The FM-Index: A Compressed Full-Text Index Based on the BWT

Paolo Ferragina
Università di Pisa
ferragina@di.unipi.it

Giovanni Manzini
Università del Piemonte Orientale
manzini@unipmn.it

In this talk we address the issue of indexing compressed data both from the theoretical and the practical point of view.

We start by introducing the *FM-index* data structure [2] that supports substring searches and occupies a space which is a function of the entropy of the indexed data. The key feature of the FM-index is that it encapsulates the indexed data (*self-index*) and achieves the space reduction at no significant slowdown in the query performance. Precisely, given a text $T[1, n]$ to be indexed, the FM-index occupies at most $5nH_k(T) + o(n)$ bits of storage, where $H_k(T)$ is the k-th order entropy of T, and allows the search for the *occ* occurrences of a pattern $P[1, p]$ within T in $O(p + occ \log^{1+\epsilon} n)$ time, where $\epsilon > 0$ is an arbitrary constant fixed in advance.

The design of the FM-index is based upon the relationship between the Burrows-Wheeler compression algorithm [1] and the suffix array data structure [9]. It is therefore a sort of *compressed suffix array* that takes advantage of the compressibility of the indexed data in order to achieve space occupancy close to the Information Theoretic minimum. Indeed, the design of the FM-index does not depend on the parameter k and its space bound holds *simultaneously over all* $k \geq 0$.

These remarkable theoretical properties have been validated by experimental results [3, 4] and applications [7, 10]. In particular it has been shown that the FM-index achieves a space occupancy close to the best known compressors and, unlike them, it allows to search for arbitrary substrings in a hundred of megabytes within few millisecs, since it does not decompress the whole file.

We will conclude the talk by sketching two intriguing variants of the FM-index. One achieves $O(p + occ)$ query time (i.e. *output sensitivity*) and uses $O(nH_k(T) \log^{\epsilon} n) + o(n)$ bits of storage. This data structure exploits the interplay between two compressors: the Burrows-Wheeler algorithm and the LZ78 algorithm [11]. Our other proposal [8] combines two recent and elegant techniques—the compression boosting [5] and the wavelet tree [6]—to design a variant of the FM-index that scales well with the size of the input alphabet.

References

[1] M. Burrows and D. Wheeler. A block sorting lossless data compression algorithm. Technical Report 124, Digital Equipment Corporation, 1994.

[2] P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *Proc. of the 41st IEEE Symposium on Foundations of Computer Science*, pages 390–398, 2000.

[3] P. Ferragina and G. Manzini. An experimental study of a compressed index. *Information Sciences: special issue on "Dictionary Based Compression"*, 135:13–28, 2001.

[4] P. Ferragina and G. Manzini. An experimental study of an opportunistic index. In *Proc. 12th ACM-SIAM Symposium on Discrete Algorithms*, pages 269–278, 2001.

[5] P. Ferragina and G. Manzini. Compression boosting in optimal linear time using the Burrows-Wheeler transform. In *Proc. 15th ACM-SIAM Symposium on Discrete Algorithms (SODA '04)*, 2004.

[6] R. Grossi, A. Gupta, and J. Vitter. High-order entropy-compressed text indexes. In *Proc. 14th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA '03)*, pages 841–850, 2003.

[7] J. Healy, E. E. Thomas, J. T. Schwartz, and M. Wigler. Annotating large genomes with exact word matches. *Genome Research*, 13:2306–2315, 2003.

[8] P. Ferragina, G. Manzini, V. Mäkinen and G. Navarro. *An Alphabet-Friendly FM-index*. Subbmitted. 2004.

[9] U. Manber and G. Myers. Suffix arrays: a new method for on-line string searches. *SIAM Journal on Computing*, 22(5):935–948, 1993.

[10] K. Sadakane and T. Shibuya. Indexing huge genome sequences for solving various problems. *Genome Informatics*, 12:175–183, 2001.

[11] J. Ziv and A. Lempel. Compression of individual sequences via variable length coding. *IEEE Transaction on Information Theory*, 24:530–536, 1978.