품질 정보를 이용한 서열 배치 알고리즘[1)]

노강호[0] 박근수
서울대학교 전기·컴퓨터공학부
{khroh[0], kpark}@theory.snu.ac.kr

Sequence Alignment Algorithm using Quality Information

Kangho Roh[0] Kunsoo Park
School of Computer Science and Engineering

요 약

서열 배치 문제는 두 개의 서열에서 가장 유사한 부분을 찾는 문제이다. 이 문제를 푸는 알고리즘으로 가장 많이 쓰이는 것은 Smith-Waterman 알고리즘이다. Smith-Waterman 알고리즘은 동적 프로그래밍을 이용하여 두 서열에서 유사한 부분을 찾아낸다. 그러나 Smith-Waterman 알고리즘은 서열을 이루는 문자들의 품질 정보를 사용하지는 않는다. 각 문자가 얼마 정도의 신뢰도를 가지고 있는지를 나타내는 품질 정보는 생물학에서는 중요한 정보이다.

본 논문에서는 각 문자에 주어지는 품질이 서로 다를 때에, 품질 정보를 이용하여 가장 적합한 부분 배치를 찾아내는 알고리즘을 제시한다. 실제로 현재 서열 배치에 가장 많이 사용되고 있는 프로그램 중 하나인, Phred/Phrap에서 사용하는 LLR 값을 이용해서 비교했을 때, 본 논문에서 제시한 알고리즘은 기존의 Smith-Waterman 알고리즘보다 더 좋은 결과를 얻었다.

1. 서 론

최근에 분자 생물학이 발전하면서 서열(sequence)의 관계를 알아볼 필요가 생겼다. 이 때문에 중요해진 것이 서열 배치 문제이다. 서열 배치 문제는 두 개의 서열에서 가장 유사한 배치를 찾는 문제이다.

Smith와 Waterman은 부분 배치 문제를 찾는 가장 일반적인 알고리즘을 제안하였다[1][3]. Smith-Waterman 알고리즘은 두 서열의 길이의 곱의 크기를 갖는 2차원 배열을 구하여 가장 적합한 부분배치를 찾는다.

그러나, Smith-Waterman 알고리즘은 부분배치를 구할 때에 각 문자의 품질 정보(quality information)를 고려하지 않는다. 품질 정보는 그 문자를 얼마나 신뢰할 수 있는지를 알려준다. Fragment Assembly에서 유전자 서열에는 각 문자의 품질 정보가 있다. 그런데 Smith-Waterman 알고리즘은 이 정보를 이용하지 않아서, 생물학적으로 가장 적합한 부분 배치를 찾아내지 못하는 경우가 있다.

본 논문은 3장에서 서열의 품질 정보를 이용하여 Smith-Waterman 알고리즘을 개선한 알고리즘을 제안하고, 4장에서 실험결과를 밝힌다.

2. 이전 연구

2.1.1 전체 배치(global alignment)

서열(Sequence)은 알파벳 Σ에 속한 문자들을 연속해서 나열한 것이다. 공백은 $\Delta \notin \Sigma$로 나타낸다. 서열 a에 대해서 $a = a_1 a_2 \ldots a_n$ ($a_i, 1 \le i \le n$는 문자)이라고 하자. 이 때 서열 a의 길이 $|a| = n$이고, 서열 a의 부분 즉 $a_i \ldots a_j (1 \le i < j \le n)$을 a의 부분 서열이라고 한다.

두 서열 a, b를 배치할 때, 배치의 점수가 가장 큰 배치를 찾는 것을 전체 배치라고 한다. 다음은 서열 $AACCT$와 $ACCCT$ 의 전체 배치의 예이다.

$$a^* = AA\Delta CCT$$
$$b^* = A\Delta CCCT$$

배치의 점수는 다음과 같이 구한다. 배치에서 문자 (a^*_i, b^*_i)의 대응은 다음과 같은 3가지가 있다.

① 일치 : $a^*_i = b^*_i$

② 불일치 : $a^*_i \neq b^*_i$

③ 갭 : a^*_i, b^*_i중 하나가 Δ

일치, 불일치, 갭에 대해서 각각 점수가 있다. 일치의 경우는 (+)점수, 불일치와 갭의 경우는 (-) 점수를 준다.

배치의 점수는 배치에서 각각 문자쌍의 점수의 합이 된다. 배치 점수가 가장 큰 배치를 찾는 문제를 전체 배치 문제라고 한다.

2.1.2 부분 배치

두 서열의 전체 배치를 보면, 전체 값은 좋지 않지만, 배치의 일부분의 값은 아주 좋은 경우가 있다. 생물학적으로 볼 때, 이런 부분이 더 중요하다. 그래서 점수가 높은 부분

1) 본 논문은 IMT-2000 출연금 정보통신 선도기술개발사업 "농생물 유전체 자동화 분석 시스템 개발"의 지원을 받았음

배치를 찾는 것이 필요하다. 두 서열에서 배치 점수가 높은 부분 배치를 찾는 것을 부분 배치 문제라고 한다. 부분 배치를 찾는데 가장 널리 사용되는 알고리즘이 Smith-Waterman 알고리즘이다.

2.1.3 Smith-Waterman 알고리즘

Smith-Waterman 알고리즘은 두 서열 a, b가 있을 때, $(|a|+1)(|b|+1)$의 2차원 배열을 동적으로 구하여 배치 점수가 높은 부분 배치를 찾는다.

Smith-Waterman 알고리즘은 다음의 점화식을 이용하여 부분배치를 찾는다.

$$H_{i,0}=0 \quad (0\le i\le|a|)$$
$$H_{0,j}=0 \quad (0\le j\le|b|)$$

$$H_{i,j}=\max\{0, H_{i-1,j-1}+s(a_i,b_j), H_{i,j-1}-\mu, H_{i-1,j}-\mu\}$$

$s(a_i,b_j)$: a_i, b_j의 배치 점수

μ : 갭 벌점

Smith-Waterman 알고리즘은 두 서열의 길이를 m, n이라고 할 때, $m\times n$ 크기의 2차원 배열을 만들게 되므로, 시간 복잡도는 $O(mn)$이고, 공간 복잡도는 $O(m)(m\le n)$인 방법이 알려져 있다.

2.1.4 품질 정보

실제 유전자 정보를 보면 시퀀서가 문자를 읽을 때, 정확히 어떤 문자라고 결정하기 어려운 경우가 있다. 따라서 어떤 문자가 어느 정도의 신뢰도를 가지고 있는지를 가리키는 정보가 필요하다. 이 값을 품질이라고 한다.

각 문자에 품질 값이 부여된다. 그래서 품질이 좋으면 그 문자는 확실하다고 생각할 수 있고, 품질이 떨어지면 그 문자라고 확신할 수 없다. 품질 값은 1에서 99사이의 자연수 값을 갖는다. 품질 값을 x라고 하면, 그 문자가 틀린 값을 가질 확률은 $10^{-(x/10)}$이다. 그러므로 만약 $a_i=A$이고 품질이 40인 경우와 $a_j=A$이고, 품질이 60인 경우가 있다면 a_j가 더 확실히 A라고 볼 수 있다.

3. 제안 방법

3.1 품질 정보를 이용한 배치

앞에서 설명한 품질 정보를 이용하여 배치를 해 보자. 각 문자의 품질에 따라서 다음과 같은 경우를 생각해 볼 수 있다.

① 일치 : 서열을 배치했을 때, 문자 A가 일치했다고 하자 (괄호 안은 품질 값).

i)	ii)	iii)
A(40)	A(60)	A(50)
A(50)	A(80)	A(90)

위 예를 보면 i), ii), iii) 모두 문자쌍이 일치되어 있다. 그러나 i)의 경우보다 ii)의 경우가 품질이 더 높게 나와 있다. 그렇다면 i)보다는 ii)가 배치 점수가 높아야 한다. 그리고 iii)의 경우보다는 ii)의 경우가 배치 점수가 높아야 한다.

② 불일치

i)	ii)	iii)
A(30)	A(40)	A(20)
C(50)	C(70)	C(90)

위 예를 보면 i)보다는 ii)가 품질 값이 큼을 알 수 있다. 생물학적으로는 품질이 높은 두 문자쌍이 불일치인 경우는 좋지 않다. 따라서 위의 경우에 i)의 배치가, ii)보다 배치 점수가 높아야 한다. 또한 ii)와 iii)을 살펴보면, ii) 보다는 iii)이 A의 품질은 더 나쁘고, C의 품질은 좋은 경우이다. 이 때는 iii)의 경우가 ii)보다 배치 점수가 높아야 한다.

③ 갭

i)	ii)
A(30)	A(40)
-	-

위의 예를 보면 i)보다는 ii)의 A가 품질이 더 좋다. 품질이 더 좋은 문자가 갭으로 나타나는 것은 좋지 않으므로, i)이 ii)보다 더 좋은 배치이다.

3.2 품질을 이용한 배열 알고리즘

앞에서 보인 품질 정보를 이용하여 생물학적으로 더 좋은 배치를 찾는 알고리즘을 제시한다.

3.2.1 제안 알고리즘

Smith-Waterman 알고리즘의 점화식을 변경하여 새로운 알고리즘을 제시하였다. 새 알고리즘은 다음의 점화식을 이용한다.

$$H_{i,0}=0 \quad (0\le i\le|a|)$$
$$H_{0,j}=0 \quad (0\le j\le|b|)$$

$$H_{i,j}=\max\begin{cases} H_{i-1,j-1}+s(a_i,b_j)\times\left(\dfrac{ave(q_{a,i},q_{b,j})}{\max Q}\right) \\ H_{i-1,j}-\left(\mu\times\dfrac{q_{a,i}}{\max Q}\right) \\ H_{i,j-1}-\left(\mu\times\dfrac{q_{b,j}}{\max Q}\right)\end{cases}$$

$\max Q$: 품질 중에서 나올 수 있는 최대 값

$ave(q_{a,i},q_{b,j})=\dfrac{2q_{a,i}q_{b,j}}{q_{a,i}+q_{b,j}}$: $q_{a,i}, q_{b,j}$의 조화 평균 값

$q_{a,i}$: 문자 a_i의 품질

위의 점화식에서 $\max Q$는 각 문자에 나올 수 있는 최대 품질 값을 말한다(본 논문의 이후에는 이 값을 100으로 가정하자).

제시한 알고리즘은 품질에 따라서 서로 일치, 불일치, 갭의 점수를 다르게 부여하였다.

① 일치/불일치 : 높은 품질의 문자가 일치일 경우에는 높은 점수를 주어야 하고, 반대로 불일치일 경우에는 벌점을 많이 주어야 하기 때문이다. 그래서 일치와 불일치의 경우 점수는 원래 점수에 두 문자의 품질의 조화평균을 곱한 값을 사용하였다. 조화 평균을 이용한 이유는 위의 3.1의 예 ②의 ii)와 iii)를 구분하기 위함이다.

② 갭 : 갭의 경우에는 갭인 부분의 점수가 높으면 더 많은 점수를 감하였다. 왜냐하면 높은 품질의 문자가 갭이 되는 것은 피하는 것이 좋기 때문이다. 그래서 원래의 갭 벌점에 품질을 곱한 값을 사용하였다.

3.2.2 제시한 알고리즘의 시간 복잡도

본 논문에서 제시한 알고리즘은 기본적으로 Smith-Waterman과 유사한 방법을 이용하고 있다.

따라서 두 서열의 길이가 $m=|a|, n=|b|$라고 하면 시간 복잡도는 $O(mn)$이고, 공간복잡도는 $O(n)(n<m)$인 방법이 알려져 잇다.

4 실험 결과

새롭게 제안한 방법을 이용한 실험결과이다. 배치의 성능을 알아보기 위해서 본 논문에서는 가장 널리 사용되는 Sequence Assembly 프로그램 중 하나인 Phrap에서 사용하는 LLR 값을 이용하였다[4].

4.1 LLR 값

Phrap은 두 문자열의 배열과 각 문자열의 품질을 이용하여 LLR 값을 구한다. LLR 값을 구하는 방법은 다음과 같다.

서열 a, b의 배치를 a^*b^*라고 하자. a^*b^*의 i번째 문자 a_i^*, a_i^*의 품질을 각각 $q_{a,i}^*, q_{b,i}^*$라고 하자. 그러면 다음 3가지 경우에 문자 쌍 a_i^*, b_i^*의 배치 값은 다음과 같다.

① 일치 : $10\times\log_{10}(1/0.95)\approx 0.223$

② 불일치 : $mismatchLLR[\min(q_{a,i}^*, q_{b,i}^*)]$

③ 갭 : $mismatchLLR[q_{a,i}^*]$: $b_i^*=\Delta$ 일 때

$mismatchLLR[q_{b,i}^*]$: $a_i^*=\Delta$ 일 때

$$mismatchLLR[i] = 10\times(-10\times\log_{10}(0.05+(0.95\times e^{(i\times(-\ln(10)/10))}-i)))$$

배치의 각 자리 문자 쌍 위의 값을 구해서 모두 더해 주면 배치 전체의 LLR 값이 나온다. 서열 a, b의 배치가 서로 생물학적으로 좋다면, 배치의 LLR 값이 높게 나온다.

4.2 실험 결과

다음은 몇 가지 서열 쌍에 대한 실험 결과이다. 각 서열 쌍을 Smith-Waterman 알고리즘과, 제안한 알고리즘을 이용하여 구한 부분 배치의 LLR 값을 구해 보았다. 모두 3번 실험을 하였다. 각 실험에서 사용된 서열의 길이와 Smith-Waterman 알고리즘과 새롭게 제안한 알고리즘이 구한 배치의 길이와 LLR 값을 조사하였다.

표 1

실험 번호	서열 길이	Smith-Waterman		제안 알고리즘	
		배치 길이	LLR 값	배치 길이	LLR 값
1	60	45	-486	48	-125
2	1500	845	-5972	862	-5410
3	15000	11108	-154468	11422	-139023

앞의 결과를 보면, 새롭게 제안한 알고리즘으로 구한 LLR 값이 더 높게 나온 것을 알 수 있다. 이로써 새롭게 제안한 방법이 Smith-Waterman 알고리즘보다, 생물학적으로 더 좋은 배치를 찾음을 알 수 있다.

5 결론 및 토의

기존의 Smith-Waterman 알고리즘은 품질 정보를 이용하지 않기 때문에, 생물학적인 면에서 응용할 때는 적합하지 않은 결과를 내는 경우가 있었다.

그러나 본 논문에서는 이 부분을 보완하여 품질 정보를 첨가한 점화식을 제시하였고, 널리 쓰이는 Sequence Assembly 프로그램인, phrap에서 사용하는 LLR 값을 구해 보았을 때, 생물학적으로 더 좋은 배열을 찾아냄을 알 수 있었다.

참고 문헌

[1] T.F. Smith and M.S. Waterman, Identification of common molecular subsequences, *Journal of Molecular Biology* 147 (1981), 195-197.

[2] O. Gotoh, An improved algorithm for matching biological sequences, *Journal of Molecular Biology* 162 (1982), 705-708.

[3] M.S. Waterman, *Introduction to Computational Biology*, Chapman & Hall, London (1995).

[4] P. Green, *Documentation for phrap*, Genome Center, University of Washington, http://www.phrap.org/phrap.docs/phrap.html