BIOINFORMATICS APPLICATIONS NOTE Vol. 23 no. 1 2007, pages 119–121
doi:10.1093/bioinformatics/btl519

Genome analysis

Repseek, a tool to retrieve approximate repeats from large DNA sequences

Guillaume Achaz[1,2,*], Frédéric Boyer[3], Eduardo P. C. Rocha[1,4], Alain Viari[3] and Eric Coissac[3,5]

[1]Atelier de Bioinformatique, Université Pierre et Marie Curie-Paris 6, 12, rue Cuvier, 75005 Paris, France, [2]UMR 7138 Systématique, Adaptation, Evolution, Université Pierre et Marie Curie-Paris 6, Bâtiment A, 7, quai St Bernard, 75252 Paris Cedex 05, France, [3]INRIA-Rhône Alpes projet HELIX, 655, avenue de l'Europe, Montbonnot, 38334 Saint Ismier Cedex, France, [4]Unité Génétique des Génomes Bactériens, Institut Pasteur, 28, rue du Dr Roux, 75724 Paris Cedex 15, France and [5]UMR 5163 LAPM, Université Joseph Fourier, BP 53, 38041 Grenoble Cedex 9, France

Received on July 20, 2006; revised on September 12, 2006; accepted on October 6, 2006
Advance Access publication October 11, 2006
Associate Editor: John Quackenbush

ABSTRACT

Summary: Chromosomes or other long DNA sequences contain many highly similar repeated sub-sequences. While there are efficient methods for detecting strict repeats or detecting already characterized repeats, there is no software available for detecting approximate repeats in large DNA sequences allowing for weighted substitutions and indels in a coherent statistical framework. Here, we present an implementation of a two-steps method (seed detection followed by their extension) that detects those approximate repeats. Our method is computationally efficient enough to handle large sequences and is flexible enough to account for influencing factors, such as sequence-composition biases both at the seed detection and alignment levels.

Availability: http://wwwabi.snv.jussieu.fr/public/RepSeek/

Contact: achaz@abi.snv.jussieu.fr, http://www.repetmasker.org

INTRODUCTION

The importance of genome redundancy has been strongly emphasized in the field of genome dynamics and evolution as well as in medical biology. A repeat is a sequence present twice or more with a high degree of similarity within a larger sequence (e.g. a chromosome) or set of sequences (e.g. a genome with several chromosomes). Each instance of the repeated sub-sequence is called a 'copy' of the repeat. Repseek aims at detecting as many as possible pairs of copies within or between large DNA sequences. Unlike RepeatMasker (Smit *et al*., 2004), we do not search for already well characterized repeated elements but instead we retrieve all repeated sequences without any a priori on the nature of the repeats. Furthermore, we do not construct families of repeats, which is the objective of multiple seeds extension (Price *et al*., 2005) or of clustering algorithms (Bao and Eddy, 2002; Pevzner *et al*., 2004), though our program can be used to feed the clustering algorithms. The detection of repeats is not a trivial problem and there is no satisfactory methodology available apart from recursive local alignment (using dynamic programming) of sequences with themselves (Waterman and Eggert, 1987). Such algorithms, however, are quadratic in computation time and memory and cannot be used for large sequences. Our approach, like most current methods to detect similarity in large sequences (Altschul *et al*., 1997; Vincens *et al*., 1998), works around the problem through a two-step strategy (Fig. 1). First, it detects seeds (strict repeats, i.e. repeats with neither indels nor substitutions) and, then it extends them into larger approximate repeats. The statistical evaluation of the repeats can be undertaken on seeds length or on repeats score (setting $L_{\min}$ and/or $S_{\min}$ parameters). Starting with longer seeds is faster but increases the chance to miss degenerate repeats. Both statistics can be used for the detection of repeats within a single sequence or between two sequences.

ALGORITHM

Several efficient algorithms are already available for computing the seeds (Abouelhoda *et al*., 2002; Kurtz and Schleiermacher, 1999) (see user's guide for a complete comparison). Repseek can accept as input a list of seeds; however, for simplicty, it also provides an exact builtin seeds detection algorithm, based on the KMR algorithm (Karp *et al*., 1972) that proves to be very efficient in practice. One of the main advantage of KMR in this context is that it can be implemented in a memory efficient way. Our current implementation requires $9n$ bytes direct repeats (where n is the sequence length) and $17n$ bytes for inverted repeats. All pairs of seeds are then extended on both sides, accepting substitutions or indels, by using a dynamic programming approach (Smith and Waterman, 1981). The edit matrix is filled as in the classical local alignment procedure, but the optimal path is anchored at the seeds extremities and ends up at a maximum of the matrix. To reduce the time and memory requirements, we use a heuristic similar to the one introduced in BLAST2 (Altschul *et al*., 1997). At the end, if more than one repeat share the same localization, only the one with the highest score is kept (users can tune how much overlap is required to do so). The use of a simple

*To whom correspondence should be addressed.

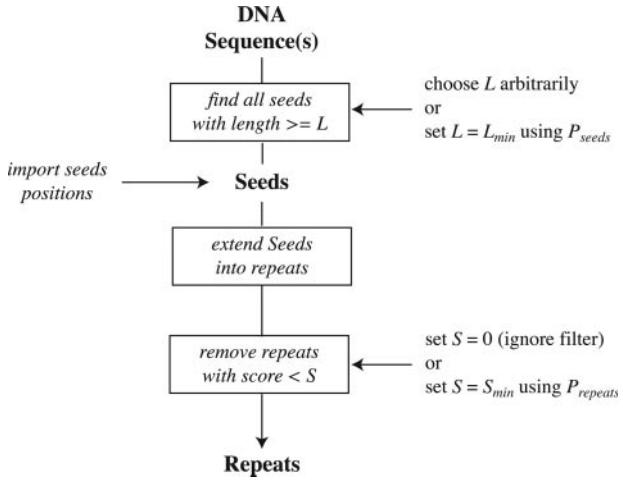

Fig. 1. Schematic workflow of repseek.

identity substitution matrix can create biases for sequences where the relative frequency of each nucleotide is not 1/4 (Achaz *et al*., 2003). In highly biased sequences, this results in longer alignments composed of the most abundant nucleotides. To fix this potential problem, repseek uses a matrix based on nucleotide frequencies that can correct for biases in sequence composition. The score for a match or a mismatch is scaled by the log of the product of the corresponding nucleotide frequencies. Other programs, such as reputer (Kurtz and Schleiermacher 1999) or RepeatScout (Price *et al*., 2005) also handle mismatches; though, to the best of our knowledge, no published program accepts indels or corrects for composition bias by using an adapted substitution matrix.

STATISTICS

Repseek proposes two statistics (P_{seeds} or P_{repeats}) to evaluate analytically the significance of a repeat. P_{seeds} is usually expressed as the probability $P(L_{\text{longest}-\text{seed}} \geq L)$ that the longest seed observed in a random sequence of same size and nucleotide composition is longer than L (Karlin and Ost 1985). Reciprocally, by imposing a statistical threshold, one can calculate the smallest length $L_{\min}$ above which no such seed is expected to occur by chance in a random sequence. An equivalent statistics is available for the analysis of seeds between two sequences. P_{repeats} is the probability $P(S_{\text{best}-\text{repeat}} \geq S)$ that the score of the best local alignment observed between two random sequences of size n and m is larger than a given score S. This probability can be well approximated by $P = e^{-\gamma mne^{tS}}$ (Karlin and Altschul, 1993). We evaluated the unknown parameters γ and t using the method proposed by (Waterman and Vingron 1994) for a range of sequence lengths (1 kb, 10 kb, 100 kb and 1 Mb) and compositions ($d_{GC} = |GC\% - 50|$ ranging from 0 to 35 by step of 5%). This was done by randomizing 10 000 random sequences for each combination of length and composition and using a least-square regression to estimate both parameters. Hence, we can associate a chosen probability with a minimum score $S_{\min}$ above which no repeats are expected to be found in a random sequence of same size and same composition.

PERFORMANCE

Repseek's memory and time consumption are typically small enough to handle large DNA sequences. On a G4 MacOSX, it takes 1 min with $L = 24$, $S = 0$ (i.e. $P_{\text{seed}} = 10^{-3}$) and around 3 min with $L = 16$, $S = 31.01$ (i.e. $P_{\text{repeats}} = 10^{-3}$) to retrieve all repeats from the genome of *Escherichia coli* (4.6 Mb). It takes 49 min or 5 h (depending on the chosen statistics) to detect all repeats on the chromosome V of *Caenorhabditis elegans* (20 Mb). Memory consumption is maximum at the seed detection step and is 80 Mb for the genome of *E.coli* and 359 Mb for the chromosome V of *C.elegans*. This shows that repseek can be potentially used to detect repeats on very large sequences on modern computers.

We performed a comparison of the repeated elements annotated by RepeatMasker in each *C.elegans* chromosome with the ones detected by repseek (using $P_{\text{repeats}} = 10^{-3}$, i.e. $L_{\min} = 17$ or 18 and $34.34 \leq S_{\min} \leq 34.94$ depending on the chromosome). Results shows that, on average, the sequence is composed at 12% of repeats detected by both Repeat-Masker and repseek, at 15% of repeats detected by repseek only and at 1% of repeats detected by Repeat-Masker only. This shows that, not only repseek retrieves almost all characterized repeats annotated by Repeat-Masker, but it also unravels a lot of yet uncharacterized repeated sequences. Interestingly, these later repeats are not only located in exons (i.e. gene duplicates), but span mostly intronic and intergenic regions.

Repseek is a fast and handy software that can detect approximate repeats in large chromosomes. The statistical pertinence of the detected repeats is evaluated considering the length and composition of the analyzed sequence. The C sources as well as a more-detailled user's guide can be found at the URL given above. Sources are publicly available and users are more than welcome to make improvements that will be incorporated in forthcoming releases.

ACKNOWLEDGEMENTS

The authors thank A. Platt and J. Pothier. G.A. was funded by La Fondation Singer-Polignac. The authors acknowledge the support of IMPBIO grant to EVOLREP. Funding to pay the Open Access publication charges for this article was provided by INRIA.

Conflict of Interest: none declared.

REFERENCES

Abouelhoda,M.I., Kurtz,S. and Ohlebusch,E. (2002) The enhanced suffix array and its applications to genome analysis. In *Proceedings of the Second Workshop on Algorithms in Bioinformatics*. Lecture Notes in Computer Science 2452, Springer-Verlag, pp. 449–463.

Achaz,G. *et al*. (2003) Associations between inverted repeats and the structural evolution of bacterial genomes. *Genetics*, **164**, 1279–1289.

Altschul,S.F. *et al*. (1997) Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res*., **25**, 3389–3402.

Bao,Z. and Eddy,S.R. (2002) Automated de novo identification of repeat sequence families in sequenced genomes. *Genome Res*., **12**, 1269–1276.

Karlin,S. and Altschul,S.F. (1993) Applications and statistics for multiple high-scoring segments in molecular sequences. *Proc. Natl Acad. Sci. USA*, **90**, 5873–5877.

Karlin,S. and Ost,F. (1985) Maximal segmental match length among random sequences from a finite alphabet. In Cam,L.M.L. and Olshen,R.A. (eds), *Berkeley Conference in Honor of Jerzy Neyman and Jack Kiefer*. Association for Computing Machinery, NY, Vol. **1**, pp. 225–243.

Karp,R.M., Miller,R.E. and Rosenberg,A.L. (1972) Rapid identification of repeated patterns in strings, trees and array. In *4th annual ACM symposium theory of computing,* ACM, pp. 125–136.

Kurtz,S. and Schleiermacher,C. (1999) Reputer: fast computation of maximal repeats in complete genomes. *Bioinformatics*, **15**, 426–427.

Pevzner,P.A. *et al*. (2004) De novo repeat classification and fragment assembly. *Genome Res.*, **14**, 1786–1796.

Price,A.L., Jones,N.C. and Pevzner,P.A. (2005) *De novo* identification of repeat families in large genomes. *Bioinformatics*, **21**, i351–i358.

Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.

Smit,A.F.A, Hubley,R. and Green,P. (1996–2004) , RepeatMasker Open-3.0.

Vincens,P. *et al*. (1998) A strategy for finding regions of similarity in complete genome sequences. *Bioinformatics*, **14**, 715–725.

Waterman,M.S. and Eggert,M. (1987) A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons. *J. Mol. Biol.*, **197**, 723–728.

Waterman,M.S. and Vingron,M. (1994) Rapid and accurate estimates of statistical significance for sequence data base searches. *Proc. Natl Acad. Sci. USA*, **91**, 4625–4628.