A grammar compressor for string collections with applications to the construction of the BWT

Diego Díaz-Domínguez[1,2] and Gonzalo Navarro[1,2]

[1] Department of Computer Science, University of Chile, Chile
{diediaz,gnavarro}@dcc.uchile.cl
[2] CeBiB - Center for Biotechnology and Bioengineering, Chile

Abstract. The analysis of massive and repetitive strings collections in succinct space is an active field of research. In this regard, self-indexes that rely on *context-free grammars* (CFG) or Lempel-Ziv factorization achieve high compression ratios. Nevertheless, the string queries they can efficiently answer are still limited. On the other hand, classical self-indexes based on the *Burrows-Wheeler Transform* (BWT) answer different types of queries efficiently. However, they rely mostly on statistical entropy for compacting the data, which is not a good approach for capturing the long blocks of identical or highly similar patterns in repetitive text. Motivated by these limitations is that we propose a new grammar compression framework that facilitates the processing of repetitive data. We produce a CFG that only generates the input text, and that can be quickly transformed into the BWT of the collection. Thus, if the input is not used, then we maintain it in its CFG form to reduce space usage. However, if we need to retrieve information about the strings (maximal repeats, suffix-prefix overlaps, or something similar), then we quickly transform the CFG into a BWT-based self-index. Our framework consists of three algorithms, called LMSg, SuffPair and infBWT. The first one produces a preliminary CFG whose nonterminal symbols are used as building blocks for inferring the BWT; the second one reduces the size of the CFG while maintaining its properties, and the third algorithm computes the BWT from the final CFG. Preliminary experimental results show that the compression ratio achieved by the combination of LMSg and SuffPair is competitive with the classical LZMA and Deflate techniques. However, our approach compacts the data much faster. Also, further preliminary experiments show that the space usage of infBWT is not greater than the size of the uncompressed text. The primary application for our work is the processing of DNA sequencing reads, massive but highly-compressible string collections whose most relevant operation is the computation of suffix-prefix overlaps.