**Short read alignment with populations of genomes (Summary)**

The increasing amount of genomic data poses problems to: short DNA sequence (read) alignment to reference genomes and variation discovery of newly sequenced genomes with respect to those previously sequenced. Already constructed short-read alignment programs use BWT as it returns linear time and a small memory requirement. The BWT of the reference genome is pre-run so later on, the reads can be mapped to it using BWT backwards search. This leads to biases and lower accuracy according to the reference chosen.

Read mapping of a newly sequenced human genome to a collection of genomes has not yet been accomplished. One attempt is GenomeMapper (Schneeberger et al.) which uses hash-based data structures and k-mer indexing whereby identical regions (due to the redundancy of sequenced genomes) are stored only once. Another (by Siren et al.) uses prefix-sorted finite automaton and BWT-based indexing. Both have a large memory requirement. To handle genetic variants and avoid any bias, BWBBLE (a BWT-based read alignment algorithm) was constructed to provide little memory consumption and relatively efficient computation time.

A reference genome can be augmented with genomic variant data from a collection of genomes; the augmented reference being the reference multi-genome. SNPs are handled by extended the reference's alphabet from 4 to 16 letters for the IUPAC nucleotide code. Thus, when aligning a read to the reference, a nucleotide can match more than one character leading to multiple separate SA intervals. These need to be minimised by the four-bit Gray code (reflected binary code), theoretically the optimal order for such a problem. Other than SNPs, differing types of genomic variations are common; i.e. insertions, deletions, inversions and translocations. These generate multiple branches (bubbles) with the SA. While these augmentations aid in handling alignment, a higher memory overhead is produced. It is however possible to reduce memory consumption by filtering out branches – lower accuracy.

The first method implemented for the BWBBLE program is that of exact matching of a read to the reference multi-genome, an extension of the BWT-based backwards search algorithm. This was then further extended to allow mismatches and gaps; i.e. inexact matching, an extension of the inexact search algorithm employed by BWA with reduced search space and improved performance. As for the memory required by the program, to reduce it the BWT string is compressed and only a subset of the SA arrays is stored. Further reduction would unfortunately lead to increased time consumption.

The BWBBLE aligner's performance was evaluated against BWA (a greatly known BWT-based single-genome aligner) resulting in BWBBLE performing better on reads which cover a larger number of indels with a high SNP count. It does have a slower running time due to the larger amount of SA intervals but running on a multi-genome as opposed to a single-genome will lead to less running time so it compensates for itself. Furthermore, most of the same reads are aligned by both. It was also evaluated against GCSA (executes short read alignment with multiple sequences) resulting in BWBBLE taking a lot less time to construct its index.

Thus, a disadvantage of this aligner is its use of read length being dependent on padding to gather variants implying construction of multiple indices for each length. An advantage is its efficiency in aligning a multi-genome when compared to aligning each genome separately.