SeqAn을 이용한 한글 정렬 - DeVAC과의 비교를 중심으로 -

An Alignment of Hangul using SeqAn
- A Comparison with DeVAC -

부산대학교 컴퓨터공학과
Park Sun Young
E-mail : parksy@pusan.ac.kr

Revised at 2010.11.30

ABSTRACT

유사 문서 탐색 시스템인 DeVAC은 생물정보학에서 활용되는 유전자 비교 분석 방법을 기반으로 문서의 내용을 탐색하여 유사 여부를 판별하는 프로그램이다. 생물정보학에서는 유전자의 비교 분석을 위하여 수많은 알고리즘과 강력한 도구들이 개발되어 사용되고 있는데, SeqAn은 그 중 하나이다. 본 보고서에서는 SeqAn이 지원하는 정렬 기능의 사용법에 대해 알아보고, 이를 이용하여 한글 문서를 비교하여 그 성능을 DeVAC과 비교해 보았다. SeqAn은 전역 정렬(global alignmen), 지역 정렬(local alignment), Motif 탐색 등이 가능하며, 그래프 알고리즘도 다수 지원한다. 한글 문서를 SeqAn의 지역정렬 방법으로 탐색해 본 결과, DeVAC과 거의 동등한 수준으로 유사한 영역을 잘 찾아내었다. 하지만 탐색 시간이 DeVAC에 비해 70배 이상 많이 걸리는 등 전처리 없이 실제로 활용하기에는 무리가 있었다. 또한 한글 문서를 이용해 Motif 탐색을 수행하고 싶었으나 char 데이터형을 원활하게 지원하지 않아 사용하지 못했다. 추후 이 기능을 제대로 사용할 수 있는 방법을 찾아서 적용할 수 있는 응용 분야를 연구할 계획이다.

KEYWORDS text plagiarism, document evolution, similar document, SeqAn, Motif

1 서 론

내용 기반의 유사 문서 탐색 시스템인 DeVAC은 문서를 상호 비교하기 위하여 각 문서를 하나의 유전자로 보고 생물정보학에서 사용되는 유전자 비교 분석 방법을 이용한다. 생물정보학에서는 유전자를 비교 분석하기 위해 많은 알고리즘과 강력한 도구들이 제안되어 활발하게 사용되고 있는데, 이 중 BLAST [1], BLAT [2], PatternHunter [3] 등이 널리 사용되는 도구이다. 본 보고서에서는 SeqAn [4]이라는 라이브러리에 대해 소개하고, 이 라이브러리의 설치 활용법에 대해 설명한다. 또한 이 라이브러리를 사용하여 한글 문서를 비교한 후 DeVAC[5]과 기능 및 성능에서의 장단점을 비교 분석할 것이다.

2 SeqAn 소개 및 설치

SeqAn은 독일 Freie Universität에서 개발한 시퀀스 분석 라이브러리이다. SeqAn은 정렬 알고리즘과 그래프 알고리즘, 인덱싱 등의 기능을 제공하며, 이를 위하여 각종 데이터 형과 입출력 인터페이스를

제공한다. SeqAn은 비교적 최근(2009년 9월)의 최신 버전까지 활발하게 업데이트가 되고 있으며, 이를 이용하여 생물학에서 유전자 서열을 분석하는 논문이 지속적으로 발표되고 있으며 관련 라이브러리가 책으로 출판되어 있다[6] [7]. SeqAn을 사용하기 위해서는 SeqAn 공식 홈페이지[4]에 접속하여 Downloads 메뉴의 Releases 에서 다운받을 수 있다. 2010년 10월 현재, 최신 버전은 2009년 공개된 Seqan_Release_1.2.zip (4.2MB)이다. 이를 다운받은 후 압축을 풀면 즉시 사용할 수 있다. 사용 가능 환경은 다음과 같다.

1. 작성 언어 : ISO C++
2. 실행 가능 환경

   Windows 계열 : Visual C++ 7,8,9, MinGW

   Linux 계열 : Linux, Mac OS X, Solaris (G++ 3.x, G++ 4.x)

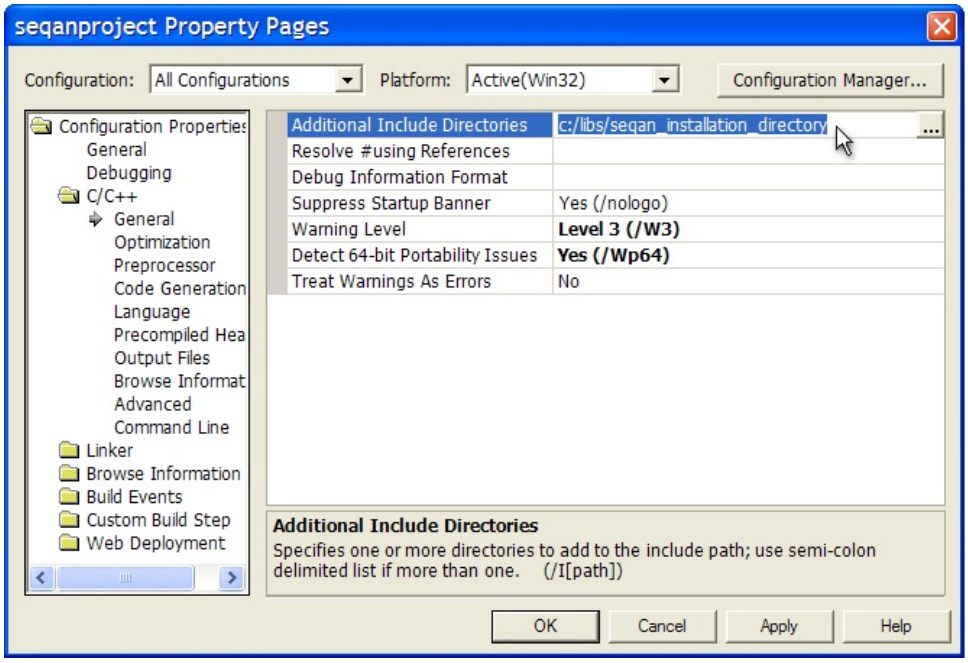

FIGURE 1. MS Visual C++ 계열에서 SeqAn을 사용하기 위한 프로젝트 특성 설정 창. Additional Include Directories 항목에 SeqAn이 설치되어 있는 폴더를 지정하면 된다. 위 그림에서 예로 든 설치 폴더는 c:/libs/seqan_installation_directory 이다.

Visual C++에서 이 라이브러리를 사용하기 위해서는 VC++의 설정을 변경하여야 한다. Project - Properties - All Configurations - C/C++ 메뉴에서 Additional Include Directories에 포함하여야 한다. 그림 1을 참조하면 된다. 두 번째로 그림 2와 같이 Tools - Options - Projects - VC++ Directories 에서 Show directories for를 "Include files"로 고친 후 가운데 창에서 New Line 버튼을 누르고 SeqAn의 설치 폴더를 입력하면 된다. 다른 방법으로는 apps 폴더와 demos 폴더에 VC버전 별 프로젝트 파일까지 포함되어 있으므로, 프로젝트를 그대로 불러와서 사용하여도 된다. Linux, Darwin,

Solaris, MinGW 환경의 사용자들은 컴파일 시 -I 옵션을 포함하면 된다. 이 역시 demos 폴더에 예시 Makefile 파일을 생성해 두었기 때문에 이를 참조하면 도움이 된다. 단, 일부 gcc 버전에서는 -pedantic 옵션도 같이 포함하여야만 정상적으로 작동하며, 32-bit OS 에서 4GB 이상의 큰 파일에 접근할 필요가 있을 때에는 -D_LARGEFILE_SOURCE 와 -D_FILE_OFFSET_BITS=64 옵션을 붙여주어야 한다.

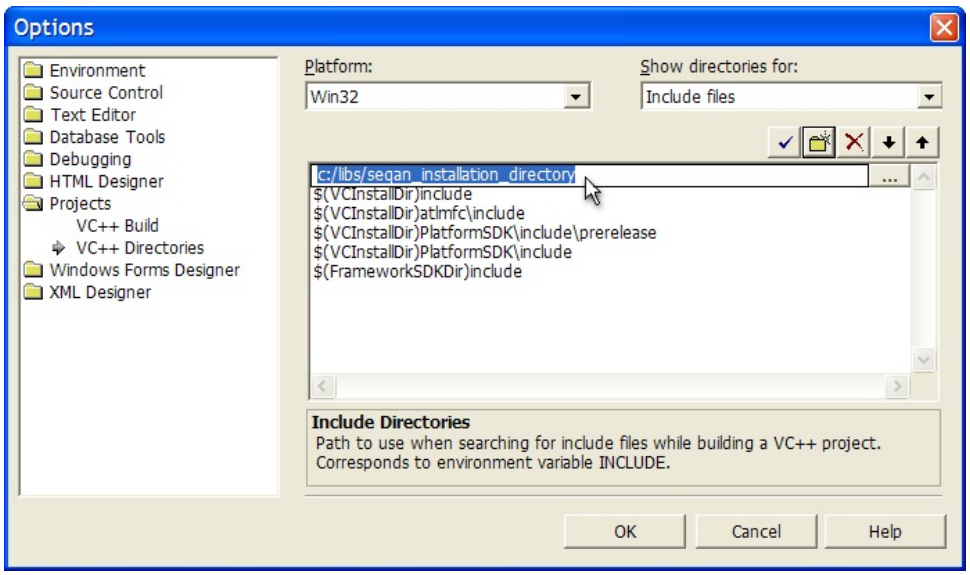

FIGURE 2. 그림 MS Visual C++ 계열에서 SeqAn을 사용하기 위한 프로젝트 옵션 설정 창. 항목에 SeqAn이 설치되어 있는 폴더를 지정하면 된다. c:/libs/seqan_installation_directory 이다.

3 SeqAn을 이용한 한글 정렬

3.1 데이터 타입

SeqAn에서는 String 형을 사용해 데이터를 처리하는데, 이 String은 C++ 표준에서 지원하는 string 이 아니라 '<>' 연산자를 사용하여 내부 데이터 형을 지정해야 하는 컨테이너의 일종이다. 이 String 내부에 들어갈 수 있는 데이터형으로는 character, nucleotide, amino acids, DNA, DNA5, lupac 등을 지원하는데, 주로 생물정보학에서 사용하는 내용들이다. 한글을 사용하기 위해서는 character 형을 사용하면 된다. 나머지 데이터형을 선언한 후 한글을 입력하면 무효가 되어 'N'이 삽입되므로 데이터를 활용할 수 없게 되므로 주의해야 한다. 선언하기 위해서는 String<char> str; 과 같이 선언하면 된다. 또한 SeqAn의 String 형은 덧셈, 삽입, 검색, 비교, 문자열 반전 등 많은 기능이 구현되어 있어 쉽게 사용할 수 있고, StringSet 이라는 컨테이너가 있어 이를 이용해 여러 개의 String을 한꺼번에 처리하는 것이 가능하다. 또한 Score 컨테이너가 존재하는데, 여기에는 int 형만 올 수 있다.

3.2 한글 정렬

SeqAn을 이용하여 한글을 전역정렬하기 위해서는 일반 영어 문장을 정렬할때와 동일한 방법으로 SeqAn을 사용하면 된다. 즉, String<char> 변수 2개에 원하는 한글 문장을 입력하고, Score 변수를 설정한 다음, Align 컨테이너에 값을 초기화하고 globalAlignment 함수를 호출하면 된다. 지역정렬도 동일하나, localAlignment 함수를 호출한다는 점이 다르다. 전역정렬 및 지역정렬은 한 번씩 수행하는 소스 코드는 다음과 같다.

```
// Alignment 예제
// Dna형 시퀀스 2개 선언
typedef String<char> TSequence;
TSequence seq1 = "한글 정렬을 해 보겠습니다.";
TSequence seq2 = "우리 한글 정렬을 해 볼까요?";
// Score 변수 설정. match=0, mismatch=-1, gapexten=-1, gapopen=-2
Score<int> score(3, -3, -2, -2);

// Align을 선언하고 seq1과 seq2를 입력
Align<TSequence, ArrayGaps> align;
resize(rows(align), 2);
assignSource(row(align, 0), seq1);
assignSource(row(align, 1), seq2);

// MyersHirschberg 알고리즘으로 두 시퀀스를 전역정렬 후 점수 및 정렬된 문자열 출력
::std::cout << "Score = " << globalAlignment(align, score, MyersHirschberg())
  << ::std::endl;
::std::cout << align << ::std::endl;

// SmithWaterman 알고리즘으로 두 시퀀스를 지역정렬
::std::cout << "Score = "
   << localAlignment(align, Score<int>(3,-3,-2, -2), SmithWaterman())
   << ::std::endl;
::std::cout << align << ::std::endl;
```

TABLE 1. SeqAn을 활용한 전역 정렬 및 지역 정렬 예제.

위 소스코드의 결과는 그림 3과 같다. 영문과 마찬가지로 한글에 대한 정렬 연산이 정상적으로 수행되었으며, 정렬 점수(alignment score)도 잘 계산이 된 것을 확인할 수 있다. 단, 한글 1글자는 2바이트이기 때문에 점수도 2배로 연산이 된 것을 알 수 있다. 실제로 활용하기 위해서는 이를 고려하여 Score 변수를 책정해야 할 것으로 보인다.

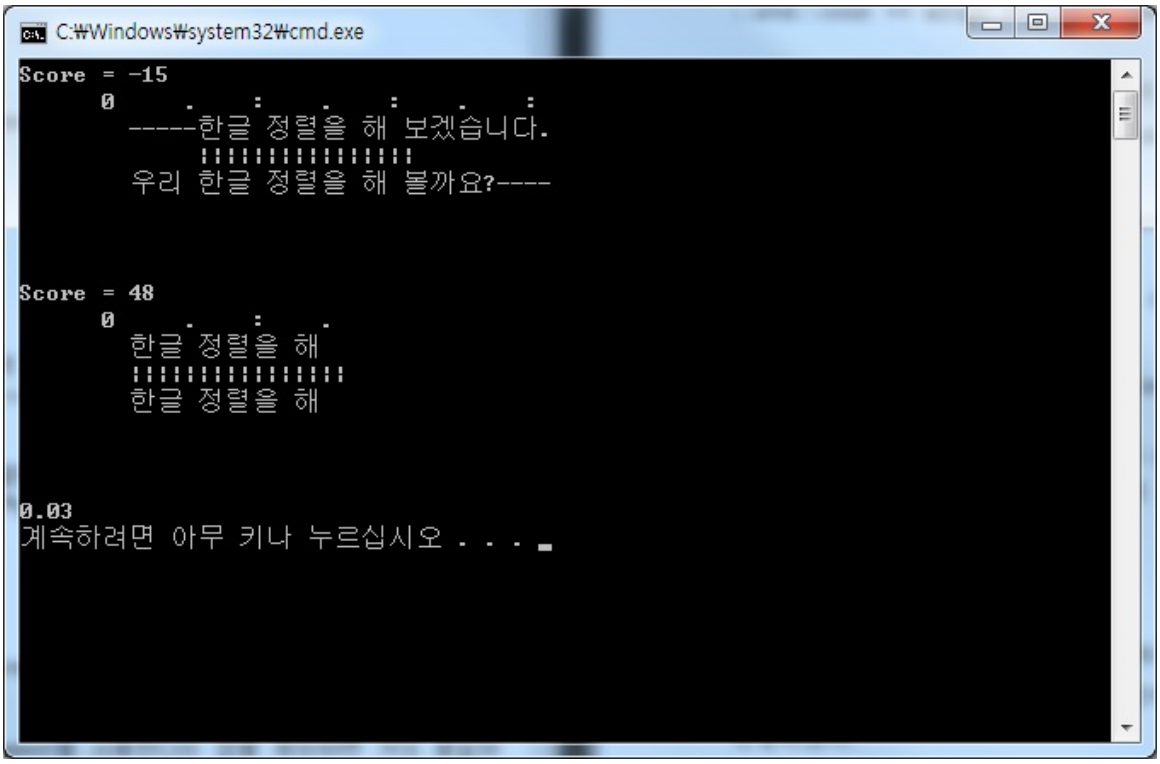

FIGURE 3. SeqAn을 이용한 한글 문서에 대한 전역 정렬 및 지역 정렬 결과

4 SeqAn과 DeVAC의 성능 비교 실험

4.1 실험 데이터 및 실험 방법

SeqAn을 이용해 한글 문서를 처리하는 경우의 성능을 측정하여 DeVAC의 한글 처리 능력과 비교해 보았다. 실험 데이터는 20~200KB까지의 영문 및 한글 문서를 입력하여 전역 정렬에 걸리는 시간을 측정하였다. 전역 정렬에 적용한 알고리즘은 MyersHirschberg 알고리즘을 적용하였다. 또한 DeVAC에서 200KB 문서 2개를 입력하여 그 시간을 측정하였다.

4.2 실험 결과

SeqAn의 전역 정렬 측정 결과는 표 2와 같다.

실험 결과, SeqAn에서는 한글 처리를 위해 특별히 더 많은 컴퓨팅 파워를 요구하지는 않는다. 다만 길이에 비례하여 연산속도가 증가하기 때문에 긴 문서를 처리하기에는 적합하지 않다. DeVAC에서 200KB 문서 2개를 검사할 때에는 0.5초 이내에 검사가 완료되었다. 충분한 전처리 과정을 거치기 때문인데, SeqAn에는 4 종류의 알려진 정렬 알고리즘 중 하나를 선택할 수 있는 옵션만 있을 뿐 전처리와 관련한 옵션은 존재하지 않았다.

No.	문장 1의 길이(KB)	문장 2의 길이(KB)	영어문서(초)	한글문서(초)
1	20	20	0.73	0.81
2	20	40	1.61	1.52
3	40	40	3.11	3.20
4	100	100	19.37	18.97
5	200	100	36.12	37.32
6	200	200	74.11	75.54

TABLE 2. SeqAn을 이용한 영어 문서와 한글 문서에 대한 전역 정렬에 걸리는 시간. 문장 1,2에 해당하는 영어문서와 한글 문서 각각에 대해 실험하였다. 전역 사전에 걸리는 시간은 영어문서와 한글문서 사이에 차이가 없었으며, 문장1과 문장2의 길이에 비례했다.

5 결론 및 추후 연구

본 보고서에서는 SeqAn이라는 라이브러리를 이용하여 한글 문서를 정렬해 보았다. SeqAn은 시퀀스 분석을 위한 강력한 툴로써 정렬알고리즘 및 그래프 알고리즘 등을 지원한다. 한글 문서에 대한 정렬 실험 결과, 정렬 자체는 잘 되나 그 속도가 DeVAC에 비해 70배 이상 느려 실제로 적용하기 위해서는 반드시 문서 내부에서 전처리를 수행해야 한다는 것을 알 수 있었다. 한글 Motif 검색을 위해 소스 코드를 수정했으나 Motif Finder 클래스에서 DNA형(A,C,G,T)외에 입력받을 수 있는 자료형을 찾는데 실패했다. 또한 SeqAn은 여러 개의 시퀀스에서 가장 유사한 하나의 영역을 찾아내는 motif 탐색 기능을 지원한다. 이 기능을 한글 문서에 적용해보고자 하였으나 MotifFinder 클래스를 분석하는 과정에서 이 클래스가 DnaString형만 지원하는 등 DNA에 최적화되어 있어 데이터를 정상적으로 입력할 수 없었다. 만약 이 기능을 정상적으로 사용하게 되어 그 응용 분야를 찾는다면 좋은 연구가 될 것으로 판단된다. 이 부분에 대해서는 motif 탐색에서의 데이터 형 지원에 대해 더 조사해보고 추후 보고를 할 것이다. 추후 Motif Finder 클래스를 분석하여 SeqAn에서 한글 motif이 탐색한지 실험할 계획이다.

References

1. Altschul S.F., W. Gish, W. Miller, and E. Myers, "Basic local alignment search tool," *Jour. of Mol.Biol.*, , no. 215, 1990.
2. W.J. Kent, "Blat: The blast-like alignment tool," *Genome Res*, , no. 12, 2002.
3. B. Ma, J. Tromp, , and M. Li, "Patternhunter-faster and more sensitive homology search," *Bioinformatics*, , no. 18, 2002.
4. Freie Universtät, "Seqan," http://www.seqan.de/.
5. 류창건, 김형준, 박선영, 조환규, "Devac(document evolution analysis center," http://devac.cs.pusan.ac.kr/.
6. T. Rausch, A.K. Emde, D. Weese, A. Döring, C. Notredame, and K. Reinert, "Segment-based multiple sequence alignment," *Bioinformatics*, , no. 24(16), 2008.

7. A. Gogol-Dring and K. Reinert, *Biological Sequence Analysis Using the SeqAn C++ Library*, 2009.