제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

다중 지역 정렬을 위한 알고리즘

장석봉* 이계성**
* ** 단국대학교 전자컴퓨터학부
e-mail:hollo98@korea.com

An Algorithm for multiple local alignment

Suk-Bong Jang* Gye-Sung Lee**
* ** Dept of Computer Science and Electronics, Dan-Kook University

요 약

본 연구는 생물정보학(Bioinformatics)의 가장 기초적인 분야중 하나인, 새롭게 밝혀진 유전자 서열과 이미 밝혀진 유전자 서열 사이의 유사성(similarity)이나 상동성(homology)을 찾기 위한 방법에 대한 연구 중 지역 서열정렬로 사용하는 알고리즘인 Smith-Waterman 알고리즘이 갖고 있는 문제를 파악한다. 긴 서열에 대한 선호를 막고 대신 부분적인 지역 정렬을 다수 개 찾아 정렬시키는 알고리즘을 제안하기로 한다.

1. 서론

생물 정보학에 있어서 DNA 염기서열 분석이나 단백질 서열 분석은 매우 중요한 부분 중에 하나이다 [1,2]. 서열 정렬이란 핵산이나 단백질의 서열을 적절히 배열시켜 서열간의 상관관계를 보여 줄 수 있고, 이를 통해 핵산이나 단백질의 상동성(homology)을 판단할 수 있는 방법으로, 유전체 연구에 많이 활용되고 있는 기술이다. 상동성의 종류에 따라 크게 두 가지로 나눠지게 되는데 전역정렬은 서로 동일한 종류의 핵산이나 단백질 서열을 비교하여 최대의 상동성이 나타나도록 정렬하는 경우에 사용되며, 대표적인 알고리즘으로 Needleman-Wunsch 알고리즘이 있다. 이 경우는 두 서열의 시작에서 끝까지 사이에서 가장 좋은 점수를 갖도록 정렬시켜 최상의 정렬을 찾는 것이다. 반면에 좀 더 보편적인 경우로, 전체 대신에 두 서열간의 일치하는 부분을 찾아 정렬시키는 방법을 생각할 수 있는데, 이 경우는 여러 가지 실제 상황에서 자주 일어나게 된다. 두 단백질 서열이 공통된 도메인을 공유하거나 핵산 서열의 확장된 부분을 비교할 경우가 그에 해당되는 예이다. 두 개의 다변화된 서열간을 비교하거나, 특히, 서로 다른 종으로 구분되지만 진화적으로 공통된 원점을 공유하는 경우에 서로간의 상동성을 조사할 때 유용하게 사용될 수 있는 방법이 된다. 이런 경우, 많은 부분은 서로 정렬될 수 없을 정도로 변화되어 축적되지만 일 부분은 감지할 만큼의 상동성을 그대로 유지 보존되어 현재에까지 이르는 경우가 있는데, 이런 경우 적절히 응용될 수 있는 방법이 바로 지역 정렬이다. 또한 유사한 기능의 서열이나 전역 상동성이 낮아서 쉽게 상동성을 나타내지 못하지만, 일부분에서는 높은 유사성을 보이는, 즉, 일부분에서 높은 지역 상동성이 있는 경우도 지역 정렬을 실행하는 것이 효과적이다. 지역 정렬의 대표적인 알고리즘으로 실제 응용에서 많이 사용되고 있는 것이 Smith-Waterman 알고리즘이 있다.

2. Smith-Waterman 알고리즘

Smith-Waterman 지역 서열 정렬 알고리즘은 computational 분자생물학 분야에서 가장 중요하게 사용되는 기법 중의 하나임은 분명하다. 이 알고리즘은 잘 보존된 부분을 찾고 그렇지 않은 부분을 제거하도록 설계되어 졌다. 두 개의 서열, X와 Y가 아래와 같이 정의된다:

$X=x_1x_2...x_n$와 $Y=y_1y_2...y_m$ ($n \geq m$)

지역 정렬은 X와 Y의 부분 서열, 즉, I와 J의 쌍을 포함하는 정렬을 의미한다. 두 서열은 각기 가로와 세로 축을 형성하며 각각의 행과 열에 해당되는 셀에 각기 지역 정렬 점수, $S_{i,j}$가 할당되는데, 이 값을 최대화하는 고전적 다이나믹 프로그래밍 공식이 아래와 같이 정의된다:

$$S(i,j)=\max\{\ 0,\ S(i-1,j)-d,\ S(i-1,j-1)+s(x_i,y_j),\ S(i,j-1)-d\ \}$$

여기서, $S_{i,j}=0$ ($i=0$ 또는 $j=0$일때)

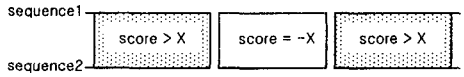
4가지 중 0은 새로운 정렬을 시도하는 것에 해당된다. 이 점에서 정렬 점수는 증가할 수 있고 이점에 이르는 동안 정렬점수의 합은 음수로 되어 임의로 정렬하였을 경우 정렬점수는 도리어 낮아지는 결과가 되므로 차라리 새로운 정렬을 찾는 것이 지금까지 맞춰온 것을 확장하는 것보다 낫다는 의미가 된다. 따라서 정렬점수를 구하는 행렬의 최 상단과 좌측 열에는 모두 0으로 설정하게된다.

지역 정렬은 행렬 상에서 가장 큰 값을 갖는 점으로부터 역으로 추적하여 정렬을 찾아내는 방식이다. 따라서 최대값을 갖는 행렬의 셀을 계속 유지하고 있어야 한다.

지역 정렬 알고리즘이 제대로 동작하려면 무작위 일치에 대한 기대점수가 음수가 되어야 한다. 만일 그렇지 않다면, 서로 연관이 없는 긴 서열의 정렬에 있어 큰 점수를 갖게 되기 때문이다. 실제로 중요한 부분서열 정렬이 있음에도 불구하고 서열이 길다는 점만으로 이를 포함하여 불필요한 정렬이 일어나는 점이 Smith-Waterman 알고리즘의 단점으로 등장한다. 이 문제는 다음절에서 좀더 자세히 알아보기로 한다.

3. Smith-Waterman 알고리즘의 제한점

이 알고리즘에는 중요한 몇 가지 제약이 내재하고 있다. 그 중 하나는 유전적으로 잘 보존(well-conserved)되지 못한 부분이 전체 정렬에 포함되어 잘 보존된 부분과 섞여 있을 때, 이것이 정렬을 왜곡시키는 결과로 나타나는 경우가 발생한다. 그 이유로 Smith-Waterman 지역 정렬 알고리즘은 최고의 점수를 갖는 지역 정렬은 찾지만 최고의 유사성, 최고의 일치도를 갖는 지역 정렬은 찾지 못하는 문제가 알고리즘 자체에 포함되기 때문이다. 결과적으로 유사성의 척도를 가지고 두 서열간의 정렬을 통해 지역 정렬을 찾는 것이 이 알고리즘의 한계라고 알려져 있다[4]. 실제로 Smith-Waterman 알고리즘의 주요 특성은 서열 정렬에서 비유사도 (non-similarity)가 높은 시작 부분과 끝 부분 조각을 제거하는 데는 효과적이나 비 유사성을 갖는 조각이 잘 보존된 정렬 사이에 위치해 있을 때는, 포함된 이들 세 조각을 하나의 정렬로 묶어 주는 결과로 나타난다. 본 연구에서는 이 문제를 본 논문에서는 비일치 병합 정렬 (mismatched join alignment)이라 정의하기로 한다. <그림 1>은 이 문제점을 설명하고 있다. score가 -X인 지역이 X보다 큰 두 지역에 쌓여 있을 때 X 값의 크기에 상관없이 3개의 조각은 하나의 정렬로 합쳐지게 된다 [4]. 중간에 위치한 조각은 생물학적으로 아무 의미를 갖지 못하는 조각으로 정렬의 의미를 감소시키는 역효과를 가져오게 된다.



<그림 1.> 두 서열간 정렬 예

이 문제를 해결하려는 시도가 여러 연구를 통해 이뤄졌다. 그 중 최소 비일치 (mismatch) 밀도를 이용한 정렬을 통해 이 문제를 해결하려는 시도가 있었으나 [3] 성공적인 알고리즘으로 귀결되지는 않았다. 최근에 [4]에서 제안된 해결책으로, 위에서 설명한 조각 문제를 갖지 않는 부분 정렬 문제로 긴 정렬문제를 분해시키는 방안이 제시되었다. Zhang의 해결책은 후처리(post-processing) 방식으로 처리되도록 설계되어 있어, 지역 정렬을 완전히 구성한 후에 부분 서열 정렬이 이뤄지도록 제안되어 있다. 이 방법의 단점으로 만일 지역 정렬에 포함되지 않으면서 의미 있는 부분 정렬이 있을 경우에는 그것을 찾을 수 없다는 점이 있다. 또 다른 처리 방식으로 고려할 수 있는 방법으로, 매우 긴 갭을 포함시키는 방향으로 정렬알고리즘을 수정할 수 있다. 본 연구에서는 지역정렬 과정의 일 부분으로 부분 정렬을 구하는 알고리즘을 제안하며, 휴리스틱을 이용해서 효율적인 실행이 이뤄지도록 알고리즘을 설계하기로 한다.

4. 다중 지역 정렬 알고리즘

본 연구에서는 Smith-Waterman 알고리즘을 기반으로 하되 한 개의 최대 값을 찾아 지역 정렬을 찾는 한계를 확장하여 지역 정렬 내의 다수 개의 최대 값을 추적하여 지역 정렬 내에 존재하는 부분 지역 정렬을 찾아내는 알고리즘을 제안한다.

4.1 지역 정렬내의 부분 지역 정렬

다수 개의 최대 값은 지역 정렬의 값의 변화에 따라 결정된다. 지역정렬 내의 값을 추적할 때, 다수개의 봉우리를 형성하게 되면 봉우리의 첨도를 중심으로 봉우리가 유효한지를 결정한다. 우선 이를 위해 Smith-Waterman 알고리즘으로 지역정렬을 위해 S(i, j) 값을 구해 나가면서 각각의 (i, j) 셀에 값의 변화를 추적하는 순방향 변화율, fc(i,j)를 하나 추가하고 이 변수의 값은 아래와 같이 정의한다:

$$\begin{aligned} fc(i,j) = 0, &\quad \text{if } fc(k,l)<0, \& \\ &\quad S(i,j)<S(k,l) \\ fc(k,l)+1, &\quad \text{if } S(i,j)\geq S(k,l) \\ fc(k,l)-1, &\quad \text{if } S(i,j)<S(k,l) \end{aligned}$$

여기서,
$(k,l) \in \{(i-1,j-1),(i,j-1),(i-1,j)\}$ &
$S(i,j)$가 최대가 되는(k,l)

이 변화율과 함께, 값의 변화를 나타내는 (i,j) 셀은 지역적으로 최대가 되는 점과 지역적으로 최소가 되는 두 개의 점으로 나눠져 지역적 봉우리와 지역적 계곡으로 나타낸다. 각 봉우리와 계곡은 정렬점수를 계산하는 과정에서 파악되고, 각각은 연결(linked)리스트로 연결되어 있어, 이웃하고 있는 봉우리와 봉우리 사이, 계곡과 계곡 사이간을 파악할 수 있고, 이들을 분석하여 유효한 봉우리와 계곡 셀을 식별한다.

변화율 자체는 정렬 점수의 증감을 나타내고 양의 값의 클수록 값의 증가폭이 크다는 점을 암시한다. 이 변화율 값이 음의 값을 갖게 되면 0으로 설정하여 음의 값에서 양의 값으로 전환할 때는 바로 증가로 나타날 수 있게 한다. 0의 값으로 대체하는 것은 지역 정렬에서 음의 값을 없애 비 정렬부분을 정렬 대상에서 제외시키는 효과와 같은 결과를 내게 만드는 역할을 하게된다.

지역 정렬에서 최대가 되는 지점에서의 변화율 값은 양의 값을 갖게되는데, 지역정렬을 찾기 위해 역방향으로 추적하는 과정에서 변화율 값이 0값에 이르게 되면 여기까지가 하나의 부분 지역 정렬을 형성하게된다. 계속해서 역방향으로 부분 지역 정렬을 찾아나가지만 만일 여러 개의 소규모 봉우리가 존재한다면, 계곡에 해당되는, 구간 내 최저 점에 이르면서 역방향으로 정렬점수가 증가하지만 큰 봉우리를 형성하지 못하게 된다. 소규모 봉우리의 제거는 봉우리간 최소 점을 중심으로 비정렬 병합에서 사용하는 비정렬 병합 파라메터인 X값의 변화를 보고 지역적으로 최대 값에 해당하는지를 결정한다. 만일 최소 값을 중심으로 양옆의 두 최대 값이 비일치 병합 정렬 조건을 만족하게 되면 그 최대 값을 갖는 셀부터 역방향으로 지역정렬 구간을 구하게 된다. 만일 그 조건을 만족하지 못한다면, 그 다음에 있는 최소점을 중심으로 같은 계산을 반복하면서 지역 최대값을 구해나간다. 단, 최대값은 지금까지 조사된 구간의 최대값을 가지고 비일치 병합 정렬 조건을 조사하게 되어야 한다. 이렇게 계산함으로 전체적으로 값의 변화가 미세한 소규모 봉우리를 부분지역정렬에 포함시킬 수 있게된다.

이 변화율은 크기의 증가폭에 대한 구체적인 영향을 고려하지 않은 점이 단점으로 나타났다. 단백질 정렬에 있어서는 일치될 때의 정렬점수 값이 그렇지 않을 경우와 다르다. 따라서 계속 정렬 점수가 증가하더라도 큰 폭의 변화가 없이 서서히 증가하는 경우와, 반대로 감소하더라도 작은 폭의 감소가 연속되어 산등성이를 형성한다면 부분 정렬로 취하기가 어렵다. 따라서 변화 폭에 따른 변화율로 수정하면 다음과 같이 변화될 수 있다:

$$\begin{aligned} fc(i,j) = 0, &\quad \text{if } fc(k,l)<0, \& \\ &\quad S(i,j)<S(k,l) \\ fc(k,l)+s(i,j)/m, &\quad \text{if } S(i,j)\geq S(k,l) \\ fc(k,l)-s(i,j)/m, &\quad \text{if } S(i,j)<S(k,l) \end{aligned}$$

여기서,
$(k,l) \in \{(i-1,j-1),(i,j-1),(i-1,j)\}$ &
$S(i,j)$가 최대가 되는(k,l)

그리고 m은 단백질 정렬의 경우 아미노산간 일치될 때의 점수의 평균으로 BLOSUM50의 경우 7.5로 m값은 최소 0.67에서 2까지의 값을 갖는다.

$$m = \sum_{i=1}^{20} match(p_i, p_i)$$

4.2 실험 결과

다수개의 지역정렬을 산출해내는 알고리즘을 이용하여 2개의 단백질간 서열 정렬을 시험해 보기로 한

다. 핵산 서열 정렬과 달리 단백질 정렬에서는 BLOSUM50 대체 행렬 (substitution matrix) 또는 PAM 행렬을 이용하여 정렬의 점수를 계산한다. 본 연구에서는 BLOSUM50 대체 행렬을 사용하여 실험하였다. 두 서열은 다음과 같이 정의되었다.

서열 1: HEAGGAWGGPQRCKAGEE

서열 2: PGAWIMCKAGAE

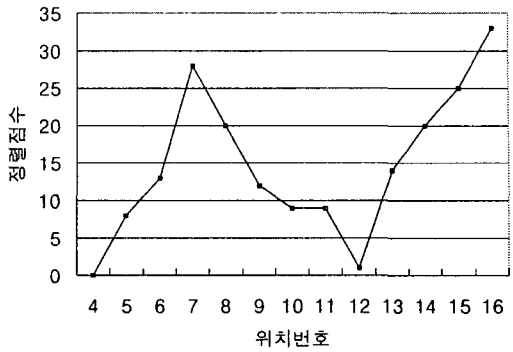
Smith-Waterman 알고리즘의 다이나믹 프로그래밍을 이용하여 각각의 셀의 값을 계산하여 저장한다. 문제를 간략화하기 위해서 gap penalty는 gap의 수에 비례하는 선형 점수를 부과하기로 한다. 최대 값을 갖는 노드에서 역 방향으로 추적하면서 정렬해 나가면서, 정렬 점수가 0이 되는 지점에서 정렬을 멈추면 지역 정렬된 두 서열을 얻게된다. 두 서열을 자세히 살펴보면, 두 서열의 초반 부분과 끝 부분에 각기 동일한 서열 'GAW'와 'CKAG' 부분이 포함되어 있음을 확인할 수 있다. 기본적인 Smith-Waterman 알고리즘은 그 안에 포함된 조각 부분을 포함하여 하나의 긴 정렬을 출력하게 된다 (<그림 2>). 가운데 포함된 조각은 -27점, 앞부분의 'GAW' 부분은 28점, 그리고 마지막 부분의 'CKAG'는 32을 나타내어 전체적으로는 33점의 최고점을 갖게된다. 가운데 부분의 -27점은 양옆의 점수와 비교하면 비슷한 절대값을 가지면서 부호가 반대되는 경우로 위에서 소개한 비일치 병합 정렬 조건이 만족되어 비일치 병합 정렬이 일어나게 됨을 확인할 수 있다.

서열 1	G	A	W	G	G	P	Q	R	C	K	A	G
서열 2	G	A	W	-	-	I	M	-	C	K	A	G

<그림 2> 서열 정렬 결과

정렬 경로를 통해 변화되는 점수의 변화를 보면 <그림 3>과 같다. 비일치 병합 정렬 구간 (7번-12번) 상에서는 값이 계속 내려가는 변화를 볼 수 있고 두 개의 최대치를 갖는 꼭지점이 생긴다 (7번과 16번). 이 두 점을 역으로 추적하여 정렬을 찾아 나가면 두 개의 지역 정렬이 생기는 것을 알 수 있다. 12 번 위치에서의 정렬 점수는 1로 0에 근사하므로 여기서 지역정렬을 하나 구하게되고, 다음 7번서 4번에 이르는 두 번째 지역정렬을 구하게 되면 총 60점의 두 지역 정렬을 찾아내는 결과를 갖게된다.

지역 정렬의 구간의 끝은 정렬 점수가 위치번호 역방향으로 진행하면서 계속 감소하다가 증가하는 방향으로 전환하려 할 때 그 점의 순방향 변화율을 보고 부분 서열정렬의 끝을 결정할 수 있겠다. 일반



<그림 3> 위치별 정렬점수 변화

적으로 앞의 알고리즘으로 구한 최대 값의 위치는 서로 가까이 이웃하지 않음을 알 수 있고 서로 충분히 떨어진 곳에서 지역정렬을 다수 개 구할 수 있게 된다.

5. 논의 및 결론

본 연구에서는 Smith-Waterman 알고리즘이 갖고 있는 문제점을 살펴보고 그 문제를 해결하는 알고리즘을 제안하였다. Zhang의 후처리[4] 과정 대신에 직접 역방향으로 추적하면서 다수개의 지역 정렬을 찾아내도록 고안했다는 점에서 좀더 개선된 알고리즘이라 판단된다. 본 연구는 효율성 개선을 증명할 좀더 체계적인 분석이 요구되며, 부분적으로 소규모 지역에서 최대 값을 갖는 점들을 합병하는 방법에 대해 좀더 개선할 필요가 있다.

참고문헌

[1] 김광수 "Bioinformatics 의 소개와 이용"

[2] W.B. Goad and M.I. Kanehisa, "Pattern recognition in nucleic acid sequences, i.e., a general method for finding local homologies and symmertires," Nucelic Acid Research, 10:247-263, 1982.

[3] M.S. Waterman and M. Eggert, "A new algorithm for best subsequence alignements with application to trna-rrna comparisons," J. of Comput. Biol., 197:723-728, 1987.

[4] Z. Zhang, P. Berman, T. Wiehe, and W. Mille, "Post-processing long pairwise alignments," Bioinformatics, 15:1012-1019, 1999.