Faster Computation of Genome Mappability

Sahar Hooshmand
University of Central Florida
Orlando, FL
sahar@cs.ucf.edu

Paniz Abedin
University of Central Florida
Orlando, FL
paniz@cs.ucf.edu

Daniel Gibney
University of Central Florida
Orlando, FL
dangibney@ucf.edu

Srinivas Aluru
Georgia Institute of Technology
Atlanta, GA
aluru@cc.gatech.edu

Sharma V. Thankachan
University of Central Florida
Orlando, FL
sharma.thankachan@ucf.edu

ABSTRACT

The k-mappability problem is defined as follows: Given a sequence $S[1, n]$ of length n over a constant alphabet Σ, and two integers k and $m \leq n$, compute an array F_k, such that:

$$F_k[i] = |\{j \neq i \mid d_H(S[i, i+m-1], S[j, j+m-1]) \leq k\}|$$

The function $d_H(\cdot, \cdot)$ denotes the hamming distance. Derrien *et al.* [2] introduced this problem in the context of genome analysis. We propose a provably efficient algorithm for 1-mappability with $O(n \log n)$ worst case run time. The previous best known bound is $O(n \log^2 n)$ [1].

KEYWORDS

Genome mappability, suffix tree, heavy path decomposition, Hamming distance.

ACM Reference Format:
Sahar Hooshmand, Paniz Abedin, Daniel Gibney, Srinivas Aluru, and Sharma V. Thankachan. 2018. Faster Computation of Genome Mappability. In *ACM-BCB'18: 9th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, August 29-September 1, 2018, Washington, DC, USA.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3233547.3233645

1 INTRODUCTION

Genome mappability (see Table 1) is an important concept used in the analysis of high-throughput sequencing data, such as gene expression quantification, SNP calling and paired-end experiments. Very recently, Alzamel *et al.* [1] studied the 1-mappability problem and proposed three linear space algorithms with time complexities $O(n \log^2 n)$, $O(nm)$, and a $O(n)$ average-case time algorithm for $m = \Omega(\log n)$.

We propose a new algorithm for 1-mappability with time complexity of $O(n \log n)$ and space complexity of $O(n)$. It based on the approximate string matching framework in [3], which makes use of heavy path decomposition of the suffix tree, ST, of S.

Table 1: Mappability for $m = 3$ and $S = CCACAACA$

Position i	1	2	3	4	5	6
substring	CCA	CAC	ACA	CAA	AAC	ACA
$F_0[i]$	0	0	1	0	0	1
$F_1[i]$	3	2	2	2	1	2

2 OVERVIEW

The algorithm consists of two phases. In the first phase, we construct data structures based on ST. The first set of structures consists of compact tries for every light node in the heavy path decomposition with string depth less than m. The second set of structures consists of compact tries for every node in ST with string depth less than m. Both are created using modified suffixes involving one substitution. The essential property of these trees is that pairs of suffixes where a single modification could make the longest common prefix greater or equal to m correspond to leaves in a subtree rooted at string depth greater or equal to m. The total number of leaves in these new structures is $O(n \log n)$.

In the second phase, we traverse these structures one time. In the traversal, the subtree sizes along with the number of modified and unmodified suffixes in each trie are used to update the array F_1. Some final modifications to F_1 are done based on F_0, which is easily computed beforehand.

3 RESULTS

The creation of the data structures can be done in $O(n \log n)$ time using fast-merging techniques. Their traversal can also be done in $O(n \log n)$ time. By creating the new tries, performing the traversal, and then deleting them, space is maintained at $O(n)$. Combining these results, we have that there exists an $O(n \log n)$ time and $O(n)$ working space algorithm for the 1-mappability problem.

REFERENCES

[1] M. Alzamel, P. Charalampopoulos, C. S. Iliopoulos, S. P. Pissis, J. Radoszewski, and W.-K. Sung. Faster algorithms for 1-mappability of a sequence. In *International Conference on Combinatorial Optimization and Applications*, pages 109–121. Springer, 2017.

[2] T. Derrien, J. Estellé, S. M. Sola, D. G. Knowles, E. Raineri, R. Guigó, and P. Ribeca. Fast computation and applications of genome mappability. *PloS one*, 7(1):e30377, 2012.

[3] S. V. Thankachan, C. Aluru, S. P. Chockalingam, and S. Aluru. Algorithmic framework for approximate matching under bounded edits with applications to sequence analysis. In *Research in Computational Molecular Biology - 22nd Annual International Conference, RECOMB 2018, Paris, France, April 21-24, 2018, Proceedings*, pages 211–224, 2018.