GenRGenS: Software for Generating Random Genomic Sequences and Structures

Yann Ponty, Michel Termier, Alain Denise

BIOINFORMATICS APPLICATIONS NOTE

Vol. 00 no. 00 2006
Pages 1–2

GenRGenS: Software for Generating Random Genomic Sequences and Structures

Yann Ponty [a], Michel Termier [b], Alain Denise[a*]

[a]LRI, UMR CNRS 8623, Université Paris-Sud 11, F91405 Orsay cedex, France,
[b]IGM, UMR CNRS 8621, Université Paris-Sud 11, F91405 Orsay cedex, France

ABSTRACT

Summary: GenRGenS is a software tool dedicated to randomly generating genomic sequences and structures. It handles several classes of models useful for sequence analysis, such as Markov chains, Hidden Markov models, weighted context-free grammars, regular expressions and PROSITE expressions. GenRGenS is the only program that can handle weighted context-free grammars, thus allowing the user to model and generate structured objects (such as RNA secondary structures) of any given desired size. GenRGenS also allows the user to combine several of these different models at the same time.

Availability: Source and executable files of GenRGenS (in Java) and the complete user's manual are freely available at http://www.lri.fr/bio/GenRGenS.

Contact: dev.GenRGenS@lri.fr

1 INTRODUCTION.

Random sequences can be used to extract relevant information from biological sequences. The random sequences represent the "background noise" from which it is possible to differentiate the real biological information. Random sequences are widely used to detect over-represented and under-represented motifs, or to determine whether the scores of pairwise alignments are relevant. Analytic approaches exist for solving these kinds of problems (see *e.g.* Reinert *et al*. (2000).) although for the most complex cases, an experimental approach (*i.e.* the computer generation of random sequences) is still necessary.

Some programs are currently available for generating random sequences. For example, the GCG package contains a few generation tools, such as *HmmerEmit* that generates sequences according to HMM profiles, and *Corrupt* that adds random mutations to a given sequence (Butler, 1998). Seq-Gen randomly simulates the evolution of nucleotide sequences along a phylogeny (Rambaut and Grassly, 1997). The Expasy server has *RandSeq*, which generates random amino acid sequences according to a Bernoulli process (Gasteiger *et al*., 2003). Shufflet is a program that generates random shuffled sequences (Coward, 1999). However, until now, there has been no software package that can integrate several statistical and syntactical models of random sequences and combine them. This is the purpose of GenRGenS. Typically, GenRGenS can help answering questions such as "what is the expected score of an alignment of two random RNA secondary structures?" or "what is the probability of a random coding-like sequence to contain a given structure?", allowing to test the relevance of given hypotheses.

*to whom correspondence should be addressed.

2 PROGRAM FEATURES.

2.1 Generalities.

GenRGenS can be run two ways: either via a graphical user interface or via the command-line. The random sequence models are passed to the main generation engine through *description files*. These are text files that obey certain simple syntax criteria. Several example description files for various models can be downloaded from the main GenRGenS website. For Markov models, a description file can be built automatically from a sequence using the *BuildMarkov* tool bundled with GenRGenS.

Sequences are generated by the user by simply specifying the name of a description file together with some additional parameters, such as the number of desired sequences and their length. GenRGenS then generates the sequences either on the screen or in a specified file.

The random sequence models currently handled by GenRGenS are the following:

- Markov Chains and Hidden Markov Models (HMM),
- Weighted context-free grammars,
- PROSITE patterns and rational expressions,
- Hierarchical models, *i.e.* compositions of several of the above models.

Below, we give the main characteristics of these models. More details can be found on the GenRGenS web site.

2.2 Markovian models.

Markovian sequence models are probably the most widely used for genomic purposes. These random sequences are generated letter by letter, with the probability of drawing a letter at each step depending on the subset of letters previously generated.

Homogenous Markov chains: In a homogenous model of order k, the probability of a letter only depends on the k previous letters. The parameters of these models (*i.e.* the probabilities of the transition matrix) are usually computed according to a reference biological sequence. These probabilities depend on the number of occurrences of words of length $k+1$ in this sequence. Essentially, the description file of a Markovian model contains the number of occurrences of these words. GenRGenS computes the transition matrix from this and can then generate any number of sequences

of a given length. The description files can be automatically built from a reference sequence in FASTA format using *BuildMarkov*, a tool bundled with GenRGenS.

"Framed" Markov chains: In a framed model, a sequence is virtually split into blocks of a given size, called *frames*. The probability of generating a letter now depends both on the k previous letters, and on its position in the *frame*. A framed model should be used if the user wants to generate coding-like DNA sequences, because the probability of a base also depends on its position inside a codon. Therefore, the user can define an integer parameter that determines the number of positions within the frame (3 for coding DNA), and thus the number of different transition matrices for the Markov chain.

Hidden Markov models: A HMM can be considered as a two-level Markov model. When generating sequences, this model aims to simulate regions having different statistical distributions. For example, this allows the modelling of alternating coding-like and non-coding-like regions in random sequences. As for ordinary Markov chains, GenRGenS provides a tool that can build a description file for framed Markov chains and for HMM's from a reference biological sequence and some additional parameters.

2.3 Weighted context-free grammars.

Context-free grammars allows both sequential and structural constraints of sequences to be modelled. In particular, they can handle certain kinds of long-range interactions such as base pairings in secondary RNA structures (Searls, 1999). Stochastic context-free grammars (SCFG's) have long been used to model the structural properties of genomic sequences, particularly for predicting the structure of sequences or for searching for motifs (Durbin *et al*., 1998). SCFG's can also be used to generate random sequences. However, they do not allow the user to fix the length of these sequences.

The grammatical features of GenRGenS are based on the similar yet different concept of *weighted* context free grammars (WCFG) (Denise *et al*., 2000). Briefly, a WCFG is a context-free grammar in which a *weight* is associated to each terminal symbol. The weight of a word is the product of the weights of its letters. Given a WCFG and an integer n, GenRGenS can draw words of length n that obey the grammar, such that the probability of any generated word is proportional to its weight. This allows WCFGs to be defined in which the grammatical part defines the structure of the words and the weights allow the user to fix some statistical properties, such as the expected number of loops or bulges, or other structural elements in an RNA family.

The algorithm used in GenRGenS to generate these sequences is based on the well-known algorithm by Flajolet *et al*. (1994), used to generate combinatorial structures uniformly at random.

2.4 PROSITE patterns and rational expressions.

GenRGenS can generate random sequences from a rational expression or a PROSITE pattern. It is well known that the expressivity of these formalisms is included in context-free grammars. Thus, any rational expression or PROSITE pattern can be described by a context-free grammar. However, there are more efficient generation algorithms available for these subclasses. Moreover, GenRGenS support for PROSITE patterns allows a convenient *copy/paste* approach. As with WCFGs, a weight can be assigned to each letter, enabling certain statistical properties to be fixed.

2.5 Hierarchical models

This class of models has been included to allow different models to be combined. For example, a simple model for the Intergenic/ORF alternation could describe the framed aspect of the ORFs (using the Framed Markov model) as well as the lack of a start codon in the intergenic areas (using the Rational expression model).

Hierarchical models comprise a main *master* model, a set of auxiliary models and a set of rules defining how to rewrite letters of the master sequence using the auxiliary models. This can also be seen as a generalisation of the HMMs, as the hidden state sequence can be computed independently from the letters arising from each hidden state. A sequence length distribution can be provided with each rule through an expressive language, enabling complex constraints such as *the overall length of the sequence is a multiple of* 3 to be expressed.

3 REQUIREMENTS

A version of Java runtime environment 1.1.2 or higher must be installed on the computer.

Note: GenRGenS is free under the GNU license. GenRGenS uses the JavaCUP software. Therefore, some java_cup v0.10k files are joined to the GenRGenS distribution.

ACKNOWLEDGEMENTS

This work was partially supported by the French IMPG and "ACI IMPBio" programs, and the CNRS Specific Action "Modélisation et Algorithmique des Structures d'ARN".

REFERENCES

Butler,B.A. (1998) Sequence analysis using GCG. *Methods Biochem. Anal.* **39**:74-97.

Coward,E. (1999) Shufflet: Shuffling sequences while conserving the k-let counts, *Bioinformatics*, **15**, 1058-1059.

Denise,A., Roques,O., Termier,M. (2000) Random generation of words of context-free languages according to the frequencies of letters. In D. Gardy and A. Mokkadem, editors, *Mathematics and Computer Science: Algorithms, Trees, Combinatorics and probabilities*, Trends in Mathematics, 113-125. Birkhaüser.

Durbin,R., Eddy,S., Krogh,A., Mitchison,G. (1998) *Biological sequence analysis. Probabilistic models of proteins and nucleic acids*. Cambridge University Press.

Flajolet,P., Zimmerman,P. ,Van Cutsem, B. (1994) A Calculus for the Random Generation of Labelled Combinatorial Structures *Theoretical Computer Science*, **132**(1-2):1-35.

Gasteiger E., Gattiker A., Hoogland C., Ivanyi I., Appel R.D., Bairoch A (2003) ExPASy: the proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Res.* **31**:3784-3788.

Rambaut, A., Grassly, N. C. (1997) Seq-Gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.* **13**:235-238.

Reinert,G., Schbath,S., Waterman,M.S. (2000) Probabilistic and statistical properties of words: An overview. *Journal of Computational Biology*, **7**(1-2):1-46.

Searls,D.B. (1999) Formal language theory and biological macromolecules. *Series in Discrete Mathematics and Theoretical Computer Science*, **47**:117-140.