Whole genome graph creation from multiple assemblies

CSHL Graph Genome group

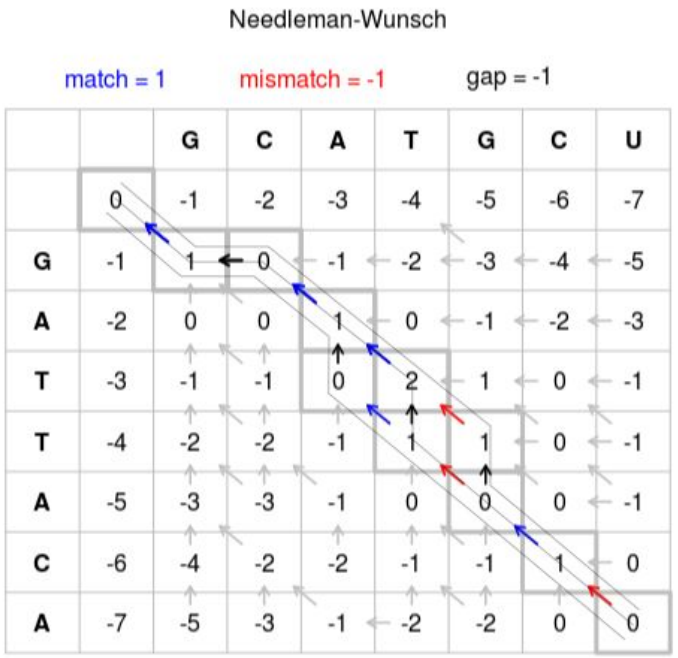
Step 1: Align all Contigs to GrCH38

1. Input:
    a. FASTA contig files (CHM1, CHM13, HG003, HG004, HX1, Korean, NA19240)
    b. Reference FASTA (GRChr38)
2. Output:
    a. BAM
3. Tools:
    a. BWA
4. Notes: **COMPLETE**

Overheard: *This is easy*

Step 1.1: Globally Align Contigs

1. Input:
   a. FASTA files (contigs)
   b. Reference FASTA (GRChr38)
   c. BAM
2. Output: BAM
   a. each contig has single alignment
3. Tools: **Local to Global Alignment**
   a. Biederstedt extension of Needleman-Wunsch algorithm



		G	C	A	T	G	C	U
	0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5
A	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
T	-4	-2	-2	-1	1	1	0	-1
A	-5	-3	-3	-1	0	0	0	-1
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

https://upload.wikimedia.org/wikipedia/commons/3/3f/Needleman-Wunsch_pairwise_sequence_alignment.png

Overheard: *The edges are what, again?*

Step 2: Divide & Conquer Multiple Sequence Align

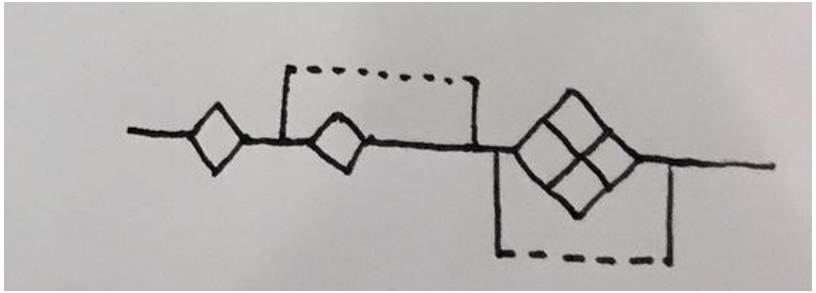
1. Input:
    a. BAM
2. Output:
    a. BAM
3. Tools:
    a. MAFFT (alignment)
    b. Some C++/python/PERL code (chopping & reassembling)
        i. Identify window coordinates* (start with windows of size ~10kb) in alignment file
        ii. Extract sequence data for each window
        iii. Convert window to FASTA format
        iv. Run MAFFT for each window [could be parallelized]
        v. Reassemble* the individual multiple sequence alignments into a single alignment file

Overheard:*Why are you doing this to me?*

* These are potential bottlenecks, both in the current coding process, as well as implementation (they require serial processing)

Step 3: Export Graph Genome

1. Input:
   a. BAM
2. Output:
   a. VCF (intermediate)
   b. GFA
3. Tools:
   a. Vg
   b. Other wrappers



Overheard:*When I do it by hand it works...*