복수 염기서열 정렬을 위한 휴리스틱에 관하여

김진[+] 장연아[°] 최홍식[°]
건국대학교 전산과학과[+], 한림대학교 컴퓨터공학과[°],

On heuristics for multiple sequence alignment

Jin Kim, Yeonah Chang, Hongsik Choi
Dept of Computer Science, Konkuk University
Dept. of Computer Engineering, Hallym University

< 요 약 >

복수 염기서열 정렬(multiple sequence alignment)은 염기서열들 사이의 진화관계, 단백질의 구조와 기능에 관한 연구에 필수적인 도구이다. 다이나믹 프로그래밍(dynamic programming) 방법은 대부분의 경우에 있어 최적의 염기서열 정렬 결과를 제공할 수 있다. 그러나 그것이 사용하는 갭 비용함수 때문에 특별한 경우에 최적의 염기서열 정렬을 만들어 내지 못한다. 본 논문에서는 다이나믹 프로그래밍에 의해 획득된 염기서열을 개선하기 위한 휴리스틱 방법을 제안한 후, 실제 단백질 데이터를 가지고 성능 분석을 한다.

1. 서론

생물학 역사상 가장 중요한 프로젝트의 하나인 Human Genome Project의 기본적인 목표는 인체의 게놈과 생명체의 유전자 염기서열(molecular sequence)의 인식을 목표로 하고 있다. 이 프로젝트에 의해 발생되는 엄청난 양의 염기서열 관련 데이터는 의약과 생물학 분야에 절대적인 영향력을 미치고 있으며 이러한 추세는 더욱 심화될 것이라 예상된다. 이러한 염기서열 관련 데이터를 처리하여 중요한 생물학적 정보를 얻기 위해서는 전산학의 도움이 필수적이다. 전산학에서 염기서열은 스트링으로 간주된다. 본 논문에서는 게놈 프로젝트에서 파생된 가장 중요한 문제 중에 하나인 복수 염기서열 정렬 문제에 대하여 논한다[3][4][5][6][9].

염기서열 정렬은 단백질(protein), DNA 및 RNA의 생물학적 분석에 관련된 문제이다. 생물학자들은 두 개 혹은 그 이상의 유전자 염기서열들을 자연적인 생물학적 측정 단위(natural biological metric)를 사용하여 최소값을 가지는 염기서열 정렬을 획득하려 한다. 최소값을 가지는 염기서열 정렬은 미지의 염기서열(unknown sequence)의 확인을 위한 유전자 데이터베이스 검색, 유사한 단백질 분자구조와 관련된 패턴 인식, 염기서열의 기능 및 기능과 진화에 관한 중요한 정보를 획득하기 위하여 사용된다.

염기서열 정렬 기법 중 다이나믹 프로그래밍을 사용하면 최적해(optimal) 혹은 최적해에 가까운(near optimal) 값을 가지는 복수 염기서열 정렬을 얻을 수 있다[8]. 그러나 정렬하고자 하는 염기서열의 개수가 증가함에 따라 필요한 실행 시간도 지수함수적으로 증가하기 때문에 4~7개 이상의 염기서열 정렬에는 효율적이지 못하며, 특정한 비용함수에는 사용할 수 없다는 단점이 있다

본 논문에서는 다이나믹 프로그래밍 방법(MSA)을 사용하여 획득된 염기서열을 휴리스틱 방법을 사용하여 개선해 보도록 하였고, 실험 데이터로 실제 단백질 염기서열을 사용하도록 하였다. 실험 결과 짧은 시간으로 최적에 가까운 비용을 가지는 복수 염기서열 정렬을 얻을 수 있었다.

본 논문은 다음과 같이 구성되어 진다. 2장에서 복수 염기서열 정렬에 대하여 설명을 하고, 3장에서는 다이나믹 프로그래밍을 이용하여 염기서열 정렬을 하는 방법과 문제점에 대하여 살펴본 후, 4장에서 정렬 결과를 개선할 수 있는 휴리스틱 방법(Segmented Refinement Algorithm)을 제안한 후, 5장에서 실험결과와 결론을 맺도록 한다.

2. 복수 염기서열 정렬

복수 염기서열 정렬은 염기서열들 간의 element들의 대응관계를 알아내는 것과 관련이 있다. 복수 염기서열 정렬은 생물학적 데이터 분석에 있어 중요한 작업으로 미지의 염기서열의 확인을 위한 데이터베이스 검색, 유사한 분자구조와 관련된 패턴인식, 염기서열의 기능 및 진화에 관한 중요한 정보를 획득하기 위하여 사용되어진다. 이러한 염기서열의 정렬을 위해 정렬의 양호성(goodness)을 측정할 수 있는 비용함수(f)가 필요하다. 특정 염기서열들의 집합에 대해, 최적 정렬(optimal alignment)은 최소비용(minimum cost)을 가지는 정렬이다. 최소비용을 가진 정렬을 얻기 위해 각 염기서열들의 원소들 사이에 매치(match), 삽입(insert), 교체(substitution), 삭제(deletion)의 방법이 사용된다. 복수 염기서열 정렬 문제는 다음과 같이 정의할 수 있다.

정의1

- 알파벳(alphabet) Σ는 문자들과 널(null, '-')로 이루어진 유한 집합이다. 널의 생물학적인 의미는 하나의 염기서열에서 삽입이 발생했거나, 혹은 다른 염기서열에서 삭제가 발생했다는 것이다
- 염기서열(sequence)은 문자로 이루어진 유한한 길이의 스트링이다.
- 염기서열 $S_1 = s_{11}s_{12}\cdots s_{1n1}, S_2 = s_{21}\cdots s_{2n2}, \cdots, S_k = s_{k1}\cdots s_{knk}$은 각각 길이 $n_1, n_2, \cdots, n_k$의 k개의 입력 염기서열들이며 염기서열 정렬은 널이 패드된(padded) 동일한 길이 l의 의사염기서열 $S_1' = s_{11}'s_{12}'\cdots s_{1l}', S_2' = s_{21}'\cdots s_{2l}', \cdots, S_k' = s_{k1}'s_{k2}'\cdots s_{kl}'$을 얻는 것이다. 이 때 패드된 의사 염기서열(pseudo-sequence) S_i'로부터 널들을 제거하면 원래의 염기서열 S_i를 얻을 수 있다. 이때 삽입되어지는 널의 위치와 길이 l과 관련되어 많은 종류의 복수 염기서열 정렬이 얻어질 수 있다.
- 복수 염기서열 정렬 문제는 정렬에 대한 최적도의 기준이 주어졌을 때, 최적 비용(optimal cost)을 가지는 복수 염기서열 정렬을 찾는 문제이다.

3. 다이나믹 프로그래밍에 의한 복수 염기서열 정렬

염기서열 정렬에 사용되어 지는 비용함수는 정렬의 질(quality)에 대한 명백한 측정수단이어야 한다. Altschul[1][2]은 복수 염기서열 정렬을 위한 몇 가지 비용함수를 분류하였다. 이들 비용함수는 교체비용(substitution cost)과 갭비용(gap cost)로 구성된다. 교체비용은 DNA나 단백질 분자를 대표하는 글자들을 정렬하는데 드는 비용이다. Altschul은 생물학적으로 자연스러운 갭비용함수로서 natural gap cost를 제안하였다. 그러나 다이나믹 프로그램은 natural gap cost를 갭비용함수로 사용하는 경우 매우 긴 실행시간을 필요로 한다. 다이나믹 프로그램은 속도의 향상을 위해 quasi-natural gap cost를 갭비용함수로 사용하였다. quasi-natural gap cost는 하나의 염기서열의 갭이 다른 염기서열에 완전히 포함되어지는 경우 실제보다 갭을 하나 더 센다는 점을 제외하고는 natural gap csot와 동일하다. 다이나믹 프로그램은 갭비용함수로 quasi-natural gap cost를 사용하므로 최적이 아닌 염기서열 정렬을 최적의 정렬로 잘못 판단하게 된다.

4 Segmented Refinement Algorithm(SRA)

이번 장에서는 다이나믹 프로그램의 결과 중 완전히 포함되어지는 형태의 갭을 가질 수 있는 서브스트링만을 발췌하여, 그 서브스트링에 대해서 natural gap cost의 갭비용함수를 적용하여 최적의 염기서열 정렬을 만들어 낼 수 있는 SRA방법에 대하여 설명하도록 한다.

4-1 입력 서브스트링 선택전략

다이나믹 프로그램의 결과 생성되어지는 염기서열 정렬은 최적의 염기서열 정렬과 매우 유사하다[7].

```
IIGGVESIPHSRPYMAHLDIVTEKGLRVICGGFLISRQFVLTAAHC
IVGGTNSSWGEWPWQVSIQVKLTAQR-HLCGGSLIGHQWVLTAAHC
IVNGEEAVPGSWPWQVSLQDKTGF---HFCGGSLINENWVVTAAIC
IVGGYTCGANTVPYQVSL--NSGY---IFCGGSLINSQFVVSAAHC
VVGGTEAQRNSWPSQISLQYRSGSSWAHTCGGTLIRQNWVMTAAHC
VVCCTRAAQGEFPFMVRL--SWG-------CGGALYAQDIVLTAAHC
```

그림 1 다이나믹 프로그래밍에 의한 염기서열 정렬

```
IIGGVESIPHSRPYMAHLDIVTEKGLRVICGGFLISRQFVLTAAHC
IVGGTNSSWGEWPWQVSLQVKLTAQ-RHLCGGSLIGHQWVLTAAHC
IVNGEEAVPGSWPWQVSLQDKTG---FHFCGGSLINENWVVTAAHC
IVGGYTCGANTVPYQVSLNSG-----YHFCGGSLINSQWVVSAAHC
VVGGTEAQRNSWPSQISLQYRSGSSWAHTCGGTLIRQNWVMTAAHC
VVCCTRAAQGEFPFMVRLSWG---------CGGALYAQDIVLTAAHC
```

그림 2 최적의 염기서열 정렬

그림1과 그림 2는 각각 다이나믹 프로그램을 이용한 염기서열 정렬과 최적의 염기서열 정렬을 보여주고 있다. 그림을 보면 두 개의 염기서열 정렬은 널이 위치한 서브스트링을 제외한 나머지 전체 염기서열 부분이 같다는 것을 쉽게 알 수 있다. 따라서 다이나믹 프로그램이 만들어 낸 염기서열 정렬에서 널이 위치한 서브스트링에 대해 최적의 정렬을 만들 수 있다면 쉽게 전체 염기서열을 최적 혹은 최적에 가깝도록

정렬할 수 있다. 전체 염기서열 중 선택되어지는 염기서열 서브스트링은 다음의 두 가지 조건을 만족하는 스트링이다.

- 갭이 존재하는 서브스트링의 수가 2이상 (1)
- 가장 많은 널의 개수와 '0'을 제외한 가장 적은 널의 개수의 차기 2이상 (2)

조건 (1)과 (2)를 만족하도록 발췌되어진 서브스트링의 길이는 갭비용 함수로 natural gap cost를 사용할 수 있을 만큼 충분히 짧은 길이이다.

4-2 서브스트링 염기서열 정렬을 위한 SRA 방법

정렬하고자 하는 서브스트링의 길이는 원래의 염기서열 길이에 비해 매우 작은 길이를 갖는다. 그렇기 때문에 서브스트링에 대해서 brute-force 방식을 조금 개선한 휴리스틱 알고리즘을 이용하여 만족할 만한 시간 안에 최적 혹은 최적에 가깝도록 정렬할 수 있다.

원칙적으로 최적의 염기서열 정렬을 얻어내기 위해서는 각 염기서열에서 널이 이동하여 생길 수 있는 모든 가능한 경우를 다 살펴보아야 한다. 그러나 다이나믹 프로그램의 결과를 입력으로 하는 알고리즘의 특성상 고려해야 할 후보 해 집단의 범위는 축소되어질 수 있다. SRA방법은 전단계의 고정 널의 위치가 다음 단계의 널의 위치를 제한하도록 하는 방법이다. 이런 식으로 널의 위치를 제한하게 되면 염기서열 내에 널이 삽입되어 생겨나는 염기서열 정렬의 가지수를 줄일 수 있게 된다. 또한 같은 위치의 널은 서로 상쇄되어질 수 있다는 갭비용함수의 정의에 따라 비용이 적은 염기서열 정렬을 유도할 수 있다.

5. 실험 결과 및 결론

SRA방법은 MSA에 의해 획득된 염기서열 정렬을 개선하기 위한 것이다. MSA는 갭비용함수로 quasi-natural gap cost를 사용하나 SRA는 natural gap cost를 사용한다.

```
GDSGGPLLCAGV-----AHGIVSYG      GDSGGPLLCAG-----VAHGIVSYG
GDSGGPLVCKHN-GMWRLVGITSWG      GDSGGPLVCKHN-GMWRLVGITSWG
GDSGGPLVCKKN-GAWTLVGIVSWG      GDSGGPLVCKKN-GAWTLVGIVSWG
GDSGGPVVCSGK-----LQGIVSWG      GDSGGPVVCSG-----KLQGIVSWG
GDSGGPLHCLVN-GQYAVHGVTSFV      GDSGGPLHCLVN-GQYAVHGVTSFV
GDSGGPMFRKDNADEWIQVGIVSWG      GDSGGPMFRKDNADEWIQVGIVSWG
          (a)                            (b)
```

그림 3 염기서열 정렬

그림 3은 MSA에 의한 염기서열 정렬(a)과 SRA 방법으로 개선된 염기서열 정렬(b)을 보여주고 있다. 그림에서 표시된 부분이 SRA 방법으로 개선되어진 부분이다. 그림 3의 (a)의 비용은 5038이고 (b)는 5030이다.

복수 염기서열 정렬에 사용되어지는 최적화 방법으로 다이나믹 프로그램이 있다 그러나 다이나믹 프로그램은 갭비용함수로 quasi-natural gap cost를 사용하기 때문에 염기서열의 갭이 다른 염기서열에 완전히 포함되어지는 형태의 갭을 최적 염기서열 정렬로 가지는 경우에 대해서는 최적 염기서열 정렬을 만들어 낼 수가 없다. 이를 위해 다이나믹 프로그램으로 만들어진 염기서열 정렬 중 일부 서브스트링을 발췌하여 발췌된 서브스트링에 대해 최적의 정렬을 만들어 낸 결과 전체적인 염기서열을 최적의 염기서열 정렬에 접근하도록 만들 수 있었다.

참고문헌

[1] Altschul, S.F. 1989. Gap costs for multiple sequence alignment. Journal of Theor. Biol. 138, 297-309.

[2] Altschul, S.F. and D.J. Lipman. 1989. Trees, stars and multiple biological sequence alignment. SIAM J. Appl. Math. 49, 153-161

[3] Bacon, D.G. and W.F. Anderson. 1986. Multiple sequence alignment. J. Mol. Biol. 191, 153-161.

[4] Chan, S.C., A.K.C. Wong and D.K.Y. Chiu. 1992. A survey of multiple sequence comparison methods. Bull. Math Biol. 54, 563-598.

[5] Carrillo, H. and D. Lipman. 1988. The multiple sequence alignment problem in biology. SIAM J Appl. Math. 48, 1073-1082.

[6] Fickett, J.W 1987. Fast optimal alignment Nucleic Acids Res. 12, 175-180.

[7] Kim, J., S. Pramanik and M.J.Chung. 1994. Multiple sequence alignment using simulated annealing. Comp. Appl. Biosci. 10, 419-426.

[8] Lipman, D.J., S.F. Altschul and J.D. Kececioglu. 1989. Tool for multiple sequence alignment. Proc. Natl. Acad. Sci. USA. 86, pp 4412-4415.

[9] Waterman, M.S. 1984. Ceneral methods of sequence comparison. Bull. Math. Biol. 46, 473-500.