System Configuration:

CPU: 2.2Ghz Intel Core i7
RAM: 16GB DDR4

Statistics:

Input:
Reference number of characters : 5362495
Number of reads: 500000
Output: MappingResults_Peach_simulated_reads.txt

Timing statistics:
ConstructST: 3.803692 seconds
PrepareST: 0.691481 seconds
MapReads: 1270 seconds
Output: Used printf

Other statistics:
Total number of nodes suffix tree = 8945576
Avg. number of alignments performed per read: 1229318/500000 = 2.46

HitRate : (500000-112)/500000 = 0.9998
Number of misses 112.

Justification:
The time taken to construct suffix tree is linear O(n) and takes around 3.8 seconds. The number of nodes consumed for ST is 8945576 and is less than twice the number of characters.

The major time consumed in the algorithm is MapReads consuming around 20 minutes. This number is proportional to the number of local alignments performed per read.
Currently using ST the per read number of alignment is around 2.46.

This way we reduce the alignment search from O(reference_sequence length x read_length) = O(5362495 x 100) to O(2.46 x 200 x 100) i.e.O(avg num align x 2*read_length x read_length)

Also, the number of misses is around 112. I.e for 112 cases we did not find a good local match giving a hit rate of 99.98%. This is also dependent on the minimum string depth (l) value. Since currently this value is 25, we restrict per read number of alignment to 2.46. Reducing this would increase the per read number of alignment and thus more number of options to search and thus reduce the misses. By using minimum string depth (l) of around 10, the per read number of

alignment increases to 10.46 and the total search time increases to around 1 hour and reduces the number of misses to 95.

Currently the param.config is hardcoded. Would try to make it config after Thursday