**Contributions**

We propose a new data-layout of the FM-index data structures, denoted as split bit-vector encoding. The goal is to store in a compact way, occupying a minimum number of cache blocks, all data needed to perform the searching process.

**FM-index**

Most relevant data structure is the Occ table which requires several GiBs of memory. A sampled Occ table, called SFM, stores one out of d values. The BWT is used to reconstruct the missing Occ values. Last-to-First or LF() function behaves similar to the rank() query. Some previous solutions use wavelet trees or look-up tables.

**Algorithm BS: Backward Search**

**Input:** C[], Occ[], Q query, n=|T|, m=|Q|

**Output:** (sp, ep): Interval pointers of Q in T

1: sp = LF(Q[m],sp)
2: ep = LF(Q[m]+1,ep)
3: for i from m-1 to 0 step -1
4:     sp = LF(Q[i],sp)
5:     ep = LF(Q[i],ep)
6: end for
7: return (sp+1, ep)

**Computational Analysis**

**Memory Access Pattern:** Due to the nature of the input queries and the Occ table, the search loop causes an access pattern not predictable and distributed along the whole table.

**Search Throughput:** To measure the computational performance we consider the search throughput (ST), defined as the number of LFMs (Last to First Mapping operation) performed per second.

**Search Intensity:** We define search intensity (SI) as the number of LFMs performed per transferred byte from memory, relating the number of operations to the required amount of data traffic.

**Main References**


**Bit Vector Sampled FM-index (bvSFM)**

Our new data layout, Split bit-vector, compacts all data needed to search k symbols in a single step (k-step), reducing both memory movement and computing requirements at the cost of increasing memory footprint.

**Methodology and Results**

Experiments conducted on an Intel Xeon Phi 7210 processor (Knights Landing, or KNL), with 64 cores, 16 GiB of MCDRAM and 192 GiB of DDR4 DRAM, running Ubuntu 16.04.1 Linux.

**Conclusions**

Our optimized data structure packs all relevant data needed in a query step within a single cache block, minimizing the memory bandwidth demand. Our proposal clearly outperforms previous versions. The performance obtained in the Intel Xeon Phi (KNL) architecture obtain a throughput about 3x faster than previous GPU implementations.

**Acknowledgments**

This work was supported in part by grants TIN2016-80920-R (AEI/ERDF, UE), TIN2017-86355-C2-1-R (AEI/ERDF, UE), University of Malaga and gaZ: T58, 17R research group (Aragón Government and European Social Fund).