

A new efficient algorithm to find all DNA repeats with exact matching

Verónica Becher Alejandro Deymonnaz Pablo Heiber

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
March 2009

Motivation. There is significant ongoing research to identify the number and types of repetitive DNA sequences. As more genomes are sequenced, efficiency and scalability in computational tools become mandatory. Existing tools fail to find distant repeats because they can not accommodate entire chromosomes, but segments. Also, a quantitative framework for repetitive elements inside a genome or across genomes is still missing.

Results. We tackle the fragmental nature of biological repeats with the mathematical definition of a *pattern* (Gusfield (1997)). It requires exact matching of the different occurrences, allowing nested and overlapping patterns. Biological repeats could then be devised by combining smaller exact matching patterns.

We present a time and memory efficient algorithm and its implementation as a software tool to exhaustively find all patterns in sequences of up to 500 million nucleotide bases. Thus, the input can be made of possibly many entire chromosomes (a single chromosome, two, or more). The search is efficiently performed, with no upper bound on the length of the repeats, and the different occurrences can be arbitrarily distant. The output is ordered by length reporting, for each pattern, its number of occurrences and starting positions in the input data. Our tool also gives a quantification in terms of length, diversity and number of occurrences, in possibly many genomes.

The main contribution of our algorithm is its efficiency and exhaustiveness in extracting all patterns in large inputs, hence its usefulness to find novel repeats and to perform cross comparisons in different genomes. We verified the efficiency of our method with respect to the trade-off in time and memory, granted by its theoretical complexity. For inputs of size n , it has space complexity $17.25n$ and, by a convenient coding of the output, its time complexity is $\mathcal{O}(n \log n)$.

Our algorithm is based on the *suffix array* construction of Manber and Myers (1993) and a novel procedure to extract all perfect repeats in the entire input. It is well known that the linear space complexity of the alternative data structure, the *suffix trees*, hides a large constant that makes it impossible to allocate and manipulate multiple entire chromosomes even in 8 Gigabyte RAM, the currently largest addressable memory with nowadays workstations. Hence, the attractive linear time operations of suffix trees vanishes for such large inputs needed in comparative genomics. In contrast, the small constant involved in the linear space complexity of suffix arrays, and the order of $n \log n$ worst case time complexity for inputs of size n , have made suffix arrays the standard data structure replacing suffix trees (Gusfield (1997); Kärkkäinen *et al.* (2006); Puglisi *et al.* (2007); Poddar *et al.* (2007)).

Case study. We tested the software on the Homo Sapiens DNA genome NCBI 36.49. We computed all patterns of at least 40 bases occurring in any two chromosomes with exact matching. We found that each Homo Sapiens chromosome shares approximately 10% of its full sequence with every other human chromosome, distributed more or less evenly among the chromosome surfaces. We give statistics including a quantification of repeats by diversity, length, and number of occurrences. We compared the computed repeats against all biological repeats currently obtainable from Ensembl enlarged with dusts and all elements identified by TRF and RepeatMasker ftp://ftp.ebi.ac.uk/pub/databases/ensembl/jherrero/.repeats/all_repeats.txt.bz2. We report novel repeats as well as new occurrences of repeats matching with known biological elements.

Availability. The source code, results, and visualization of some statistics are accessible from <http://kapow.dc.uba.ar/patterns/>

Contact: vbecher@dc.uba.ar adeymo@dc.uba.ar pheiber@dc.uba.ar

Funding: Agencia Nacional de Promoción Científica y Tecnológica. Biosidus and IBM Argentina.

References

- Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press.
- Kärkkäinen, J., Sanders, P., and Burkhardt, S. (2006). Linear work suffix array construction. *Journal ACM*, **53**(6), 918–936.
- Manber, U. and Myers, G. (1993). Suffix arrays: a new method for on-line string searches. *SIAM J. Comput.*, **22**(5), 935–948. SODA '90: Proc. 1st annual ACM-SIAM symposium on Discrete algorithms, 319–327, San Francisco, 1990.
- Poddar, A., Chandra, N., Ganapathiraju, M., Sekar, K., Klein-Seetharaman, J., Reddy, R., and Balakrishnan, N. (2007). Evolutionary insights from suffix array-based genome sequence analysis. *J Biosci.*, **32**(5), 871–881.
- Puglisi, S. J., Smyth, W. F., and Turpin, A. H. (2007). A taxonomy of suffix array construction algorithms. *ACM Computing Surveys*, **39**(2), 4.