**Algorithm Assignment**

Local Alignment

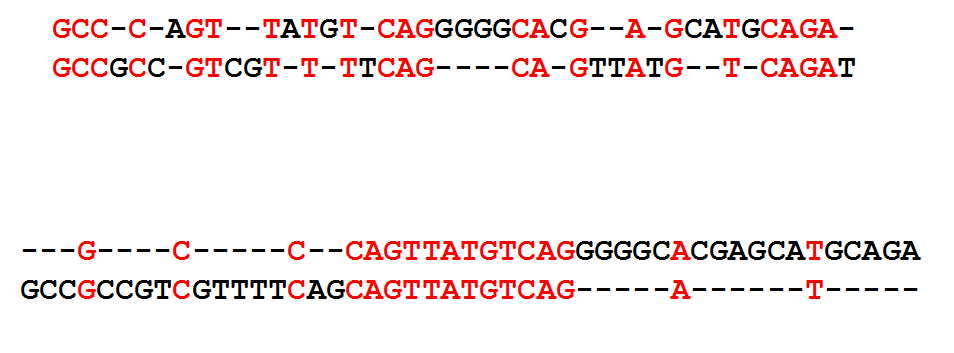
Due : 12/9

**목표**

수업시간에 배운 LCS(Longest Common sequence)를 바탕으로 Local Alignment를 구현하여 보자.

**Local Alignment란?**

생물학에서 두 유전자 서열을 나타내는 string을 align하는 문제를 LCS 문제로 이해할 수 있다. 특히, LCS는 Global alignment를 구하는 문제에 해당하는데, 여기서는 두 서열 전체를 가장 많이 일치하도록 align하는 optimal solution을 찾는다. 하지만 실제로 유전자 서열 사이의 유사성을 알아보기 위해 alignment를 할 때는 전체적으로 더 많은 수의 matched character를 찾는 것보다는 국지적으로라도 서로 잘 맞는 부분을 찾는 것이 유용할 때가 더 많다. 예를 들면 아래 그림에서 위에 나타낸 global alignment는 전체적으로 더 많은 match를 포함하지만 아래에 나타낸 local alignment처럼 특정한 영역에서의 match가 훨씬 좋게 되는 alignment를 선호한다.



이를 위해서 alignment 에서 match, mismatch, gap의 각 경우에 대해 score 및 penalty를 정의한다. 아래와 같은 scoring matrix가 입력으로 주어졌을 때 local alignment를 위한 recurrence를 정의하고 이를 구현해 보시오.

**Score(및 Penalty)**

Local alignment를 위한 score(및 penalty)는 아래와 같이 3가지로 정의 할 수 있다.

1. **Match**  
   두 character가 같을 때 주어지는 score
2. **Mismatch**  
   두 character가 다를 때 주어지는 penalty
3. **Gap**두 Character를 Match나 Mismatch로 대응시키는 대신, Character를 공백으로 건너 뛰는 것을 Gap이라 한다.

**Score Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **A** | **C** | **T** | **G** | **⧿** |
| **A** | 5 | -4 | -4 | -4 | -10 |
| **C** |  | 5 | -4 | -4 | -10 |
| **T** |  |  | 5 | -4 | -10 |
| **G** |  |  |  | 5 | -10 |

예시는 아래와 같다.

**S1 : ACAGT**

**S2 : AGTC**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Match | Miss Match | Gap | | Match | Gap |
| A | C | A | G | T | - |
| A | G | - | - | T | C |

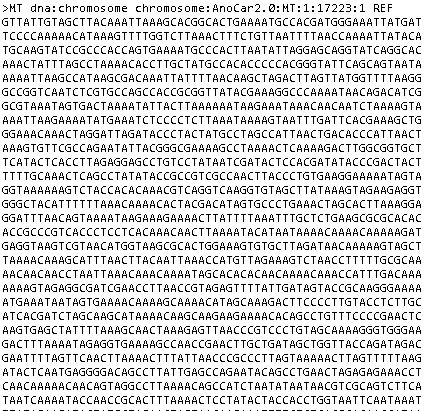
Score : 2 Match + 1 Mismatch + 3 Gap

= 2\*(5) + 1\*(-4) + 3\*(-10) = -24

??? AGT?? 연속포함된거? 알아서 북치고장구치고 다하라는 겨 말라는 겨?

**입력**

1. **DB와 Query file의 절대 경로와 output file의 경로를 argument로 받는다.**
2. **DB와 Query의 Data format DB와 Query의 format은 첫줄에 ‘>name’와 같은 형식으로 이름이 나오고, 그 다음줄부터 파일 끝까지 name에 해당하는 sequnce가 나열된다. 예시는 아래와 같다.**



**출력**

**Output file 이름은 입력으로 받은 DBName\_QueryName.txt로 하고, Local alignment 결과를 아래와 같은 형식으로 출력한다. (출력 예시는 Test 폴더 아래의 DB와 query를 사용해서 얻은 Test\_answer를 참고)**

**DB : file name**

**Query : file name**

**Alignment:  
<Test\_answer의 출력 예시 참고>**

**제출**

1. **보고서**
   1. **TestDB를 DB로 사용하고 TestQuery# 을 query로 사용하였을 때, query의 길이 변화에 따른 수행시간, memory 사용량을 비교하여 분석하여 본다.**
   2. **3가지 생물의 서열이 \_\_\_\_MT.fa 파일로 각각 주어졌다. 이 세 종류의 서열을 서로 locally align 한 결과를 제출한다. 즉, 세 개의 서열 중 하나를 DB로, 다른 하나를 query로 사용해서 local alignment를 수행한 결과를 모두 구한다. 또한 수행시간과 memory 사용량을 비교하여 분석해 본다.**
   3. **GitLab 에 제출**
2. **소스 코드 (Java로 구현)**

**- 컴파일 되지 않을 시 감점**

**- coding convention 참고하고 최대한 지킬 것**

**- inline document 없으면 0점**