환형 문자열에 대한 쌍합 기반의 다중서열배치

이태형[O,1,*] 나중채[2,**] 심정섭[3,***] 박근수[1,*]
[1]서울대학교 [2]세종대학교 [3]인하대학교
thlee@theory.snu.ac.kr, jcna@sejong.ac.kr, jssim@inha.ac.kr, kpark@theory.snu.ac.kr

Multiple Sequence Alignment for Circular Strings based on Sum-of-Pairs

Taehyung Lee[O,1], Joong Chae Na[2], Jeong Seop Sim[3], and Kunsoo Park[1]
[1]Seoul National University, [2]Sejong University, [3]Inha University

1. 서론

환형문자열(circular string)은 문자열의 첫 글자와 마지막 글자가 연결되어 고리 모양을 이루는 문자열로서 자연계에서 박테리아나 미토콘드리아의 DNA 등에서 흔히 발견된다. 다중서열배치 문제는 주어진 문자열 집합에서 유사한 부분을 중심으로 모든 문자열을 배치하는 문제로 분자 생물학 및 생물 정보학에서 여러 가지 연구에 응용되는 중요한 문제이다. 기존의 다중서열배치 연구는 주로 선형문자열(linear string)을 대상으로 이루어져왔다. 반면, 환형문자열에 대한 다중서열배치 연구는 비교적 최근에서야 이루어지기 시작하였다 [1, 2]. 이들 논문에서는 목적함수로 쌍합을 이용하고 범용 다중서열배치 알고리즘인 clustalW [3]를 사용하였다.

본 논문에서는 환형문자열에 대하여 해밍거리(Hamming distance) 기반의 쌍합(Sum-of-pairs) 목적함수를 최소화하는 다중서열배치(multiple sequence alignment) 문제를 정의하고, 이를 해결하는 두 가지 알고리즘을 제안한다. 첫 번째 알고리즘은 이산 합성곱(discrete convolution)을 이용한 알고리즘으로 작은 크기의 환형문자열 집합에 대해서 효율적이다. 두 번째 알고리즘은 인접 배치간의 관계를 이용하며, 첫 번째 알고리즘에 비해 공간적 측면에서 효율적이다.

2. 본 론

길이 n인 **환형문자열** S는 문자 n개로 이루어진 서열로, 첫 번째 문자 $S[0]$가 마지막 문자 $S[n-1]$ 다음으로 연결된 고리 형태의 문자열이다. 길이 n인 환형문자열 S에서 $r(=0,1,...n-1)$번째 문자로부터 시작되는 선형문자열을 r번째 개체라 정의하고 $S(r)=S[r]S[r+1 \bmod n]...S[r+n-1 \bmod n]$로 표기한다. 길이 n인 환형문자열 K개로 (일반적으로 $K \ll n$) 이루어진 집합 $\mathbb{S}=\{S_1, S_2, \ldots, S_K\}$가 주어질 때, $\mathbb{S}$에 대한 **배치**는 각각의 문자열에서 임의의 개체를 선택하여 만든 선형문자열의 집합으로 정의되고, 다음과 같이 K-튜플로 표현한다. K개 인덱스 $0 \le r_1, r_2, \ldots, r_K < n$에 대해, 개체 $S_1(r_1), S_2(r_2), \ldots S_K(r_K)$은 환형문자열 집합 $\mathbb{S}$에 대한 배치 $\rho = (r_1, r_2, \ldots, r_K)$를 나타낸다. 해밍거리는 두 문자열 사이에서 문자의 교체 연산만을 고려하기 때문에 $r_1=0$으로 가정할 수 있고, 따라서 집합 $\mathbb{S}$에 대한 서로 다른 배치의 수는 최대 n^{K-1}이다.

임의의 배치가 갖는 적절성을 판단하기 위하여 본 논문에서는 해밍거리와 쌍합을 각각 거리함수와 목적함수로 이용한다. **해밍거리**는 한 문자열을 다른 문자열로 바꿀 때, 사용되는 교체(replace) 연산의 최소 횟수로 정의된다. **쌍합**(Sum-of-pairs)은 주어진 배치에서 모든 개체 쌍의 해밍거리의 합으로 정의된다. 즉, 환형문자열 집합 $\mathbb{S}$ 및 배치 ρ에 대한 쌍합을 $SP_{\mathbb{S}}(\rho)$라 할 때, $SP_{\mathbb{S}}(\rho)=\Sigma_{1\le i<j\le K} d(S_i(r_i), S_j(r_j))$로 정의된다. 쌍합을 최소화하는 환형문자열의 다중서열배치 문제를 다음과 같이 정의한다.

> **정의 1. 쌍합을 최소화하는 환형문자열의 다중서열배치 문제**
> 길이 n인 환형문자열 K개로 이루어진 집합 $\mathbb{S}=\{S_1, S_2, \ldots, S_K\}$가 주어질 때, 모든 가능한 배치 ρ 중 쌍합 $SP_{\mathbb{S}}(\rho)$를 최소화하는 배치 $\rho^*=(r_1, r_2, \ldots, r_K)$ 및 그 값 $SP_{\mathbb{S}}(\rho^*)$을 찾아라.

본 논문에서는 환형문자열의 다중서열배치 문제를 풀기 위한 두 가지 알고리즘을 제안한다. **알고리즘 A**는 먼저 합성곱(convolution) 연산을 이용하여 모든 환형문자열 쌍에 대하여, 각 쌍이 가질 수 있는 모든 배치에 대한 해밍거리를 $O(K^2|\Sigma|n\log n)$ 시간을 사용하여 계산한다. 합성곱은 두 함수를 곱한 값을

* 본 연구는 기초기술연구회의 NAP 과제 지원으로 수행되었습니다. 이 연구를 위해 연구장비를 지원하고 공간을 제공한 서울대학교 컴퓨터연구소에 감사 드립니다. ** 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국학술진흥재단(또는 한국과학재단)의 지원을 받아 수행된 연구임(No. 2010-0015624). *** 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2010-0028134).

적분하여 새로운 함수를 구하는 수학 연산자로 수학 및 공학의 여러 분야에서 널리 사용되고 있다. **알고리즘A**에서는 길이 n인 두 환형문자열 간의 모든 가능한 배치에 대한 해밍거리 계산을 이산 합성곱(discrete convolution)으로 모델링하고 이를 고속 푸리에 변환(Fast Fourier Transform, FFT) 방법으로 계산하여 연산 시간을 $O(n^2)$에서 $O(|\Sigma|n\log n)$으로 단축한다. 이렇게 계산된 K^2n개의 해밍거리 값으로부터 모든 n^{K-1}가지 배치에 대한 쌍합을 구하는데 $O(K^2n^{K-1})$ 시간이 필요하지만, 중복 계산되는 부분을 제거하면 이를 $O(K\max(K,n)n^{K-2})$으로 개선할 수 있다. 일반적으로 $K \ll n$이므로 알고리즘A의 전체 시간 복잡도는 $O(K^2|\Sigma|n\log n + Kn^{K-1})$이 된다. 집합 $\mathbb{S}$의 크기 K가 3 이하로 작은 경우, 각각의 쌍이 가질 수 있는 모든 배치 상의 해밍거리를 단순히 계산하는 $O(K^2n^2)$ 시간의 알고리즘보다 이산 합성곱을 이용하는 $O(K^2|\Sigma|n\log n)$ 시간의 **알고리즘A**가 더 효율적이다. 한편, **알고리즘A**는 K^2n개의 해밍거리 값을 모두 저장하기 위한 $K\times K\times n$의 3차원 테이블을 사용하므로 공간 복잡도는 $O(K^2n)$이다.

알고리즘B는 임의의 배치 ρ에 대해 환형문자열 개체가 정렬되는 각 위치 별로 문자의 개수를 세는 계수 배열을 구하고, ρ와 인접한 배치 ρ'에 대한 쌍합을 배치 ρ에 대한 쌍합 및 계수 배열을 이용하여 $O(n)$ 시간에 구하는 방법을 사용한다. $\rho = (r_1=0, r_2, ..., r_K)$와 인접한 배치 ρ'은 임의의 인덱스 k $(1<k\le K)$ 및 상수 c $(0 < c \le n-1)$를 선택하여 ρ에서 r_k를 $(r_k+c) \bmod n$으로 바꾼 새로운 배치 $\rho' = (r_1, ..., (r_k+c) \bmod n, ...)$로 정의된다. 임의의 배치 $\rho=(r_1, r_2, ..., r_K)$가 주어질 때, 배치된 개체 $S_1(r_1), S_2(r_2), ... S_K(r_K)$를 나란히 놓고, 각각의 위치 i $(0 \le i \le n-1)$에서 각 문자 $\sigma \in \Sigma$를 세어 계수 배열 $C_\rho[i, \sigma]$에 저장한다고 하자. 즉, $C_\rho[i, \sigma]=|\{k : S_k(r_k)[i]= \sigma, 1 \le k \le K\}|$로 정의한다. 해밍거리 연산은 각 위치 i에서 독립적으로 실행할 수 있기 때문에 쌍합도 다음과 같이 각 위치 i에서 문자들의 분포만으로 계산된 부분합을 모든 위치 i에 대하여 더한 값으로 계산할 수 있다.

보조 정리 1.
임의의 배치 ρ에 대하여, 모든 i $(0 \le i \le n-1)$ 및 $\sigma \in \Sigma$에 대한 계수 배열 $C_\rho[i, \sigma]$가 주어질 때, 쌍합은 $\mathrm{SP}_S(\rho) = \Sigma_{0 \le i \le n-1}\Sigma_{\sigma \in \Sigma}\{C_\rho[i, \sigma]\times(n - C_\rho[i, \sigma])\}/2$ 이다.

다음은 **보조정리 1**로부터 모든 위치에서 변화된 문자들 간의 계수 배열을 이용하여 부분합 변화량을 구하고 이를 합하여 인접 배치에 대한 쌍합을 구한다.

보조 정리 2.
임의의 배치 ρ에 대하여, 모든 i $(0 \le i \le n-1)$ 및 $\sigma \in \Sigma$에 대한 계수 배열 $C_\rho[i, \sigma]$가 주어질 때, 임의의 인덱스 k $(1<k\le K)$ 및 상수 c $(0 < c \le n-1)$를 선택하여 ρ에서 r_k를 $(r_k+c) \bmod n$으로 바꾼 인접 배치 ρ'가 주어질 때, 쌍합은 $\mathrm{SP}_S(\rho') = \mathrm{SP}_S(\rho) + \Sigma_{0 \le i \le n-1}(C_\rho[i, \sigma_1] - C_\rho[i, \sigma_2])$ 이다. 이 때, 각 위치 i에 대해 $\sigma_1 = S_k(r_k)[i]$ 및 $\sigma_2 = S_k((r_k+c) \bmod n)[i]$이다.

알고리즘B는 먼저 초기 배치 $\rho=(0,0,...,0)$에 대하여 계수 배열과 쌍합 $\mathrm{SP}_S(\rho)$를 계산하고, 초기배치 ρ로부터 인접 배치 ρ'를 재귀적으로 계산하여 전체 n^{K-1}가지 배치를 모두 고려하면서 각각의 배치에 대해 **보조정리 2**를 이용하여 $\mathrm{SP}_S(\rho')$를 계산한다. 이렇게 모든 배치에 대하여 쌍합을 구하는 알고리즘의 전체 시간 복잡도는 $O((K+|\Sigma|)n + n^K)$ 이다. **알고리즘B**에서는 최대 K단계의 재귀 호출 동안 각 단계의 계수 배열을 저장해야 하므로 공간 복잡도는 $O(K|\Sigma|n)$이다.

3. 결 론

본 논문에서는 환형문자열에 대하여 해밍거리를 이용한 쌍합 목적함수를 최소화하는 다중서열배치 문제를 정의하였다. 이를 위해 이산 합성곱을 이용한 $O(K^2|\Sigma|n\log n + Kn^{K-1})$ 시간 및 $O(K^2n)$ 공간 알고리즘과 인접 배치 관계를 이용한 $O((K+|\Sigma|)n + n^K)$ 시간 및 $O(K|\Sigma|n)$ 공간 알고리즘을 제안하였다.

4. 참조문헌

[1] Fernandes, F., Pereira, L., Freitas, A.T.: CSA: an efficient algorithm to improve circular DNA multiple alignment. BMC Bioinformatics **10** 230 (2009)

[2] Mosig, A., Hofacker, I., Stadler, P.: Comparative analysis of cyclic sequences: Viroids and other small circular RNAs. Lecture Notes in Informatics **P-83** 93-102 (2006)

[3] Thompson, J.D., Higgins, D.G., Gibson, T.J.: Clustal-W - Improving the Sensitivity of Progressive Multiple Sequence Alignment through Sequence Weighting, Position-Specific Gap Penalties and Weight Matrix Choice. Nucleic Acids Research **22** 4673-4680 (1994)