Critique "Lightweight LCP Construction for Next-Generation Sequencing Datasets"

Markus J. Bauer , Anthony J. Cox ,
Giovanna Rosone , and Marinella Sciortino

The paper presents the first lightweight method that simultaneously computes, the longest common prefix array(LCP) and BWT of very large collections of sequences. Knowing the LCP of DNA sequences collection would facilitate the rapid computation of maximal exact matches, shortest unique substrings and shortest absent words. CPU-efficient algorithms for computing the LCP of a string have been described in the literature, but require the presence in RAM of large data structures.

The goal of presented method is to reduce memory consumption throught using mostly external memory. The BWT string and LCP array are built step by step reading the sequences from left to right. Data is reading sequencially from files held in external memory, so only small proportion of the symbols of string collection need to be held in RAM.

The presented method extends previous works on computing the BWT of a collection of strings and computing LCP via BWT. The paper focuses on the LCP computation algorithm and briefly describes BWT computation relying on previous paper.

The method, described in the paper, has strict and detailed explanation but it hardly to understand because of lack of examples and multiple links to previous paper. So, to completely understand and, perhaps, implement the presented algorithm it is nessessary to read the paper of the same authors about lighweight BWT construction for very large string collections.

As far as I can see the paper doesn't includes strict proofs of computational and memory complexity of algorithms. The main problem of the paper is that proof of main theorem is omitted due to lack of space but authors promise to defer it in the other paper. It looks very doubtfully because the presented algorithm is based in this therem at most.

The authors test method on the whole human genome sequencing data from HapMap project to analyze the additional overhead in runtime and memory consumption of simultaneously computing both BWT and LCP compared with the cost to computing the BWT alone. It shows increased CPU efficiency comparing to computing only BWT, 2-4 times more time consumption and about 2 times more memory consumption.

Also authors compared the lightweight LCP construction with common LCP construction algorithm. Common LCP construction required 18Gb of RAM to create the LCP in about 1 hour 45 minutes with precomputed BWT. The presented method needed 4.7Gb of RAM to create both BWT and LCP in just under 18 hours on 200 million of sequences 100 base long. On bigger dataset common algorithm exeeds 64Gb available RAM on 64Gb RAM machine.

As a conclusion, the presented algorithm is great for LCP construction on very large datasets. It require about 3 times less amount of RAM than the common approach at the expense of time consumption. It will be very useful for LCP construction on very large string collections when in-memory methods will fail due to lack of RAM.