériser le non-codant, et utilisant les proportions GC/AT de chacun des organismes ;
- un modèle de Markov interpolé [5] d'ordre 8 et 3-périodique (M8), estimé sur des séquences de gènes entiers de chacun des génomes, contenant le codon STOP final.

Ces modèles permettent de générer aléatoirement des séquences dans le style non-codant (M0) ou codant (M8) à volonté pour ensuite estimer des statistiques simples telles que la longueur moyenne, en codons, des sous-séquences maximales ne contenant pas de STOP et ce pour chacun des modèles et des organismes. Les résultats de cette analyse sur des séquences de 3Mb apparaissent dans la table ci-dessous, dans un ordre croissant de GC% :

Modèle	GC%	Phase 1	Phase 2	Phase 3	Phase -1	Phase -2	Phase -3
Bacil. M0	44.4	17.85	17.82	17.95	17.93	17.89	18.01
Bacil. M8		279.25	16.76	14.14	26.61	25.92	17.85
Rhizo M0	62.0	34.21	34.36	34.41	33.94	34.20	34.23
Rhizo M8		335.57	33.25	69.90	153.66	37.29	26.01
Ralsto M0	67.2	44.63	44.71	44.69	44.32	44.13	43.90
Ralsto M8		357.65	32.60	125.57	410.85	56.01	33.12

TAB. 1 – Longueur moyenne en codons des sous-séquences maximales sans STOP

Analysons ces résultats :

1. Les modèles M0 confirment le fait que la richesse en GC seule tend à induire un relatif appauvrissement en codon STOP, dans toutes les phases. Les longueurs moyennes des sous-séquences maximales sans STOP valent successivement $18 < 34 < 44$ codons pour une GC-richesse croissante.
2. Mais ce phénomène reste négligeable devant l'influence de l'utilisation du style codant des génomes, tel qu'il a été capturé par les modèles de Markov 3-périodiques M8. La phase la

plus affectée est sans conteste la phase -1 pour laquelle les longueurs moyennes des sous-séquences maximales sans STOP valent successivement $26 < 154 < 411$ codons pour une GC-richesse croissante. La longueur de la séquence étant de 3Mb (i.e., un multiple de 3) cette phase -1 est précisément la phase du brin complément pour laquelle le premier nucléotide des codons correspond au troisième nucléotide des codons en phase 1. Dans le cas de génomes très GC-riches, comme *Ralstonia solanacerum*, la longueur moyenne de ces sous-séquences maximales sans STOP en phase -1 est même plus importante qu'en phase 1.

Ces résultats corroborent donc, à moindre coût, l'hypothèse que les contraintes de codage et de GC-richesse entraînent conjointement la création de longues ORF qui sont des "miroirs" des véritables gènes. Naturellement, de telles ORF ont généralement un caractère non codant.

Malheureusement, de nombreux logiciels de prédictions de gènes procaryotes, tel que Glimmer par exemple, s'appuient de façon cruciale sur cette notion de phase de lecture ouverte de grande taille et produisent des résultats de mauvaise qualité sur ces génomes GC-riches. Nous avons donc été amenés a construire un outil d'annotation qui reprend les ingrédients de base de Glimmer mais qui s'appuie plus fortement sur les modèles de Markov interpolés pour décider si une ORF est ou n'est pas un gène.

2 L'outil FrameD

Les ingrédients de base de Glimmer sont :

- un modèle de Markov interpolé 3-périodique qui est utilisé pour estimer le caractère codant dans chacune des 6 phases.
- un modèle de Markov d'ordre 0 qui est utilisé pour modéliser les sections non codantes.

Étant donnée une ORF $O = (n_1, \ldots, n_\ell)$, les modèles de Markov peuvent être utilisés pour calculer la vraisemblance $V_k(O)$ de l'ORF O dans chaque phase $k \in \{-1, -2, -3, 1, 2, 3\}$ et selon le modèle non-codant ($k = 0$). Le fonctionnement de Glimmer dépend alors de la longueur de l'ORF :

- pour une ORF courte, les $V_k(O)$ sont normalisées pour construire un score $S(O)$:

$$S(O) = \frac{V_1(O)}{\sum\limits_{j \in \{-3,-2,-1,0,1,2,3\}} V_j(O)}$$

- pour une ORF assez longue, le modèle non-codant n'est pas utilisé dans la normalisation :

$$S(O) = \frac{V_1(O)}{\sum\limits_{j \in \{-3,-2,-1,1,2,3\}} V_j(O)}$$

  On considère donc implicitement que toute ORF longue est codante.

Si la valeur de $S(O)$ est supérieure à un seuil de 0.9, alors l'ORF est considérée comme gène potentiel. Dans le cas de génomes GC-riches, cette pratique conduit à une très faible spécificité car, comme on l'a vu, il peut y avoir des ORF longues et cependant non codantes.

Pour améliorer le choix du START, Glimmer permet aussi d'utiliser le résultat d'un calcul d'énergie d'hybridation après alignement avec un motif RBS (Shine-Delgarno) en amont de chaque START. Par défaut, cette information n'est pas utilisée par Glimmer et le premier START de l'ORF est choisi mais si l'on spécifie l'option correspondante, Glimmer sélectionnera, pour chaque ORF de score suffisant, le premier START de qualité supérieure à un seuil fixé.

L'approche que nous avons suivie reprend les ingrédients de base de Glimmer et n'est pas sans ressemblance avec une approche de type HMM (chaînes de Markov cachées). Elle est aussi similaire à l'approche que nous avons utilisée dans une version eucaryote, plus complexe, appelée EuGène [6]. Dans une première version, on construit un graphe dirigé sans circuit tel que tout chemin dans ce graphe correspond à une détection de gènes possible (cohérente avec les règles d'usage des codons START et STOP, les gènes peuvent être partiels). On supposera dans un premier temps que les gènes ne se superposent pas. Le graphe ci-dessous constitue un exemple du graphe considéré sur une séquence courte. Il comprend 7 pistes en parallèle qui correspondent respectivement et de bas en haut à l'hypothèse que l'on se trouve dans une région codante en phase -3, -2 et -1, dans une région non codante ou enfin dans une région codante en phase 1, 2 et 3. Chaque nucléotide n_i de la séquence est alors représenté par 7 arêtes, une par piste (situées sous chaque nucléotide dans la figure 2).

Les signaux (START, STOP) qui apparaissent sur la séquence sont représentés par des « aiguillages » entre ces pistes. On considère que les signaux sont ponctuels et apparaissent entre deux nucléotides :

- l'existence d'un START en phase i engendre la création d'une arête permettant de passer de la piste centrale « non-codant » à la piste « codant en phase i » à la position du START.
- un signal STOP en phase i engendre la création d'une arête permettant de passer de la piste « codant en phase i » à la piste « non-codant » et supprime de plus la possibilité de continuer sur la piste « codant en phase i » par suppression d'une arête.

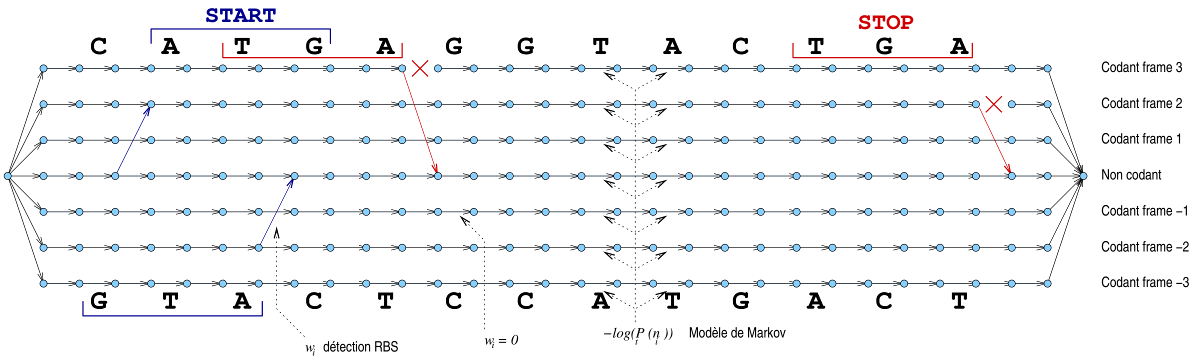


FIG. 2 – Le graphe dirigé sans circuit considéré pour CATGAGGTACTGA

On vérifie bien que tout chemin qui va de la source (sommet initial à gauche) au puits (sommet terminal à droite) correspond à une localisation de gènes possible. Pour différencier les nombreux chemins qui peuvent exister dans ce graphe, on pondère les arêtes du graphe. Si l'on appelle longueur d'un chemin la somme des poids des arêtes qui forment le chemin, on va chercher à pondérer les arêtes de façon à ce qu'un chemin de longueur minimum entre la source et le puits représente une localisation de gènes la plus cohérente avec les données :

- l'arête qui représente le nucléotide n_i sur la piste k est pondérée par l'opposé du logarithme de la probabilité que ce nucléotide apparaisse dans la phase correspondante. La longueur

d'un chemin sur la piste k, somme de ces logarithmes élémentaires, sera ainsi égale à l'opposé du logarithme de la vraisemblance de la séquence parcourue en pahse k. Plus cette vraisemblance est faible, plus la longueur du chamin sera importante.
– les aiguillages représentant les signaux reçoivent une pondération liée à la qualité du signal et dont le calcul est détaillé ci-dessous.

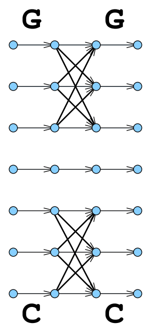
Ce premier modèle, très simple, est proche d'un modèle de Markov caché (interpolé) avec une matrice de transition entre états cachés qui n'est pas homogène. Les poids des arêtes START sont construites à partir de l'estimation de l'énergie G de l'hybridation optimale d'une section amont à chaque START avec un motif RBS. On utilise une forme paramétrique simple $w = \alpha(G-\beta)$, les constantes α et β étant optimisées sur des séquences déjà annotées. Le critère maximisé est alors le nombre de gènes correctement reconnus (y compris le choix d'un START correct). Le poids lié à un STOP n'est pas, comme cela serait le cas dans un modèle HMM, fixé à $(0 = -\log(1.0))$ mais est aussi estimé par maximisation du même critère et prend typiquement une valeur correspondant à une probabilité sensiblement inférieure à 1. Dans une analogie HMM, la non normalisation de la matrice de probabilités conditionnelles pourrait correspondre à l'existence d'un état caché supplémentaire de "rejet". Ce poids, utilisé pour toutes les arêtes représentant un STOP, permet de jouer sur le rapport sensibilité/spécificité. Il évite également la détection d'ORF trop courtes, irréalistes.

La recherche d'un plus court chemin dans ce graphe dirigé sans circuit se fait par un des premiers algorithmes de programmation dynamique, l'algorithme de Bellman (remontant à 1959, cf. [3]) et permet d'obtenir une annotation en temps et espace linéaires dans la longueur de la séquence. Mais ce modèle reste trop simple en pratique pour donner de bons résultats sur des séquences brutes en particulier du fait des erreurs de séquençage et des superpositions fréquentes de paires de gènes.

2.1 Prise en compte et détection d'erreurs de séquençage

La prise en compte d'erreurs de séquençage dans le modèle précédent est simple à mettre en œuvre. On considère uniquement les erreurs de type "frameshift" dues à l'insertion ou à la délétion de nucléotides. Un frameshift consistera juste à changer de phase de lecture : on ajoute donc au graphe précédent les arêtes correspondantes avec un poids initial important correspondant à une probabilité *a priori* faible de frameshift. Ce poids a été ajusté manuellement. Un agrandissement du graphe entre deux nucléotides est visible ci-dessous.

Ce modèle permet la détection de frameshifts et complète assez bien l'approche proposée dans [2] qui s'appuie sur la détection de plusieurs homologies avec une même séquence protéique mais sur des phases différentes. La structure de graphe sans circuit dirigé se prête de plus à de nombreux calculs via des algorithmes de programmation dynamique utilisant d'autres opérateurs que $\max$ dans la récurrence [1] : ils permettent des calculs de fiabilité, probabilité de passage par une arête…et restent linéaire en temps et en espace. FrameD calcule ainsi, pour chaque paire de nucléotides adjacents, l'espérance de passage par chacune des 31 arêtes qui peuvent séparer ces deux nucléotides :



– les 7 arêtes dont le parcours consiste à rester dans le même état ;
– les 2×6 arêtes (pas toujours présentes) liées à l'occurrence de codons START et de STOP ;

– les 12 arêtes de frameshift.

Après normalisation, ce calcul permet, à chaque position, de fournir un indicateur de présence potentielle d'un frameshift ainsi qu'une prédiction "moyenne" en sus de la prédiction optimale fournie par l'algorithme de plus court chemin. Aucun seuil théorique n'a été calculé pour la détection de frameshift.

2.2 *Superposition de gènes*

Une seconde extension consiste à permettre la superposition de gènes. Il suffit pour cela d'introduire des pistes supplémentaires correspondant à la coexistence de deux gènes dans des phases données. Si toutes les superpositions sont autorisées, cela conduit à l'ajout de 15 nouvelles pistes. Pour l'instant, nous nous sommes limités à la superposition de gènes sur un même brin qui conduit à l'ajout de 6 pistes seulement. L'apparition de superpositions n'est pas rare mais ces superpositions sont généralement courtes, la plus fréquente correspondant à la séquence `ATGA` contenant un START (`ATG`) en phase 1 et un STOP (`TGA`) en phase 2. Il est donc difficile voire impossible d'accumuler assez de séquences pour estimer un modèle de Markov significatif pour ces états doubles. La courte durée de ces états ne rend pas non plus cruciale la qualité de l'estimation. Dans la pratique, la probabilité d'être dans un tel état est fixée, de façon arbitraire, à la moyenne géométrique des probabilités des deux états.

3 Utilisation

FrameD est actuellement utilisé de façon routinière pour traiter les séquences des deux bactéries *Ralstonia solanacerum* et *Rhizobium meliloti* dans les deux phases de recherche d'erreurs de séquençage et de recherche de gènes de protéines. FrameD offre des sorties textes et graphiques (sous la forme de fichiers GIF ou PNG). La sortie graphique, qui peut inclure également des informations d'homologies avec des bases de données de protéines permet une analyse rapide par l'expert.

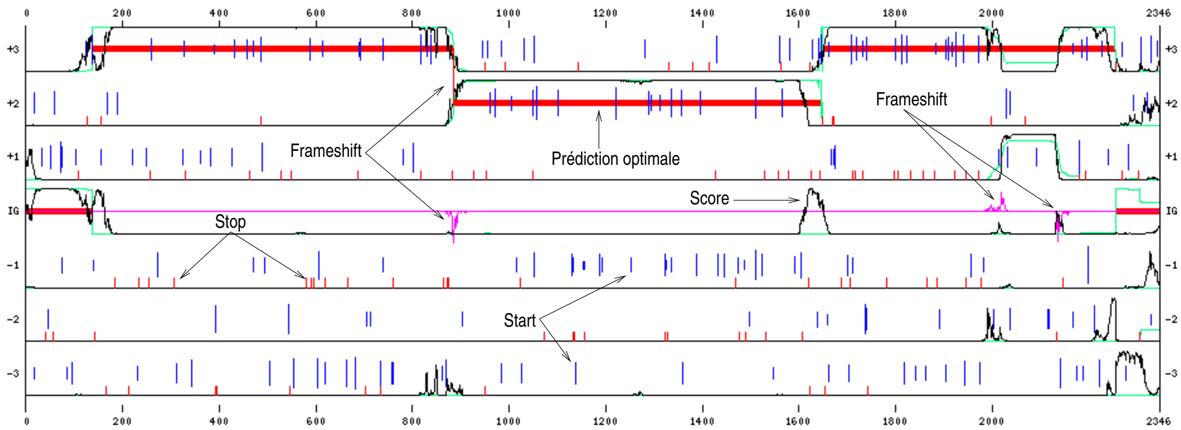


FIG. 3 – Sortie graphique de FrameD

Un exemple de sortie graphique sur une partie d'un BAC de *Rhizobium meliloti* est visible sur la figure 3. L'axe des X représente la séquence, l'axe des Y présente les différents états possibles pour la séquence : codante dans les 6 phases possibles ou non codante. Sur les 6 phases codantes, les traits verticaux indiquent l'occurrence de codons STOP (en bas) et START. La hauteur des barres représentant les codons START est liée à la qualité d'un éventuel RBS en amont. Sur chaque ligne, une courbe indique les scores des modèles de Markov interpolés, lissés sur une fenêtre glissante d'une centaine de nucléotides et normalisés. Le trait horizontal épais indique la prédiction. Enfin, au niveau non-codant, une courbe indique l'espérance de passage, sur toutes les prédictions possibles, par les arêtes de frameshifts (insertions vers le haut, délétions vers le bas). Sur cette séquence, 3 frameshifts ont été créés artificiellement :

1. un nucléotide a été effacé un peu avant la position 900. Un frameshift est effectivement détecté à cette position : la prédiction optimale, qui indiquait un gène en phase 3 se poursuit en phase 2. De plus, la courbe d'espérance de frameshift fait apparaître un pic vers le bas à cette position, signe d'une délétion.
2. deux frameshifts qui se compensent ont été créés plus loin sur la séquence (entre 2000 et 2200 environ). Ici, la prédiction optimale ne passe pas par ces deux frameshifts, mais la courbe d'espérance de frameshift met en évidence une insertion suivie d'une délétion.

En pratique, l'analyse de la sortie texte permet la détection automatique de zones où un frameshift apparaît potentiellement. Le seuil à partir duquel une vérification des traces est demandée a été déterminé par tâtonnement. Par manque de temps, aucune simulation n'a encore été effectuée pour estimer quantitativement la sensibilité de FrameD. Qualitativement, son utilisation routinière a permis la détection de plusieurs dizaines d'erreurs de séquençage pour un travail de contrôle qui reste limité et qui débute simplement par l'analyse de la sortie graphique de FrameD.

En ce qui concerne les performances de détection de gènes sur les génomes GC-riches, le gain obtenu via l'utilisation de FrameD par rapport à Glimmer 1.03, en particulier en termes de spécificité, est très net. L'exemple suivant, sur une séquence de *Ralstonia solanacerum*, est typique des comportements des deux logiciels sur des génomes GC-riches.

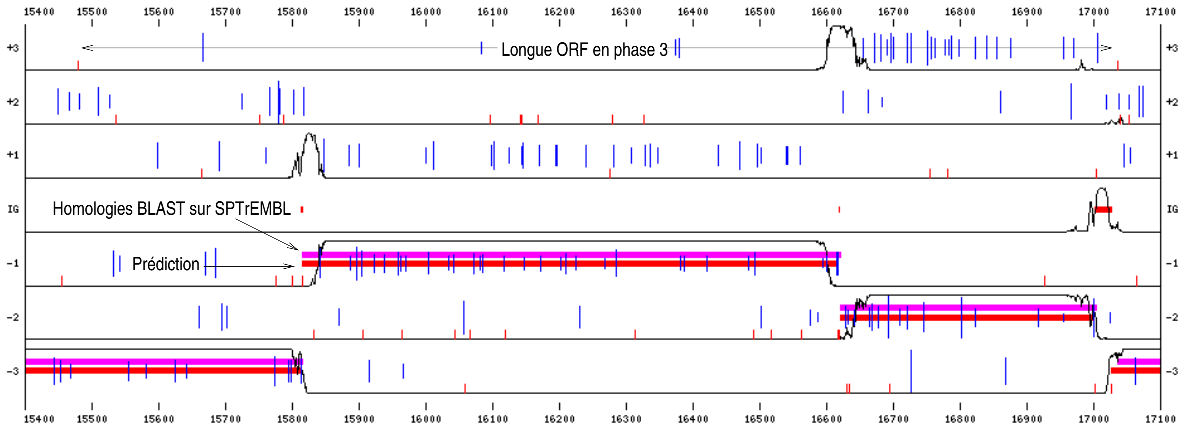


FIG. 4 – Prédiction de FrameD sur une séquence de *Ralstonia solanacerum*

La sortie graphique de FrameD sur cette séquence est visible sur la figure 4. En parallèle avec les prédictions de FrameD sont visualisées les zones ayant des homologies significatives ($e < 10^{-6}$) avec SwissProt/TrEmbl[1]. On notera la longue ORF en phase 3, sans aucune homologie.

[1]FrameD peut, si l'utilisateur le souhaite, utiliser ces informations d'homologies dans sa prédiction. Ce n'est pas le cas dans cet exemple où seule une visualisation des homologies est réalisée.

La liste finale des gènes putatifs de Glimmer 1.03, utilisant des modèles de Markov estimés sur le même jeu de séquence, contient tous les gènes prédits par FrameD mais également un gène putatif qui va de de 15663 à 17039 en phase 3 (qui recouvre deux autres gènes). L'ORF O correspondant à cette seconde prédiction contenant des STOP dans les autres phases, les vraisemblances $V_k(O)$ sont nulles pour $k \neq 3$ et on aboutit à un score $S(O)$ égal à 1.0 : l'ORF est sélectionnée alors que le score normalisé des modèles de Markov est désespérément plat dans cette phase. Il faut noter que cette faiblesse de Glimmer 1.03 est confirmée par la publication très récente [4] qui introduit Glimmer 2.0, censé corriger ces faiblesses sur les génomes GC-riches.

Comment FrameD se comporte-t-il sur des génomes qui ne sont pas GC-riches ? Nous avons effectué un test rapide sur une partie du génome de *Bacillus subtilis*, généralement considéré comme un génome annoté de façon fiable. Ce test conduit à une sensibilité[2] et une spécificité[3] au niveau nucléotide de l'ordre de 99%. Au niveau gène, la sensibilité observée de FrameD est de 98% et sa spécificité de 94%. De plus, 83% des gènes reconnus sont identifiés avec un START en accord avec les annotations. À titre indicatif, Glimmer 1.03 affiche la même sensibilité de 98% sur *Bacillus subtilis* [4]. Sa spécificité et la qualité de choix des STARTs n'est pas mentionnée par ses auteurs.

L'apport essentiel de FrameD, au-delà de ses bonnes capacités à détecter les gènes, réside dans sa faculté de détection de frameshifts potentiels même sans information d'homologie. En dehors de son utilisation sur des génomes bactériens, FrameD peut également être utilisé sur des génomes eucaryotes, pour chercher les erreurs de séquençage dans les séquences d'ADN codant. L'inclusion de ce mécanisme de détection d'erreur de séquençage dans le prototype de localisation de gènes eucaryotes (EuGène [6]) constitue une prochaine étape qui pose des problèmes algorithmiques spécifiques.

Remerciements : Nous remercions Jérôme Gouzy (INRA, Toulouse) sans qui FrameD ne serait probablement pas né et n'aurait certainement pas atteint son état actuel.

Références

[1] U. BERTELÉ AND F. BRIOSHI, *Nonserial Dynamic Programming*, Academic Press, (1972).

[2] N. P. BROWN, C. SANDER, AND P. BORK, (1998), *Frame : detection of genomic sequencing errors*, BioInformatics, 14, pp. 367–371.

[3] T. H. CORMEN, C. E. LEISERSON, AND R. L. RIVEST, *Introduction to algorithms*, MIT Press, (1990). ISBN : 0-262-03141-8.

[4] A. L. DELCHER, D. HARMON, S. KASIF, O. WHITE, AND S. L. SALZBERG, (1999), *Improved microbial gene identification with* GLIMMER, Nucleic Acids Res., 27, pp. 4636–4641.

[5] S. L. SALZBERG, A. L. DELCHER, S. KASIF, AND O. WHITE, (1998), *Microbial gene identification using interpolated Markov models*, Nucleic Acids Res., 26, pp. 544–548.

[6] T. SCHIEX, A. MOISAN, L. DURET, AND P. ROUZÉ, (1999), *EuGène : A simple yet effective gene finder for eucaryotic organisms (Arabidopsis thaliana)*, in Proc. of the $2^n d$ Georgia Tech International Conference on Bioinformatics - In silico Biology, Atlanta.

[2]Pourcentage des éléments codants annotés comme tels.

[3]Pourcentage des éléments annotés codants qui le sont effectivement.