GenRGenS : génération aléatoire de séquences génomiques

GenRGenS : Generation of Random Genomic Sequences

Yann Ponty[†‡]

[†] LRI, bât.490, Université Paris-Sud, 91405 Orsay cedex
[‡] IGM, équipe BIG, bât.400, Université Paris-Sud, 91405 Orsay cedex

Courriel : dev.GenRGenS@lri.fr

Résumé

GenRGenS est un logiciel de génération aléatoire de séquences et de structures génomiques, selon des modèles markoviens ou selon des grammaires non contextuelles pondérées.

Mots clés : *séquences aléatoires, grammaires non contextuelles, grammaires pondérées, chaînes de Markov.*

1 Contexte

Un des problèmes importants liés à l'analyse *in silico* des génomes génomiques est l'évaluation du "bruit de fond". Il s'agit de déterminer dans quelle mesure des structures ou des propriétés observées sur des séquences biologiques déduites ont une réelle signification ou sont là "par hasard". Le raisonnement qui prévaut est le suivant : si une propriété est réellement signifiante d'un point de vue biologique, alors elle ne doit pas se manifester dans une séquence aléatoire, ou s'y manifester de façon très différente par rapport à une séquence biologique. Dans certains cas, comme l'évaluation de la sur-représentation de motifs, il existe des méthodes purement analytiques pour comparer séquences biologiques et séquences aléatoires. Dans d'autres cas, il est nécessaire de recourir à la simulation, c'est-à-dire la génération aléatoire *in silico* de séquences. Il s'agit d'engendrer des séquences de longueur n fixée, obéissant à un modèle déterminé.

2 GenRGenS

GenRGenS (Generation of Random Genomic Sequences) est une boîte à outils pour la génération aléatoire de séquences selon différents types de modèles. D'un usage simple (Interface Graphique) et développé en Java, ce qui le rend exécutable sur toutes les plateformes, il est destiné à des bioinformaticiens sensibilisés aux formalismes linguistiques et/ou statistiques. GenRGenS est téléchargeable avec ses sources dans sa version 1.0 à l'adresse :

`www.lri.fr\ ~denise\GenRGenS`

Les modèles de génération actuellement supportés par GenRGenS sont le modèle markovien et le modèle des grammaires non contextuelles pondérées.

2.1 Modèle markovien

Ce modèle bien connu impose des contraintes statistiques aux occurrences de sous-séquences dans les séquences engendrées. Il est paramétré par un entier, l'orde de la chaîne de Markov, qui détermine la taille des sous-séquences considérées. GenRGenS permet de gérer des chaînes de Markov hétérogènes à plusieurs phases (cas des séquences codant pour des protéines) et certains modèles de Markov cachés. La figure 1 présente un exemple de *fichier de description* qui est fourni en entrée à GenRGenS pour engendrer des séquences aléatoires markoviennes. Un tel fichier de description peut être réalisé automatiquement par GenRGenS si on lui fournit une séquence de référence et certains paramètres comme l'ordre de la chaîne et le nombre de phases.

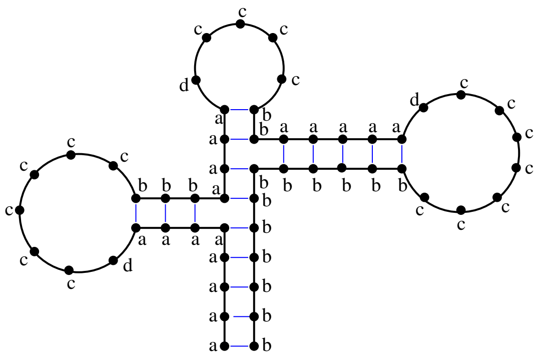
2.2 Génération avec contraintes syntaxiques

Les grammaires formelles permettent de décrire des séquences soumises à des contraintes structurelles, comme par exemple la présence ou l'absence de motifs particuliers dans les séquences, ou des interactions à distance dans les séquences d'ARN. Ainsi, le codage classique des structures secondaires par des *mots de Motzkin* [3, 2], selon lequel un nucléotide non apparié est représenté par une lettre `c` et les deux nucléotides $5'$ et $3'$ d'un appariement sont respectivement codés par un `a` et un `b`, définit un langage qui peut être décrit et engendré par une grammaire non contextuelle (figure 2). Le concept de *grammaire pondérée*, présenté dans [1], permet d'engendrer des séquences ainsi structurées tout en respectant des contraintes statistiques données portant sur les fréquences des lettres. A chaque lettre est attribuée un *poids* qui influe sur son nombre moyen d'occurrences dans les séquences engendrées.

```
TYPE = MARKOV
ORDER = 2
START =
  ggg 7
  cca 3
FREQUENCIES =
  aaa 00 aca 50 aga 12 ata 56       caa 23 cca 86 cga 10 cta 10
  aac 00 acc 78 agc 12 atc 23       cac 15 ccc 78 cgc 21 ctc 51
  aag 00 acg 51 agg 51 atg 23       cag 46 ccg 46 cgg 32 ctg 35
  aat 00 act 32 agt 53 att 02       cat 91 cct 45 cgt 51 ctt 11

  gaa 21 gca 84 gga 51 gta 31       taa 86 tca 84 tga 02 tta 61
  gac 12 gcc 64 ggc 51 gtc 51       tac 86 tcc 43 tgc 23 ttc 31
  gag 51 gcg 61 ggg 35 gtg 18       tag 48 tcg 48 tgg 35 ttg 18
  gat 86 gct 43 ggt 84 gtt 68       tat 65 tct 56 tgt 81 ttt 81
```

FIG. 1 – Un fichier de description GenRGenS pour un modèle markovien d'ordre deux. La clause START permet de déterminer une loi pour le début de la séquence : elle commencera par ggg 7 fois sur 10, par cca 3 fois sur 10. Dans la clause FREQUENCIES, le nombre qui suit chaque trinucléotide est le nombre d'occurrences observées dans une séquence biologique de référence. Les probabilités associées au modèle sont calculées en conséquence par GenRGenS.



```
TYPE = GRAMMAR
SYMBOLS = LETTERS
RULES
   S -> a S b S;
   S -> a S b;
   S -> d T c;
   T -> c T;
   T -> c c c;
WEIGHTS =
   a 16
   c 2
   d 1
```

FIG. 2 – Une structure secondaire qui correspond au mot `aaaaaaadccccccbbaaaadccccbbaaaaadcc–cccccbbbbbbbbbbbb`, et une grammaire pondérée du langage. Au codage classique on a ajouté la lettre `d` qui permet, par le biais des pondérations, de faire varier le nombre moyen de boucles dans les structures engendrées aléatoirement. Plus le poids d'une lettre est élevé, plus sa fréquence est élevée dans les mots engendrés. Ici, la forte pondération de `a` indique que l'on désire favoriser les longueurs des tiges par rapport à celles des boucles.

3 Perspectives

Bientôt, GenRGenS s'enrichira d'autres modes de génération :

1. Un mode "chainé" qui permettra de cumuler différents niveaux d'abstractions. Par exemple, la structure d'une séquence pourra être définie par une grammaire de "haut niveau" quand son détail sera défini par des liguistiques markoviennes.
2. Des modes permettant d'engendrer des séquences aléatoires à partir d'expressions régulières ou PROSITE, de profils HMM et de profils généralisés.

Références

[1] DENISE (A.), ROQUES (O.) et TERMIER (M.), « Random generation of words of context-free languages according to the frequencies of letters », dans GARDY (D.) et MOKKADEM (A.), *Mathematics and Computer Science : Algorithms, Trees, Combinatorics and probabilities*, Trends in Mathematics, Birkhaüser, 2000, p. 113–125.

[2] SEARLS (D.), « Formal language theory and biological macromolecules », *Series in Discrete Mathematics and Theoretical Computer Science*, *47*, 1999, p. 117–140.

[3] VAUCHAUSSADE DE CHAUMONT (M.) et VIENNOT (X.), « Enumeration of RNA's secondary structures by complexity », dans CAPASSO (V.), GROSSO (E.) et PAVEN-FONTANA (S.), *Mathematics in Medecine and Biology*, volume 57 de *Lecture Notes in Biomathematics*, 1985, p. 360–365.