GrashMap (Phase 2)

Changes we did:

We had computed the *contif2transcript* mapping in the Phase1, where we have a map of contig number to appropriate transcripts. These contig number are the index numbers of the contig in *contigIdmap*. This mapping is created when we do parsing of our reference file.

Now, when we parse the query file, what Mashmap was doing is it was passing the allReadMappings data structure which contains the information computed while we parsed the reference file to further map it to the query file. Now we (in GrashMap) have changed this allReadMapping data structure (computeMap.hpp):

1. map.refstartPos = map.queryLen + Contig2Transcript[x][j].length (line177)
2. map.refSeqId = stoi(Contig2Transcript[x][j].transcriptId) (line 185)
3. Map.strand = 1 if Contig2Transcript[x][j].orientation is True (line 191).

Now we are passing this structure in to the filterMappings function.

Issues in Phase 2:

We changed the allReadMappings data structure in computeMap.hpp to accommodate for filtering steps in filterMapping(). However, this is not invoked in mapping mode, it is invoked only in OETOONE mode. Therefore, it is not behaving in the way we expected it to behave.

Now, the main work for mapping is done by the function mapModuleHandleOutput(). This function is called inside a while loop which has a condition of threadpool.outputAvailable() when a thread currently being executed has a output available or threadpool.running() when it needs to check for remaining outputs.
As we have changed the window size to 31 which is being used for both query and reference processing and there is a condition which checks window size before going to this mapModuleHandleOutput. So, the line in computemap.hpp which is, if(len < param.windowSize || len < param.kmerSize || len <

param.segLength), is creating problem, which is a fixable one. The way we think to fix is to have kmerSize and windowSize precomputed to a optimum value according to the kind of input file we are providing. That is we can have a function to calculate window size and kmer size according to our input file.

Query File Creation:

1. To make a query file, take substrings from contigs in .gfa file.
2. Repeat each of this substring minimum 10 times to crate one query.
3. Maintain query file in from of fasta file.

We have pushed a sample query file onto phase2 repo by the name of query.fa .