Part 4. Sequence overlapper for the Minimus assembler (AMOS package)

Due: 11:59pm, Tuesday December 11th, 2012 (28 days from today)

No late submissions accepted

Overall weight: 40% of total project score

Specification: You must write a program that reproduces the functionality of the hash-overlap program from the AMOS package. Specifically, your program must extract a collection of sequence reads from an FASTA file (or AMOS bank), perform an all-vs-all comparison between the sequences using a local alignment algorithm with affine gap penalties and substitution matrix, then report the overlaps found in AMOS format and store them in an AMOS bank.

In addition, you must integrate this program within the Minimus assembler, essentially replacing the call for hash-overlap with a call for your program within the script running the Minimus pipeline.

Your program must allow the user to specify the following parameters:

- minimum overlap length
- minimum % identity within the overlapping region
- alignment parameters (match, mismatch, and gap open, gap extension penalties)
- maximum region that can be ignored at the beginning/end of the aligned reads (i.e. how far from the end of a read is the alignment allowed to start)

All these parameters must have default values that will be used in case the user does not directly override the information. These values must be documented in a README file.

Documentation for the AMOS package can be found at http://amos.sourceforge.net.

You can use any Bio* utilities you find useful, including alignment routines.

You can (and should) use additional heuristics to ensure your program is efficient, e.g. use exact alignment tricks to figure out which sequences might overlap before performing the expensive inexact alignment operation.

Data sizes:

Expect that your program will have to assemble a data-set of ~ 100,000 sequences of length up to 1,000 bp.

Contest:

The top performing program will receive extra credit: The student who writes the fastest program will receive a 20 point bonus. The next top 5 running times will receive a 10 point bonus. Note: correctness of the program will also be tested - your program must produce a reasonable assembly of the data-set in order to be considered for the speed competition.

Output format: for each pair of overlapping sequences you will have to report their relative orientation (assume first sequence is always "forward") as well as the amount each sequence "hangs" (extends) outside of the overlap region.

AMOS format for overlaps:

```
{OVL
adj:N
rds:0,279
scr:0
ahg:-32
bhg:-32
}
```

Details:

rds – The overlapping reads, referenced by their IIDs.

adj - Read adjacency. [N]ormal, [A]nti-normal, [I]nnie, [O]utie which are EB, BE, EE, BB overlaps respectively. Where I(nnie) indicates the second sequence is in the opposite orientation than the first one. N(ormal) would indicate both sequences are in the same orientation.

ahg - Ahang. Length of the non-overlapping portion of the first read.

bhg - Bhang. Length of the non-overlapping portion of the second read.

Note: Positive Ahang values indicate seqA starts before seqB, negative indicate seqB starts before seqA.

scr – An unsigned integer overlap score.

flg – Universal_t flags plus one additional flag (A/B/C), default to zero if unspecified.

http://sourceforge.net/apps/mediawiki/amos/index.php?title=Message_Types#Overlap_t_:_Universal_t

Additional details

The broad outline of an overlapper program:

- **(i) build a hash of all k-mers in the set of sequences to identify which sequences could overlap;**
- **(ii) run the local aligner from project 1 on each pair of sequences that could overlap in order to identify good alignments between them;**
- **(iii) if S-W indicates a proper "dove-tail" alignment, report the layout of the two sequences with respect to each other.**

Commands to convert overlap output to AMOS bank:

1) `toAmos_new -s reads.fasta -b assembly.bnk`
2) `bank-transact -b assembly.bnk -m overlaps.afg`

Overview of Minimus assembly pipeline:

```
## Parse and validate input FASTA FILE
00: Ideally modified code from project part 1

## Running overlap on FASTA file
10: YOUR_OVERLAPPER <PARAMETERS> -s <INPUT> -o <OVERLAP OUTPUT>

## Building AMOS bank
20: toAmos_new -s <INPUT> -b assembly.bnk

## Add Overlap messages to AMOS bank
30: bank-transact -b assembly.bnk -m overlaps.afg

## Running contigger
40: tigger -b assembly.bnk

## Running consensus
50: make-consensus -B -b assembly.bnk

## Outputting contigs
60: bank2contig assembly.bnk > <CONTIG FILE>

## Converting to FastA file
70: bank2fasta -b assembly.bnk > <FINAL FASTA OUTPUT>
```

Sample data available at: http://www.cbcb.umd.edu/research/benchmark.shtml

Documentation for the AMOS package can be found at http://amos.sourceforge.net.

Start early!