Sequence/Genome Assembly

Author : Deepak Khatri

```
Step 1: Take input from the contig file in format of
                1 CGGTAGGACAGGCCGGGCCGCAGCTGATGAAAGCGGTGTATGAGATGGCC
                2 CGGAAGTATGTGGGTGCTGCCGGAAGTCCGGCGCAGATGCGGCGGGCCGA
                3 GCCGATTAATTTTAATCAGAACAATCACGTGGTGATTCAGAACGACGGTA
                4 CGAAATTTTGCGACCGGAGGATTTACGGGAACCGGCGGCAAATATGAGCC
                .
                .
                .
        i.e. 'index' 'sequence' (if some other format is used to store the file change this step appropriately)

Step 2: find the overlaping sequence map.

Step 3: Find the index of first contig.

Step 4: Find order of contigs relative to first.

Step 5: Assemble sequence/genome in order.
```

In [1]:

```python
# opening the file with raw data
fileName = "data.txt"
with open(fileName) as f:
        data = dict(l.split() for l in f)
data
```

Out[1]:

```
{'1': 'CGGTAGGACAGGCCGGGCCGCAGCTGATGAAAGCGGTGTATGAGATGGCC',
 '2': 'CGGAAGTATGTGGGTGCTGCCGGAAGTCCGGCGCAGATGCGGCGGGCCGA',
 '3': 'GCCGATTAATTTTAATCAGAACAATCACGTGGTGATTCAGAACGACGGTA',
 '4': 'CGAAATTTTGCGACCGGAGGATTTACGGGAACCGGCGGCAAATATGAGCC',
 '5': 'AACCAATTGGTGTCGGGAACCTGTACCGCCTGATGCGGGGCTATGCGGAA',
 '6': 'GGTGGGGATTGTCGGGAGTATCGGCAGCGCTATTGGCGGGGCTGTTGGTG',
 '7': 'TGGTGGCATCCGCGTCAGGCGGTACAGCCATTCAGGCAGCTGCGGCGAAA',
 '8': 'CTCCGTGCTGTCCATGATGACAGAAATTCTGCTGAAGCAGGCAATGGTGG',
 '9': 'TGGCCAGGTGCGCAGGATGAGCTCCGGCTGCAGTTGCGTGATGGCGGTCT',
 '10': 'GAGCCGGATTGTCCACCGCGGGGAGTTTGTCTTCACGAAGGAGGCAACCA'}
```

In [2]:

```python
# overlaping sequence map
d = dict()
for name1, seq1 in data.items():
    for name2, seq2 in data.items():
        if name1 != name2:
            if name1 not in d:
                d[name1] = dict()
            for i in range(len(seq1)):
                if seq1[i:] == seq2[:len(seq1)-i]:
                    d[name1][name2] = len(seq1[i:])
d
```

Out[2]:

```
{'1': {'2': 1, '3': 3, '4': 1, '8': 1, '9': 5},
 '2': {'3': 5, '4': 3, '5': 1, '10': 2},
 '3': {'1': 5, '5': 1},
 '4': {'1': 1, '2': 1, '3': 3, '8': 1, '10': 5},
 '5': {'2': 5},
 '6': {'3': 1, '7': 2, '9': 2, '10': 1},
 '7': {'4': 5, '5': 1},
 '8': {'3': 1, '6': 1, '7': 3, '9': 3, '10': 1},
 '9': {'7': 1, '8': 2},
 '10': {'5': 1}}
```

In [3]:

```
# finding the first contig of the sequence
for i in d.keys():
    flag = False
    for j in d[i].keys():
        if int(j) > 3:
            flag = True
    if not flag:
        first = i
        break
first
```

Out[3]:

```
'5'
```

In [4]:

```
# finding the order of contigs
def findOrder(first, d):
    if max(d[first].values()) < 3:
        return [first]
    else:
        m = max(d[first].values())
        for k in d[first]:
            if d[first][k] == m:
                nextRead = k
        return [first] + findOrder(nextRead, d)
order = findOrder(first, d)
```

In [5]:

```
# Sequence/Genome Assembly
sequence = ''
for readName in order[:-1]:
    rightOverlap = max(x for x in d[readName].values() if x >= 3)
    sequence += data[readName][:-rightOverlap]
sequence += data[order[-1]]

sequence
```

Out[5]:

```
'AACCAATTGGTGTCGGGAACCTGTACCGCCTGATGCGGGGCTATGCGGAAGTATGTGGGTGCTGCCGGAAGTCCGGCGCAGATGCGGCGGGCCGATTAA
TTTTAATCAGAACAATCACGTGGTGATTCAGAACGACGGTAGGACAGGCCGGGCCGCAGCTGATGAAAGCGGTGTATGAGATGGCCAGGTGCGCAGGATG
AGCTCCGGCTGCAGTTGCGTGATGGCGGTCT'
```