Genetic Sequence Alignment (2014 SP)

The goal of this exercise is to identify which fragments of a DNA sequence appear within a longer reference sequence. This is an important problem from the field of computational biology.

A DNA sequence is a string made from the letters G, C, A and T. The input to your application will be a single long sequence (called the "reference") and a collection of short sequences (called "reads"). Your goal is to identify subsequences of the reads that match some part of the reference.

For example, if you are given the reference sequence

```
ref:  GATCTCTATGCAAAATACGTATTTGTACGTCCACCCTCGGAGTGGTG
```

and one of the reads is

```
read: CGTATTTGTACATCCACCCTCGG
```

your goal is to discover that they match well when lined up as follows:

```
match:                 ___________ ___________
ref:  GATCTCTATGCAAAATACGTATTTGTACGTCCACCCTCGGAGTGGTG
read:                  CGTATTTGTACATCCACCCTCGG
```

This problem can be solved using two MapReduce jobs as follows.

- In the first map phase, the input reference and reads will be broken into 10 character sequences (called "10-mers," or more generally "k-mers"). For example, the read "AGCTAGCTCAGTACC" would be mapped as follows:

```
read X: AGCTAGCTCAGTACC
output: AGCTAGCTCA       occurs in read X at offset 0
         GCTAGCTCAG      occurs in read X at offset 1
          CTAGCTCAGT     occurs in read X at offset 2
           TAGCTCAGTA    occurs in read X at offset 3
            AGCTCAGTAC   occurs in read X at offset 4
             GCTCAGTACC  occurs in read X at offset 5
```

  The mapper will output the k-mers as keys, and identifying information (such as the source sequence and offset) as values.

- The first reduce phase collects all of the occurrences of the given k-mer, and outputs a list of all possible matches between a sequence and a read.

- The second map phase will take the k-mer matches as input and will output them with keys given by the identities of the reference and read to which they correspond.
- The final reduce phase will combine adjacent shared k-mers. For example, if the 10-mers at offsets 5, 6, and 7 of a read r match the reference at offsets 17, 18, and 19, then we know that the subsequence of r of length 12 starting at offset 5 matches the reference at offset 17.

At the end of these two phases, the output will be a list of partial matches between reads and references.

Sample data

We have provided some sample data for you to work with. Each line in the example files contains an identifier, the string ”READ” or ”REF”, and a sequence of characters.