DOI: 10.3745/KIPSTB.2009.16B.5.359

단백질 3차원 구조의 지역적 유사성을 이용한 Flexible 단백질 구조 정렬에 관한 연구

박 찬 용[†] · 황 치 정[††]

요 약

구조적 생물 정보학 분야는 단백질의 3차원 구조를 대상으로 단백질을 연구하는 분야이며, 본 논문에서는 구조적 생물 정보학 분야의 핵심 연구 주제중의 하나인 Flexible 단백질 구조 정렬에 관한 새로운 알고리즘을 제시한다. Flexible 단백질 구조 정렬을 위하여, 단백질의 3차원 구조의 지역적인 유사성을 이용하여 두 단백질의 유사한 부분 구조를 추출해 내고, 이 추출된 유사 구조간에 연결 가능성을 검색하여 정렬이 가능한 모든 유사 구조를 찾고, 이 유사 구조에 꺾임점을 도입하여 Flexible 단백질 구조 정렬을 수행하였다. 이 과정에서 단백질의 지역적 유사성을 정확히 비교하기 위하여 RDA를 이용한 방법을 제안하였고, Flexible 단백질 구조 정렬시 신뢰성 있는 꺾임점 위치 선정 방법과 그래프를 이용한 최적화 방법을 제안하였다. 성능 평가를 위하여 다양한 방법으로 Flexible 단백질 구조 정렬의 성능 평가를 수행하였고, 기존의 방법인 DALI, CE, FATCAT 보다 성능의 우수함을 나타내었다.

키워드 : 단백질 구조 정렬, 바이오인포매틱스, RDA, 그래프최적화

A Study of Flexible Protein Structure Alignment Using Three Dimensional Local Similarities

Chan-Yong Park[†] · Chi-Jung Hwang[††]

ABSTRACT

Analysis of 3-dimensional (3D) protein structure plays an important role of structural bioinformatics. The protein structure alignment is the main subjects of the structural bioinformatics and the most fundamental problem. Protein Structures are flexible and undergo structural changes as part of their function, and most existing protein structure comparison methods treat them as rigid bodies, which may lead to incorrect alignment. We present a new method that carries out the flexible structure alignment by means of finding *SSPs*(*Similar Substructure Pairs*) and flexible points of the protein. In order to find SSPs, we encode the coordinates of atoms in the backbone of protein into *RDA*(*Relative Direction Angle*) using local similarity of protein structure. We connect the SSPs with Floyd-Warshall algorithm and make compatible SSPs. We compare the two compatible SSPs and find optimal flexible point in the protein. On our well defined performance experiment, 68 benchmark data set is used and our method is better than three widely used methods (DALI, CE, FATCAT) in terms of alignment accuracy.

Keywords : Protein Structure Alignment, Bioinformatics, RDA, Graph Optimization

1. 서 론

단백질은 생체 내에서 생명현상 유지를 위한 다양한 기능과 역할을 한다[1]. 대표적인 기능으로 생체내의 화학반응을 촉매 하는 여러 가지 효소, 정보전달에 관여하는 세포막 수용체, 생체를 방어하는 항체, 근육 수축, 이완 단백질, 혈액 응고인자, 산소, 탄소가스의 운반체, 영양소의 운반체, 뼈와 연골 등을 구성하는 콜라겐 등이다. 단백질들은 생체 내에서 공유 결합이나, 수소 결합 등 자연계에 존재하는 힘에 의해 특정한 형태의 3차원 구조를 형성하고 있다. 단백질의 3차원 구조는 단백질의 기능과 밀접한 관련이 있기 때문에, 단백질의 기능을 규명하기 위하여 단백질의 3차원 구조 연구는 필수적이다.

최근 구조가 밝혀진 단백질의 개수가 급속하게 증가하고, 단백질-단백질과의 상호작용에 관련된 연구가 더욱 활발해짐에 따라, 단백질의 기능을 효과적으로 분석하기 위한 단백질 구조 비교 연구의 필요성이 더욱 증가하게 되었다[2].

단백질 구조 비교는 두 단백질이 구조적으로 얼만큼 유사

† 정 회 원 : 한국전자통신연구원 라이프인포매틱스팀 선임연구원
†† 종신회원 : 충남대학교 전기정보통신공학부 교수
논문접수 : 2009년 1월 22일
수 정 일 : 1차 2009년 7월 10일, 2차 2009년 7월 17일
심사완료 : 2009년 7월 17일

한지를 판단하는 것으로, 일반적으로 단백질 구조 비교는 단백질 구조 정렬(Structure alignment)의 방법으로 수행된다. 단백질 구조 정렬의 정의는 두 단백질에서 가장 큰 공통부분 구조를 구성하는 잔기(Residue)쌍을 찾는 것이다. 즉, 단백질 구조 정렬은 두 개의 단백질 구조 A 와 B 가 주어졌을 때, 두 단백질의 정렬쌍인 $\{A_1, A_2, \cdots, A_i, A_{i+1}, \cdots, A_N\}$, 과 $\{B_1, B_2, \cdots, B_j, B_{j+1}, \cdots, B_N\}$를 찾는 것이다. 이 두 정렬쌍 들은 단백질 구조 정렬 알고리즘에 의해 구해진 구조적으로 가장 유사한 최적의 집합을 구성해야 한다. 구조가 정렬되었을 때, A_i 와 B_j 는 서로 1:1 로 대응되는 잔기들이고, 잔기의 개수는 두 단백질에서 동일한 N 개이다($1 \le i \le N$, $1 \le j \le N$). 대응되는 정렬쌍을 구하였을 때, 정렬쌍 간에 최대로 겹쳐질 수 있는 3차원 변환은 최소 제곱법(Least square method)방법으로 $O(N)$ 시간에 구할 수 있다[3]. 그러므로, 단백질 구조 정렬의 핵심적인 문제는 최적의 유사도로 정렬될 두 단백질의 대응되는 잔기쌍을 찾는 방법이다. 이 방법는 일반적으로 NP-complete 문제[4]로 알려져 있어, 많은 휴리스틱(Heuristic) 방법이 제안되어 왔다.

단백질 구조 정렬을 위하여 DALI(Distance alignment matrix)[5], CE(Combinatorial Extension)[6], VAST(Vector Alignment Search Tool)[7], 3dSearch[8] 등과 같은 많은 단백질 구조 정렬 방법들이 제안되어 왔다. 지금까지 제안된 많은 방법들은 단백질을 강체(Rigid body)로 가정하고 단백질 구조 정렬 알고리즘을 수행하였다. 그러나, 일반적으로 단백질은 생체 내에서 3차원 구조가 고정된 강체라기보다 꺽임이 가능한 부분과 강체인 부분이 혼용된 상태라고 알려져 있다[9]. 그러므로, 이러한 꺽임이 있을 수 있는 부분에서 단백질의 3차원 구조가 변형이 발생하였을 경우, 단백질을 강체로 가정한 알고리즘으로는 두 단백질의 유사도를 정확하게 측정할 수 없다. 그러므로, 단백질 구조 정렬을 보다 신뢰성 있게 측정하기 위해서는, 생체 내에서 단백질의 상태에 적응력이 있는 꺽임을 허용하는(flexible) 단백질 구조 정렬 방법이 필요하다. 최근까지 flexible 단백질 구조 정렬방법은 FlexProt[10], FATCAT[11] 등 상대적으로 소수의 알고리즘만이 제안되었다.

본 논문에서는 꺽임이 허용되는 단백질 구조 정렬 방법을 제안한다. Flexible 단백질 구조 정렬 방법은 단백질의 3차원 구조의 지역적인 유사성을 이용하여 두 단백질의 유사한 부분 구조를 추출해 내고, 이 추출된 유사 구조들의 연결 가능함을 검색하여 정렬이 가능한 모든 유사 구조를 찾는다. 정렬이 가능한 모든 유사 구조를 그래프(Graph)로 맵핑하고, Floyd-Warshall 알고리즘을 적용하여 최종적인 flexible 단백질 구조 정렬을 수행한다.

본 논문의 구성은 다음과 같다. 2장 관련연구에서는 단백질 구조 정렬에 관한 기존의 방법을 설명한다. 3장에서는 flexible 단백질 구조 정렬에 관한 새로운 방법을 제안하며, 4장 실험 및 성능 평가에서는 flexible 단백질 구조 정렬의 성능 평가를 수행하고, 마지막으로 5장에서는 본 논문의 결론과 향후 연구를 기술한다.

2. 관련연구

DALI[5]는 가장 널리 알려진 단백질 3차원 구조 정렬 방법이고, 꺽임 없는 단백질 구조 정렬 방법 중 상당히 정확한 결과를 보여주고 있다[12]. DALI 는 3차원 단백질 구조를 2차원 거리행렬(Distance matrix) 로 표현하고, 이 거리행렬을 6x6 크기의 겹쳐지는 submatrix로 분리한다. 이 submtrix를 기반으로 유사도 값을 계산하기 위하여 몬테카를로 최적화(Monte-carlo optimization)를 사용하여 최종 정렬을 수행한다. 수학적 최적화 방법인 몬테카를로 최적화는 느리고, 때때로 전역 최소점(Global minimum) 을 찾지 못하는 경우가 발생하여 정렬에 실패하게 된다.

CE[6] 는 두 단백질에서 지역적으로 유사한 구조를 가진 8개의 잔기로 구성된 AFP(Alignment Fragment Pair)를 구한 후, AFP와 최적의 다른 AFP를 연결하여 정렬을 수행한다. 이 과정이 전체 AFP에 대해 수행되어 최적 정렬을 구성한다. 이 방법은 최적화 방법을 사용하지 않고, 모든 경우의 수를 전부 비교하는 combinatorial 방법을 사용함으로써 정렬시간이 매우 많이 소요되며, 초기 AFP의 추출이 정렬 결과에 큰 영향을 미친다.

FATCAT[11]은 가장 최근에 발표된 flexible 단백질 구조 정렬 방법으로 FlexProt보다 성능이 좋다[13]. 이 방법은 꺽임에 관련된 내용을 제외하면 CE의 방법과 거의 유사하다. 정렬 방법은 CE에서 제안한 두 단백질에서 지역적으로 유사한 구조를 가진 AFP를 구한 후, 두 AFP의 연결 가능 점수를 모든 AFP 에 대해서 계산한다. 계산된 AFP와 AFP의 연결 가능 점수에 동적 프로그래밍 알고리즘을 적용하여 꺽임이 가능한 단백질 구조 정렬을 수행한다. FlexProt과 같이 동적 프로그래밍 알고리즘의 재귀식에 꺽임 요소에 대한 가중치가 포함되어 있다. 이 방법은 동적 프로그래밍의 재귀식이 FlexProt 보다 더욱 정교하게 계산되어 FlexProt보다 좋은 성능을 가지고 있지만, 동적 프로그래밍의 재귀식에 꺽임 요소를 포함하여 발생하는 문제를 완전히 해결하지는 못하였다.

3. 제안하는 Flexible 단백질 구조 정렬

이 장에서는 제안하는 단백질 구조 정렬 방법의 전체적인 개요를 설명하고, 제안하는 단백질 구조 정렬 방법을 자세하게 기술한다. 제안하는 단백질 구조 정렬 방법은 단백질 구조를 강체가 아닌 꺽임이 허용되는 구조라고 가정하고 정렬을 수행하는 *flexible 단백질 구조 정렬(Flexible protein structure alignment)*이다.

3.1 알고리즘의 개요

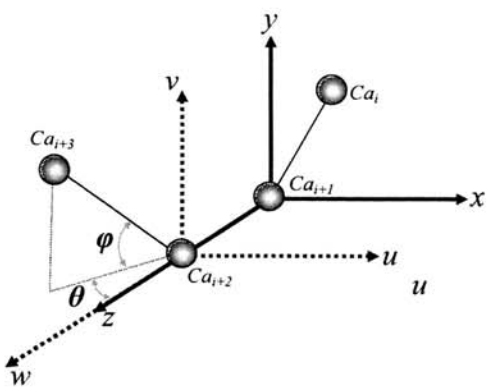
알고리즘의 입력은 단백질 A 와 단백질 B를 구성하는 원자들의 3차원 좌표이다. Flexible 단백질 구조 정렬 알고리즘은 6단계로 구성되어 있다. 단계 1 (3.2 절)과 단계 2

(3.3 절)에서 단백질의 상대적 방향각을 이용하여 지역적으로 유사한 구조를 검색한 후, 단계 3 (3.4 절)에서 유사한 구조들을 연결하여 단백질 구조 정렬을 수행한다. 단계 4 (3.5 절)와 단계 5 (3.6 절)에서 정렬된 단백질을 대상으로 꺽임점을 도입하여 flexible 단백질 구조 정렬을 수행한다. 알고리즘의 출력은 두 단백질의 flexible 단백질 구조 정렬쌍과 꺽임점의 위치이다.

3.2 RDA 코딩

단백질 구조 정렬의 첫 단계로써 두 단백질의 유사한 부분 구조를 찾기 위하여 *RDA(Relative Direction Angle)* 코딩을 수행한다. 단백질의 C_a 원자로 구성된 단백질 백본(Backbone)은 순서가 있는 3차원 좌표를 가지고 있는 하나의 실과 같은 구조를 가지고 있다. 이러한 단백질 백본의 3차원적 형태를 가지고 있는 좌표 열을 그대로 이용하여 두 단백질의 유사한 부분 구조를 검색하는 것은 복잡한 과정을 수행하여야 한다. 그러므로, 단백질 백본의 구조적 특징인 백본을 구성하는 C_a 원자간의 거리는 거의 일정하게 구성되어 있고, 백본이 순차적인 C_a 원자로 구성되어 있음을 이용하여, 3차원 좌표 열을 회전과 이동에 불변한 2차원의 값들로 변환하여 두 단백질의 유사 부분 구조를 검색한다. 먼저, 연속적인 4개의 C_a 원자에 대해 처음 3개의 C_a 원자로 새로운 직교 좌표계(Cartesian coordinate)를 구성한 후, 4번째 C_a 원자를 새로운 직교 좌표계로 변환한다. 변환된 4번째 C_a 원자의 좌표를 구형 좌표계(Spherical coordinate)로 구한다. 새로 구해진 구형좌표계는 (r, θ, ϕ) 로 구성되어 있으나, 단백질의 C_a 간의 거리가 일정함으로 r 값은 상수로 처리되어 (θ, ϕ) 만을 사용한다. 이런 과정으로, 단백질 백본의 3차원 (x,y,z) 좌표는 2차원 (θ,ϕ) 좌표계 변환이 된다. n 개의 3차원 좌표를 가지는 C_a 원자들 $\{(x_1,y_1,z_1), (x_2,y_2,z_2), \cdots (x_n,y_n,z_n)\}$에 대해서 RDA 코딩을 수행하게 되면 m 개의 2차원 좌표열 $\{(\theta_1,\phi_1), (\theta_2,\phi_2), \cdots, (\theta_m,\phi_m)\}$ 이 구해진다.

RDA 코드를 구하는 상세한 방법은 다음과 같다. 단백질에서 연속된 네 개의 Ca 원자(Ca_i, Ca_{i+1}, Ca_{i+2}, Ca_{i+3}) 가 있을 때, (그림 1)과 같이 (Ca_i, Ca_{i+1}, Ca_{i+2})를 이용하여 새로운 좌표계(u,v,w) 를 생성한다. 새로운 직교 좌표계는 업벡터(Up-vector)로 (Ca_{i+1} , Ca_i)를 가지고, 방향벡터(Direction-vector)로 (Ca_{i+1}, Ca_{i+2}) 를 가진다.



(그림 1) 4개의 원자를 이용한 RDA 코딩

새로운 좌표계로 변환하기 위한 동차 변환 행렬(*Homogeneous Coordinate* Transform matrix) T는 식 (1)과 같다.

$$T = \begin{Bmatrix} R_{11} & R_{12} & R_{13} & 0 \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ T_1 & T_2 & T_3 & 1 \end{Bmatrix} \tag{1}$$

방향벡터 $Dir(R_{31}, R_{32}, R_{33})$은 식 (2)와 같다.

$$R_{31} = \frac{x_3 - x_2}{\|v\|}, \quad R_{32} = \frac{y_3 - y_2}{\|v\|}, \quad R_{33} = \frac{z_3 - z_2}{\|v\|}, \quad \|v\| = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2} \tag{2}$$

업벡터 $Up(R_{21}, R_{22}, R_{23})$은 식 (3)과 같다.

$$Up = Up_w - (Up_w \cdot Dir) \cdot Dir \qquad Up_w = (x_1 - x_2,\ y_1 - y_2,\ z_1 - z_2) \tag{3}$$

Right벡터 $R(R_{11}, R_{12}, R_{13})$는 식 (4)와 같이 업벡터와 방향벡터를 외적(Cross product)하여 구한다.

$$R = Up \times Dir \tag{4}$$

이동벡터(Translation vector) $T(T_1,T_2,T_3)$는 Ca_{i+2}가 새로운 좌표계의 원점이 됨으로 식 (5)와 같다.

$$T = (-x3,\ -y3,\ -z3) \tag{5}$$

이렇게 구한 동차 변환 행렬 T를 Ca_{i+3}에 적용하여 Ca'_{i+3} (x_t, y_t, z_t)를 계산한 후, (x_t, y_t, z_t)를 직교 좌표계인 (u, v, w) 좌표계에서 구형 좌표계로 변환한다. 변환식은 식 (6)과 같다.

$$r = \sqrt{x_t^2 + y_t^2 + z_t^2}$$
$$\theta = \tan^{-1}\left(\frac{y_t}{x_t}\right), \qquad (-\pi < \theta < \pi)$$
$$\varphi = \cos^{-1}\left(\frac{z_t}{r}\right) = \cos^{-1}(z_t) \quad (-\pi < \varphi < \pi) \tag{6}$$

이 식에서 r은 단백질 백본은 구성하는 Ca_i와 Ca_{i+1} 간의 거리를 나타낸다. 단백질 백본을 구성하는 Ca 원자 간의 거리는 단백질 화학적 구조상 거의 일정(3.8 Å)[14]하기 때문에, r을 상수로 가정하고 RDA 코딩에 사용하지 않았다.

RDA 코딩을 단백질 A와 단백질 B의 모든 C_a에 대해 4개의 원자단위로 겹치게 적용하여, 단백질 A의 RDA 코드인 RDA_A 와 단백질 B의 RDA 코드인 RDA_B를 구한다.

$$RDA_A = \{ (\Theta_{A1},\phi_{A1}), (\Theta_{A2},\phi_{A2}), \cdots , (\Theta_{Ak},\phi_{Ak}) \}$$
$$RDA_B = \{ (\Theta_{B1},\phi_{B1}), (\Theta_{B2},\phi_{B2}), \cdots , (\Theta_{Bl},\phi_{Bl}) \}$$
$$k = |A| - 3, \quad l = |B|-3$$

3.3 SSP(Similar Substructure Pair) 생성

RDA는 단백질 백본을 구성하는 연속적인 *Ca* 원자들의 상대적인 방향각을 나타내기 때문에, RDA 값이 유사하다는 의미는 단백질의 3차원 구조가 유사하다는 의미이다. 이 단계에서는 두 단백질에서 구한 RDA_A와 RDA_B의 원소를 비교하여, 두 단백질간의 지역적 유사성을 가진 *SSP*를 생성한다.

RDA를 이용하여 두 단백질의 지역적 유사성을 검색하기 위하여, *유사성 맵(Similarity map)*을 구한 후, 연결성 정보를 이용한다. 유사성 맵은 두 단백질이 지역적으로 어느 부분이 유사한가를 나타내는 맵이다. 유사성 맵의 하나의 항목값 $S(i, j)$는 단백질 A의 i번째 RDA 코드와 단백질 B의 j번째 RDA 코드의 유사성을 이진수로 나타낸 값이다. $S(i, j)$를 구하는 식은 식 (7)과 같다. $D(i, j)$는 단백질 A의 i 번째 RDA 코드 (Θ_{Ai}, ϕ_{Ai}) 와 단백질 B의 j 번째 RDA 코드 (Θ_{Bj}, ϕ_{Bj}) 간의 유클리드 거리(Euclidian distance)를 나타낸다. i와 j에 대해서 $D(i, j)$를 구한 후에, $D(i, j)$가 특정 임계 값 (T_d) 이하인 경우에 $S(i, j)$를 1로 설정 한다. 임계 값 T_d는 10으로 설정하여 실험하였다.

$$S(i,\ j) = \begin{cases} 1, & \text{if } D(i,\ j) < T_d \\ 0, & \text{otherwise} \end{cases} \qquad (7)$$
$$D(i,\ j) = \sqrt{(\varphi_i - \varphi_j)^2 + (\theta_1 - \theta_j)^2}$$

다음 단계로 유사성 맵에서 SSP를 구하는 것이다. 하나의 SSP는 유사성 맵에서 왼쪽 상단에서 우측 하단 방향으로의 하나의 사선(Diagonal line)으로 나타난다. 그러므로, 모든 SSP를 찾기 위하여 유사성 맵 전체를 검색하여, 모든 사선을 찾는다. 하나의 SSP는 다음과 같이 정의된다.

$$SSP\ (i1,\ i2,\ j1,\ j2) = \{ \{(\Theta_{i1\cdot0},\ \phi_{i1\cdot0}\}, \{\Theta_{i1\cdot1},\ \phi_{\ i1\cdot1}\}, \cdots \{\Theta_{i2},\ \phi_{\ i2}\}\}, \{\{\Theta_{j1\cdot0},\ \phi_{\ j1\cdot0}\}, \{\Theta_{j1\cdot1},\ \phi_{\ j1\cdot1}\}, \cdots \{\Theta_{j2},\ \phi_{\ j2}\}\} \}$$

여기에서 *i1, i2*는 단백질 A의 RDA 코드의 시작과 끝 인덱스, *j1, j2*는 단백질 B 의 RDA 코드의 시작과 끝 인덱스이다.

3.4 CSSP(Compatible SSP) 생성

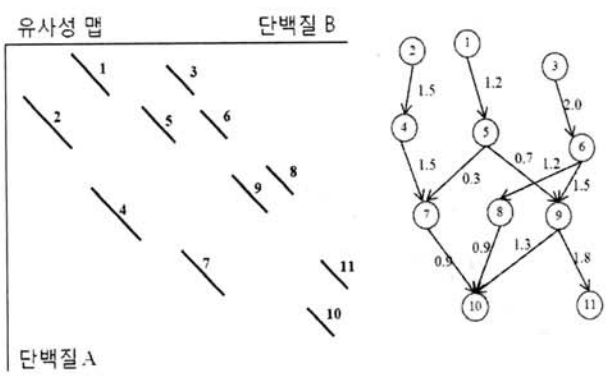
SSP는 두 단백질의 지역적 정보만을 가지고 유사성을 구한 것이므로, 하나의 단백질 구조 정렬에 많은 SSP가 포함될 수 있다. 하나의 구조 정렬에 포함된 하나 이상의 SSP를 *CSSP(Compatible SSP)*로 정의한다. 모든 CSSP를 구하기 위하여, 하나의 SSP를 그래프의 노드로 가정하고, 노드 간의 연결 강도(Weight)를 두 SSP의 연결 가능 스코어로 계산하여 DAG(Directed Acyclic Graph) 그래프를 생성한다. 이 DAG 그래프를 이용하여 하나의 정렬을 구성하는 최적의 SSP들을 찾는 문제를 그래프에서 최단 경로 검색 문제(Shortest path problem)로 전환하였다.

(그림 2)에서 SSP와 노드와의 관계를 보여준다. 유사성 맵에서 하나의 SSP는 우측 DAG 그래프에서 하나의 노드로 할당이 된다. 각 우측 노드 간의 연결강도는 좌측의 SSP들 간의 연결 가능 스코어로 계산된다. $SSP(m)$와 $SSP(n)$가 있을 때 두 SSP 간의 연결 가능 스코어 $Score(m,n)$는 식 (8)과 같이 계산된다. 연결 가능 스코어는 두 SSP가 잘 연결될수록 작은 값을 가지고, 연결 불가능일 경우에는 ∞ 값을 가진다. 두 SSP가 연결 불가능일 경우에는 그래프의 두 노드 간에 연결을 만들지 않는다.

$$\mathrm{Score}(\mathrm{m},\ \mathrm{n}) = \begin{cases} \mathrm{C}(\mathrm{m},\ \mathrm{n}), & \mathrm{if SD}(\mathrm{m},\ \mathrm{n}) < T_{\mathrm{sd}} \\ \infty, & \text{otherwise} \end{cases}$$
$$\mathrm{C}(\mathrm{m},\ \mathrm{n}) = ((RMSD(m,\ n)*5.0) + \mathrm{gap1} + \mathrm{gap2}) * (\mathrm{gap1} + \mathrm{gap2})/matchLen \qquad (8)$$

$RMSD(m,n)$은 $SSP(m)$와 $SSP(n)$을 정렬하였을 때의 *RMSD(Root Mean Square Deviation)* 값이고, *gap1*은 단백질 A에서 $SSP(m)$과 $SSP(n)$과의 간격이다. *gap2* 는 단백질 B에서 $SSP(m)$와 $SSP(n)$과의 간격이다. *matchLen*은 $SSP(m)$의 정렬쌍의 길이와 $SSP(n)$의 정렬쌍의 길이의 합이다. 만약 $RMSD(m,n)$가 임계치 T_{rmsd} 보다 작을 경우에는 $C(m,n)$으로 두 SSP의 연결 가능 스코어를 계산하고, 임계치 T_{rmsd} 보다 클 경우에는 ∞ 값을 가진다. 임계치 T_{rmsd}는 실험적으로 5.0Å 을 사용하였다.

DAG 그래프에서 최단 경로 검색 알고리즘에는 일반적으로 탐욕적 방법(Greedy method)를 이용하는 다익스트라(dijkstra) 알고리즘[15]과 동적 프로그래밍 방법(Dynamic programming method)을 이용하는 플로이드(Floyd-Warshall) 알고리즘[16]이 있다. 다익스트라 알고리즘은 시작 노드와 종점 노드가 결정되어 있는 경우 두 노드간 의 최단 경로를 구하는 알고리즘이다. 플로이드 알고리즘은 DAG 그래프에서 모든 최단 경로 검색을 하기 위하여 특정한 시작 노드와 종점 노드를 정의하지 않고, N개의 노드로부터 다른 (N-1)개의 노드까지 $N(N$-1)개의 최단경로를 구하는 알고리즘이다. DAG 그래프의 시작점과 끝점을 정의 하지 않고 모든 최단 경로를



(그림 2) CSSP를 구하기 위한 DAG 그래프

찾기 위하여 Floyd-Warshall 알고리즘을 적용하고, 찾아진 최단경로는 하나 이상의 노드로 구성된 CSSP로 생성된다.

3.5 Flexible 단백질 구조 정렬 후보 생성

정렬된 CSSP들에 꺽임점을 도입하여 flexible 단백질 구조 정렬 후보를 생성한다. Flexible 단백질 구조 정렬 후보 생성 단계는 3.4절의 방법과 유사하다. 그러나, 3.4절의 방법과 차이점은 CSSP 간의 연결가능성을 계산할 때, 꺽임이 적용됨으로 두 CSSP의 연결시 정렬가능성을 고려하지 않는다.

CSSP(m)과 CSSP(n)의 연결 가능 스코어는 식 (9)와 같이 계산되어 그래프의 노드의 연결강도로 사용된다.

$$Score(m, n) = (gap\_cssp1 + gap\_cssp2) / matchLen \quad (9)$$

Score 계산시 두 정렬된 CSSP의 RMSD 값은 고려하지 않는다. 왜냐하면, 각각의 CSSP는 이미 정렬이 수행되어 상당히 작은 RMSD 값을 가지고 있고, 꺽임이 허용되어 정렬될 경우에는 두 정렬된 CSSP를 연결하였을 때, 전체 RMSD 값이 커지지 않기 때문이다.

3.6 최종 Flexible 단백질 구조 정렬

Flexible 단백질 구조 정렬 후보들에 대해 정렬의 우수성을 비교할 수 있는 최종 정렬 스코어를 이용하여 최종적인 flexible 단백질 구조 정렬을 찾는다. 사용하는 최종 정렬 스코어는 식 (10)과 같고, 정렬이 우수할수록 높은 값을 가진다. #AlignedResidue는 flexible 단백질 구조 정렬 후보의 정렬쌍의 개수이고, |A|는 단백질 A 의 잔기 개수, |B|는 단백질 B의 잔기 개수다. *RMSD*는 flexible 단백질 구조 정렬 후보의 *RMSD* 값이고, *flex*는 꺽임점의 개수이다. 동일한 조건에서 꺽임점의 개수가 적을 경우에 더 큰 값을 가진다(좋은 정렬 결과로 계산된다). *flex*에 0.1을 곱한 것은 실험적으로 구해진 수치이다.

모든 flexible 단백질 구조 정렬 후보들에 대하여 최종 정렬 스코어를 계산하여 가장 큰 값을 가지는 flexible 단백질 구조 정렬 후보를 최종 결과로 사용한다.

$$\text{Final Score} = \sqrt{\frac{\#\text{AlignedResidue}}{\min(|A|,|B|)\times(1+RMSD+flex\times 0.1)}} \quad (10)$$

4. Flexible 단백질 구조 정렬의 실험 및 성능평가

단백질 구조비교를 위한 유사도를 측정하는 방법은 매우 많이 존재한다. 가장 일반적인 단위가 *RMSD*(*root mean square deviation*)이다[17]. 이 방법은 단백질 A 가 단백질 B 로 가장 일치되게 겹쳐질 때, 대응되는 정렬쌍을 가지고 RMSD 값을 계산한다. 작은 *RMSD* 값이 항상 최적 정렬을 나타내지는 않기 때문에, 대부분의 경우 *RMSD*만을 이용하여 3차원 단백질 구조 정렬의 성능을 측정하지는 않는다. 기존의 연구에서는 좀더 의미 있는 정렬결과를 제시하기 위하여 *RMSD* 값과 대응되는 정렬쌍의 개수를 다양한 방법으로 조합하여 하나의 수치로 유사성 결과를 나타낸다[18-20]. 그러나 현재까지 어떤 하나의 값으로 단백질 구조 정렬의 성능을 나타내는 일반화된 방법은 존재하지 않으므로, 본 논문에서는 기존 연구에서 사용되었던 4개의 단백질 구조 정렬 성능 측정법을 사용한다.

(1) *alignment score* (*S*) [18]

$$S = \frac{3\times N}{1+SD} \quad (11)$$

N은 정렬이 되었을 때, 대응되는 정렬쌍의 개수이다(N=$|A_{al}|$=$|B_{al}|$).

(2) *similarity index*(*SI*)[19]

$$SI = \frac{SD\times\min(|A|,|B|)}{N} \quad (12)$$

|A|는 단백질 A의 전체 잔기의 개수이고, |B|는 단백질 B의 전체 잔기 개수다.

(3) *match index*(*MI*)[19]

$$MI = \frac{1+N}{\left(1+\frac{SD}{\omega_n}\right)\times(1+\min(|A|,|B|))} \quad (13)$$

여기에서 ω_n=1.5로 실험을 하였다.

(4) *structural alignment score*(*SAS*)[20]

$$SAS = \frac{RMSD^*100}{N} \quad (14)$$

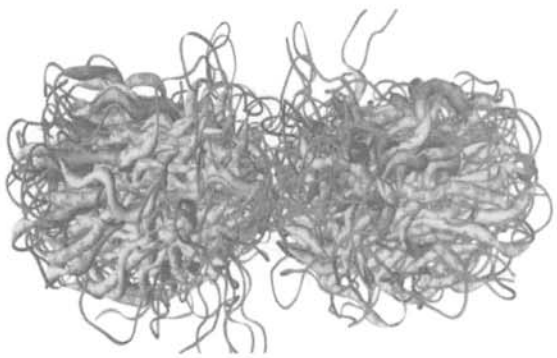
성능 비교를 위한 첫 번째 실험에서는, NMR 모델인 1MFN을 실험 대상 데이터로 정렬 실험을 하였고, 두 번째 실험에서는, 성능 평가 대상 데이터를 [21]에서 제안한 68개를 대상으로 실험을 수행하여 DALI[5], CE[6], FATCAT[11]과 성능 비교를 한다. 실험에 사용된 알고리즘인 DALI와 CE는 꺽임 없는 단백질 구조 정렬 방법으로, 꺽임 없는 단백질 구조 정렬 방법 중 가장 정확하고 가장 널리 사용되는 구조 정렬 방법으로 알려져 있다[22-23]. FATCAT은 flexible 단백질 구조 정렬 방법 중 대표적인 것이다.

4.1 하나의 단백질의 다양한 구조를 대상으로 성능 비교

단백질은 생체 내에서 여러 가지 조건에 의하여 꺽일 수 있는 변형 가능한 구조이다. 생체 내에서 하나의 단백질에

대해 여러 형태의 변형이 된 3차원 구조는 구조 규명 시 NMR 방법을 사용하면, 여러 형태의 3차원 구조를 얻어낼 수 있다. 이 실험에서는 NMR 방식으로 규명된 단백질의 각각의 MODEL에 대해서 꺽임 없는 단백질 구조 정렬인 DALI, CE와 성능 비교를 수행하고, flexible 단백질 구조 정렬인 FATCAT과의 성능 비교를 수행한다.

PDBID 1MFN은 쥐 파이브로넥틴의 세포 부착 단백질(Cell attachment modules of mouse fibronectin)로 하나의 단백질에 20개의 NMR 구조로 된 MODEL이 들어있다. (그림 3)과 같이 각각의 모델들이 같은 단백질이지만, 다양한 형태를 가지고 있다. 실험 방법은 1번째 모델에 대해 2번부터 20번까지의 19개 모델과 구조 정렬을 수행한다. 결과는 <표 1>에 보여진다. DALI 와 CE는 정렬쌍의 개수와 관계없이 평균 RMSD 값이 5.45, 4.23으로 두 단백질이 유사성이 있다고 판단하기 어려운 결과를 보여준다. 제안된 방법과 FATCAT의 꺽임점의 수는 제안된 방법이 약간 많고, 정렬쌍의 개수는 FATCAT이 조금 더 많다. 그러나, 정렬 RMSD는 제안된 방법이 우수함을 보여준다. 또한 제안된 방법의 RMSD 값의 표준편차는 0.14 정도인 것에 비해 FATCAT의 RMSD의 표준편차는 0.70으로 제안된 방법이 더 우수함을 알 수 있다.



(그림 3) PDBID 1MFN

〈표 1〉 PDBID 1MFN 의 구조정렬

	제안방법			DALI	
	#Flex	#Residue	RMSD	#Residue	RMSD
평균	1.21	173.63	1.32	180.42	5.45

	CE		FATCAT		
	#Residue	RMSD	#Flex	#Residue	RMSD
평균	167.57	4.23	0.73	182.57	2.19

4.2 Fisher Data Set을 대상으로 성능비교

두 번째 실험으로, Fisher[21]가 제안한 데이터 셋을 가지고 제안된 방법과 DALI, CE, FATCAT과의 성능비교를 수행한다. 실험에 사용한 데이터 셋은 68개의 PDB쌍 데이터로, 평균 서열 유사성이 18.6%이고, 표준 편차는 4.4%이다. 최소 서열 유사성은 8%이고, 최대 서열 유사성은 31%이다.

이 실험에서 정렬수치는 *S, SI, MI, SAS* 로 계산되어 68개 실험 데이터에 대해서 제안된 알고리즘과 기존의 3가지 방법과의 비교를 한다.

각각의 평가방법에 대해 제안된 방법과 기존의 방법을 비교하여 우수한 것의 개수를 종합한 것은 <표 2>와 같다. 제안방법이 우수한 개수(A)는 68개의 데이터 셋에 대하여 제안된 방법이 우수한 것으로 나온 개수이다. 기존방법이 우수한 개수(B)는 기존방법이 우수한 것으로 나온 개수다. 우수한 것의 비율은 68개의 데이터 중에서 제안방법이 우수한 것의 비율로, 약 89% 에 대해 제안방법이 우수한 것으로 나타났다. 이는 제안한 방법이 DALI나 CE 와 같은 꺽임점이 없는 단백질 구조 정렬을 포함하여, FATCAT 과 같은 Flexible 단백질 구조 정렬보다 우수한 결과를 보인다고 할 수 있다.

5. 결론 및 향후연구

본 연구는 Flexible 단백질 구조 정렬을 위하여, 단백질의 3차원 구조의 지역적인 유사성을 이용하여 두 단백질의 유사한 부분 구조를 추출해 내고, 이 추출된 유사 구조간에 연결 가능성을 검색하여 정렬이 가능한 모든 유사 구조를 찾고, 이 유사 구조에 꺽임점을 도입하여 flexible 단백질 구조 정렬을 수행하였다. 이 과정에서 단백질의 지역적 유사성을 정확히 비교하기 위하여 RDA를 이용한 방법을 제안

〈표 2〉 성능평가 비교표

기존방법	DALI				CE				FATCAT			
평가방법	S	SI	MI	SAS	S	SI	MI	SAS	S	SI	MI	SAS
제안방법이 우수한 개수(A)	63	63	59	63	61	60	60	60	61	65	52	65
기존방법이 우수한 개수(B)	5	5	9	5	7	8	8	8	7	3	16	3
제안방법이 우수한 것의 비율(%) (A/(A+B))*100	92.6	92.6	86.8	92.6	89.7	88.2	88.2	88.2	89.7	95.6	76.5	95.6
평균우수비율	91.17				88.60				89.33			

하였고, flexible 단백질 구조 정렬시 신뢰성 있는 꺾임점 위치 선정 방법과 그래프를 이용한 최적화 방법을 제안하였다. 성능 평가를 위하여 다양한 방법으로 flexible 단백질 구조 정렬의 성능 평가를 수행하였고, 기존의 방법인 DALI, CE, FATCAT 보다 성능의 우수함을 나타내었다.

향후 연구로서, 단백질 구조를 이용한 단백질 분류 시스템에 관한 연구를 수행할 예정이다. 단백질 구조 분류 시스템은 구조가 밝혀진 단백질을 대상으로 유사한 구조를 분류해 내는 시스템이다. 기존의 단백질 분류 시스템인 CATH, SCOP 등은 단백질을 강체로 가정하고 분류한 시스템이므로 flexible 단백질 구조 정렬을 이용한 단백질 구조 분류 시스템은, 기존의 방법보다 단백질의 생물학적 특성이 더욱 잘 반영된 신뢰성 있는 단백질 구조 분류 시스템이 될 것이다.

참 고 문 헌

[1] J. W. Kimball. Biology. Wm. C. Brown Publishers, 6th edition, 1994.

[2] 박찬용, 황치정. "기하인스턴싱 기법을 이용한 단백질 구조 가시 및 속도 향상에 관한 연구," 정보처리논문지, 제16-A권 제3호, pp.153-158, 2009.

[3] K. Arun, T. Huang, and S. Blostein, "Least-squares fitting of two 3-D point sets," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.9, No.5, pp.698-700, 1987.

[4] L. Holm and C. Sander, "3-D lookup: fast protein structure database searches at 90% reliability," In Proceedings of 3rd International Conference on Intelligent Systems for Molecular Biology (ISMB'95), pp.179-187, 1995.

[5] L. Holm and C. Sander, "Protein structure comparison by alignment of distance matrices," Journal of Molecular Biology, Vol.233, pp.123-138, 1993.

[6] I. N. Shindyalov and P. E. Bourne, "Protein structure alignment by incremental combinatorial extension (CE) of the optimal path," Protein Engineering, Vol.11, No.9, pp.739-747, 1998.

[7] J. F. Gibrat, T. Madej, and H. Bryant, "Surprising similarities in structure comparison," Current Opinion in Structural Biology, Vol.6, pp.377-385, 1996.

[8] W. Taylor and C. Orengo, "Protein structure alignment," Journal of Molecular Biology, Vol.208, pp.1-22, 1989.

[9] M. Gerstein, A.M. Lesk and C. Chothia, "Structural mechanisms for domain movements in proteins," Biochemistry Vol.33, pp.6739-6749, 1994.

[10] M. Shatsky, H.J. Wolfson, and R. Nussinov, "Flexible protein alignment and hinge detection," Proteins: Structure, Function, and Genetics Vol.48, pp.242-256, 2002.

[11] Ye Yuzhen and Adam Godzik, "Flexible structure alignment by chaining aligned fragment pairs allowing twists," Bioinformatics Vol.19, suppl.2, pp.246-255, 2003.

[12] M. L. Sierk and W. R. Pearson, "Sensitivity and selectivity in protein structure comparison," Protein Science, 13, pp. 773-785, 2004.

[13] Y. Yuzhen and A. Godzik, "Database searching by flexible protein structure alignment," Protein Science, Vol.13, No.7, pp.1841-1850, 2004.

[14] T. Holton, T. R. Ioerger, J. A. Christopher and J. C. Sacchettini, "Determining protein structure from electron-density maps using pattern matching," Acta Cryst. D56, pp.722-734, 2000.

[15] E. W. Dijkstra, "A note on two problems in connection with graphs," Numerische Math. Vol.1, pp.269-271, 1959.

[16] R. Floyd, "Algorithm 97 (shortest path)," Commun. ACM, Vol.5, No.6, pp.345, 1962.

[17] P. Koehl, "Protein structure similarities," Current Opinion in Structural Biology, Vol.11, pp.348-353, 2001.

[18] N. N. Alexandrov and D. Fischer, "Analysis of topological and nontopological structural similarities in the PDB: new examples with old structures," Proteins: Structure, Function, and Genetics, Vol.25, No.3, pp.354-365, 1996.

[19] G. J. Kleywegt and A. Jones, "Superposition," CCP4/ESF-EACBM Newsletter on Protein Crystallography, Vol.31, pp. 9-14, 1994.

[20] S. Subbiah, D. V. Laurents and M. Levitt, "Structural similarity of DNA-binding domains of bacteriophage repressors and the globin core," Current Biology, Vol.3, pp.141-148, 1993.

[21] D. Fischer, A. Elofsson, D. Rice, and D. Eisenberg, "Assessing the performance of fold recognition methods by means of a comprehensive benchmark," In Proceedings of 1996 Pacific Symposium on Biocomputing (PSB'96), pp.300-318, 1996.

[22] M. Novotny, D. Madsen, and G. J. Kleywegt, "Evaluation of protein fold comparison servers," Proteins: Structure, Function and Bioinformatics, Vol.54, pp.260-270, 2004.

[23] M. L. Sierk and W. R. Pearson, "Sensitivity and selectivity in protein structure comparison," Protein Science, Vol.13, pp. 773-785, 2004.

박 찬 용

e-mail : cypark@etri.re.kr

1994년 광운대학교 컴퓨터공학과 졸업(학사)

1996년 광운대학교 컴퓨터공학과 졸업(석사)

2007년 충남대학교 컴퓨터과학과 졸업(박사)

2008년 미국 UCSB 박사 후 과정

1995년~현 재 한국전자통신연구원 라이프 인포매틱스팀 선임연구원

관심분야 : 컴퓨터 그래픽스, 바이오인포매틱스, 단백질 구조 등

황 치 정

e-mail : cjhwang@cnu.ac.kr

1975년 2월 서강대학교 수학과 졸업

1985년 2월 코네티컷주립대학교 전산학과 (석사)

1987년 2월 코네니컷주립대학교 전산학과 (박사)

1987년 2월 코네티컷주립대학교 객원교수

1999년 2월 충남대학교 전자계산소 소장

1988년 2월~현 재 충남대학교 전기정보통신공학부 교수

관심분야 : 영상처리, 컴퓨터 비젼, 컴퓨터그래픽스