LOCK을 확장한 3차원 단백질 구조비교 및 분석시스템의 설계 및 구현

정 광 수[†] · 한 욱[††] · 박 성 희[†††] · 류 근 호[††††]

요 약

단백질의 구조는 단백질의 기능과 밀접한 연관을 가지고 있으며 단백질 구조비교는 단백질의 모티프와 패밀리를 검정하고 나아가서 그들의 기능을 파악하는데 매우 중요한 역할을 한다. 이 논문에서는 단백질 구조데이터 및 관련된 문헌 데이터의 통합된 데이터베이스를 구축하고 웹 환경에서 질의된 단백질과 유사성 비교를 진행하여 그 결과 및 연관된 문헌데이터를 검색하여 체계적으로 정보를 제공하는 단백질 분석시스템을 제안한다.

제안 시스템을 구축하기 위하여 현재까지 가장 큰 단백질 구조데이터의 저장소인 Protein Data Bank의 플랫파일 데이터에 대해 분석을 진행하고 여기에서 단백질의 구조비교 알고리즘에 필수적인 구조데이터정보를 추출하여 새로운 구조비교에 사용되는 엔드리 플랫 파일을 만들어서 데이터베이스를 구축한다. 이러한 엔트리에 연관된 분석정보 데이터는 데이터베이스 스키마를 작성하여 문헌정보 데이터베이스를 구축한다. 따라서 사용자가 인터넷을 통하여 진행한 질의는 구조비교엔진을 통하여 유사부분과 RMSD값이 계산되고 이와 연관된 문헌정보의 검색이 진행된 후 체계적으로 출력화면에 보여준다. 제안 시스템은 기존의 구조비교시스템보다 빠른 검색을 지원하고 더 훌륭한 분석환경을 제공한다.

Comparison and Analyzing System for Protein Tertiary Structure Database expands LOCK

Kwang Su Jung[†] · Yu Han[††] · Sung Hee Park[†††] · Keun Ho Ryu[††††]

ABSTRACT

Protein structure is highly related to its function and comparing protein structure is very important to identify structural motif, family and their function. In this paper, we construct an integrated database system which has all the protein structure data and their literature. The structure queries from the web interface are compared with the target structures in database, and the results are shown to the user for future analysis.

To constructs this system, we analyze the Flat-File of Protein Data Bank. Then we select the necessary structure data and store as a new formatted data. The literature data related to these structures are stored in a relational database to query the any kinds of data easily. In our structure comparison system, the structure of matched pattern and RMSD valure are calculated, then they are showed to the user with their relational documentation data. This system provides the more quick comparison and nice analyzing environment.

키워드 : 단백질 구조(Protein Structure), 단백질 데이터베이스(Protein Database), 단백질 구조비교(Protein Structure Comparison)

1. 서 론

현재 데이터베이스에서 발견되는 단백질 서열과 구조의 다양성으로 인하여 화학 결합 또는 에너지 계산에 의한 폴리머화학으로는 단백질의 기능을 해석할 수 없다. 그래서 서열과 구조의 비교는 현재 Bioinformatics 분야에서 가장 중요한 분야로 등장하였다[1]. 단백질은 폴딩(Folding)되어 복잡한 3차원 구조를 형성하는데 이러한 데이터는 X-ray crystallography 혹은 2D NMR방법을 이용하여 얻어진다. 이러한 단백질의 구조를 비교하여 정확하게 모티프를 정의하고 패밀리를 분류하는 것은 현재까지 어려운 작업이다. Protein Data Bank[9](Bernstein et al, 1977)는 현재 가장 널리 알려진 단백질 구조데이터의 저장소이다. 현재 PDB에 저장되어 있는 데이터는 20,200 개를 육박하고 있고 매월 거의 150개의 새로운 구조가 추가되고 있다. PDB 구조데이터는 개별적으로 새로운 기능적 정보를 제공하고, 집합적으로는 현재의 단백질 폴드 체제를 제공한다. 이렇게 증가되는 단백질 사이에 존재하는 유사성 관계 알아내는 것은 구조적 모티프나 폴드 패밀리를 정의하기 위해서 필요하다. 그리고 이러한 구조적 레벨에서 단백질을 비교하는 것은 생물체의

† 준 회 원 : 충북대학교 대학원 전자계산학과 박사과정
†† 준 회 원 : 충북대학교 대학원 전자계산학과 석사
††† 준 회 원 : 충북대학교 대학원 전자계산학과 박사수료
†††† 종신회원 : 충북대학교 전기전자컴퓨터공학부 교수
논문접수 : 2003년 11월 7일, 심사완료 : 2004년 7월 27일

안정적이고 기능적인 단백질을 구성하는데 사용된 폴딩 기술을 알아내는데 필수적이다. 단백질 구조는 서열보다 더 특징적이고, 또 기능과 더 큰 연관을 갖고 있기 때문에 단백질과 패밀리 사이의 진화적 간격을 보다 정확히 예측하기 위하여 이러한 단백질사이의 3차원적 연관에 대한 연구가 필요하다.

현재까지 여러 가지 단백질 구조 비교방법[2,4,5,7]들이 구조정렬 알고리즘에 기반하여 개발되었다. 이러한 알고리즘들은 단백질의 문헌정보 데이터와 분리되어 분석되기 때문에 검색을 진행한 후 정보의 편리한 제공을 할수 없고, PDB 플랫파일에 대해 전 처리를 진행하지 않았기 때문에 플랫파일에서 필요정보를 읽는데도 많은 시간을 낭비한다. 그리하여 전체 비교시간이 많이 늘어나게 된다. 이 논문에서는 임의의 PDB 단백질구조를 대상으로 빠른 구조비교를 지원하고 그 결과를 웹을 통하여 전시할 수 있는 통합된 데이터베이스 시스템을 제안한다.

우리는 PDB 플랫파일의 각 엔트리에서 2차구조와 3차구조 데이터만을 파싱하여 새로운 구조비교 알고리즘에 필요한 플랫파일 포맷으로 저장한다. 이 중 2차구조 정보가 들어있지 않은 데이터는 DSSP 알고리즘을 이용하여 2차구조 데이터를 생성하여 해당 엔트리에 추가한다. 이러한 플랫파일 포맷은 구조비교에 필요한 단백질의 2, 3차 구조정보만 들어있기 때문에 데이터 파일을 읽는 속도를 많이 단축할 수 있다. 이 단백질 엔트리들과 연관된 문헌 데이터는 우리가 설계한 데이터베이스 스키마를 적용하여 데이터베이스에 저장된다. 다음으로 사용자가 웹을 통하여 전송한 단백질 구조 혹은 단백질 엔트리 코드는 LOCK 알고리즘을 확장한 제안 구조비교 시스템을 이용하여 구조비교가 진행된다. 질의 단백질과 데이터베이스 내 단백질의 구조적으로 매치되는 부위와 RMSD 값이 단백질의 주석정보와 함께 사용자에게 전시된다. 이러한 시스템은 검색속도가 빠르고 데이터베이스내의 임의의 단백질과 1:1과 1:n 의 구조비교가 가능하며 관련정보를 쉽게 추출할 수 있고 구조와 연관된 단백질의 정보를 손쉽게 찾아낼 수 있는 장점을 가지고 있다. 따라서 이 시스템을 단백질의 분류, 모티프검색 및 새로운 단백질의 기능 예측 등에 활용할 수 있다.

2. 관련연구

이 절에서는 제안 시스템을 구축하기 위하여 단백질 구조, 단백질 데이터베이스, RMSD값의 계산 및 단백질 구조 비교기법에 대해 진행한 관련연구에 대해 설명한다.

2.1 다차원 단백질구조

단백질의 구조[6,7]는 1차, 2차, 3차 및 4차로 정의되는데 그 의미를 보면 다음과 같다.

1차 구조는 단백질의 아미노산서열인데 이는 단백질의 3-차원 구조를 결정하는 가장 큰 요인 중의 하나이다. 일반적으로, 비슷한 1차 구조를 가지고 있는 단백질은 비슷한 폴딩 구조를 가진다. 그러나 서열 유사성의 많고 적음으로 두 단백질이 동일한 구조를 가지고 있다고 표현하기 힘들다. 폴리펩타이드 체인 내부에서 물분자를 밀어내면서 밀집한 구조를 이루고 펩타이드 조각들이 규칙적이고 반복적인 외모를 가지는 것은 인접한 잔기들 사이의 상호작용에 의해서 이루어진다. 이런 지역적인 조합은 짧은(5~20잔기) 조각을 가지는데 이들을 2차구조라고 한다. 일반적인 2차구조는 α-helices, β-strands 및 reverse turn이 있다. 2차구조의 산소와 질소사이의 반복되는 수소본드에 의하여 1차 구조에서 멀리 떨어져 있는 잔기들을 가까운 곳에 끌어온다. 이런 단일 폴리펩타이드 체인 조각을 포함하고 있는 묶어진 배열을 3차구조라고 한다. 일부 단백질은 모여져서 dimmers, trimers 등과 같은 소중합체를 이룬다. 4차 구조는 소중합체 단백질에서의 하위구조 배열을 표현한다. 더 큰 단백질은 도메인이라고 부르는 분리된 구조모듈로서 이루어지는데 그들은 확장된 체인에 의해서 연결된다.

2.2 단백질구조 데이터베이스

생물학적 데이터베이스 중에서 단백질 서열과 구조정보 데이터를 보유하고 있는 데이터베이스는 SWISS-PROT[8], PDB[9], CATH[10], DSSP[11]등이 있다. SWISS-PROT은 단백질 서열, 도메인 구조와 기능 중복을 최소화하고 고 수준의 주석을 제공하는 단백질 서열 데이터베이스로서 SIB (Swiss Institute of Bioinformatics)에 의해 운영되고 있다. 이 데이터베이스에는 십만 개 이상의 단백질 서열 엔트리를 보유하고 있으며 PDB에서 제공되는 각각의 단백질 엔트리 3차 구조에 대한 링크를 제공하고 있다.

PDB(Protein Data Bank)는 이미 해결된 모든 거대분자의 원자좌표정보와 기타 주석정보는 PDB에서 다운로드 받을 수 있다. 그리고 다른 데이터베이스와 유용한 서비스에 대한 많은 링크가 포함되어 있다. 1990년대 중반까지 저장되는 데이터의 증가는 느렸으나 현재는 급속히 증가하고 있다. 데이터의 내역을 보면 2001년 3298개의 구조가 PDB에 저장되었는데 그 중 75%는 RCSB-Routgers의 팀에 의해 처리되었고 10%가 오사카대학, 14%가 유럽 바이오인포매틱스 연구소에 의해 저장되었다. 저장된 구조의 82%는 X-ray crystallo-graphic방법, 15%는 NMR방법으로 확정되었다.

CATH 데이터베이스는 PDB로부터 유도된 2차 구조와 위상에 기반 하여 분류된 데이터베이스이다. 이는 구조적인 레벨에서 한 단백질이 다른 단백질과 얼마나 연관이 있는지를 아는데 필요하다. CATH는 단백질 도메인에 대한 계층적 구조로서 단백질을 4개의 주요 클러스터 - 클래스(C), 구조(Architecture), 위상(Topology), 상동 상위패밀리(Homologous SuperFamily)로 분류하였다.

DSSP프로그램은 단백질 2차구조를 표준화하기 위하여 Wolfgang Kabsch 와 Chris Sander에 의해 개발되었다. DSSP데이터베이스에는 모든 PDB데이터베이스 그리고 그 외 단백질 데이터의 2차 구조 정보를 보유하고 있다. 이러한 2차 구조 정보는 아미노산서열과 단백질 구조사이의 연

관성을 연구하는데 필요하다.

이러한 데이터베이스들은 플랫파일 형식으로 데이터를 보유하고 있으며 PDB형식의 파일은 많은 구조 분석 프로그램에서 사용되고 있다.

2.3 Root Mean Square Distance (RMSD)

점들 사이의 기하학적 유사성을 결정하는 가장 기본적인 방법은 Root Mean Square Distance(RMSD)[12]를 계산하는 것이다. 이 비교는 점들 사이의 1대 1 대응을 필요로 한다. 점 $A=\{a_1, \ldots, a_n\}$ 과 $B=\{b_1, \ldots, b_n\}$사이의 RMSD는 A의 점들과 대응되는 B의 점들사이 거리의 평균을 더한 것의 평균값으로 정의된다. 최소 RMSD를 구하기 위하여 A를 B와 같은 중심을 가지도록 이동한다. 다음 A와 B의 연관 매트릭스를 얻는다. 이는 (p, q)엔트리가 $\sum_{i=1}^{n} a_{i,p} b_{i,q}$ 인 3*3 매트릭스이다. 이러한 매트릭스는 단일 값의 재조합으로 인수화 시킬 수 있다. 얻어진 인수는 A를 B에 가장 가까이 매치 시킬 수 있는 직교 매트릭스와 RMS값을 얻을 수 있게 한다. R^3내에서 두 그룹의 원자들 사이에서 최소 RMSD를 구하는 방법은 비슷한 두 구조를 비교할 때 아주 유용하여 분자구조정렬에 많이 사용되고 있다.

2.4 단백질구조 비교기법

현재까지 DALI[2], VAST[13], TOPS[14], LOCK[4]등 많은 단백질 구조비교 알고리즘이 개발되어왔다. DALI는 가장 오래되고 대중적인 구조인식 방법 중의 하나이다. 웹 기반 서비스 중에서 첫 번째로 선호하는 구조인식 도구이다. DALI는 또한 FSSP 데이터베이스의 구조비교 엔진으로도 사용되고 있다. DALI는 단백질 구조의 모든 Ca-Ca 사이 거리를 표현하는 2차원 거리-매트릭스의 정렬에 기반한 알고리즘이다. 주어진 한 쌍의 구조에 대하여 비슷한 접촉패턴에 대한 최적의 배열을 찾아낸다. 이러한 배열들은 리스트에 저장되고 최종 정렬결과는 이러한 리스트에서 겹치는 부분을 수집하는 것으로 이루어진다. 최적의 정렬을 찾는데 큰 검색공간을 필요로 하기 때문에 DALI는 branch-and-bound 알고리즘을 사용한다. branch-and-bound 알고리즘은 반복적으로 검색공간을 작은 하위 집합을 재 배정하면서 상위범위의 평가함수를 각각 구하는 알고리즘이다.

VAST는 NCBI에서 Entrez 시스템의 일부분으로 모든 PDB데이터에 대해 구조적 이웃을 찾는 프로그램이다. VAST는 그래프 이론에서 사용하는 알고리즘을 사용하여 2차 구조 원소들을 정렬하여 두 개 구조로부터 온 모든 같은 2차 구조 원소들은 그래프의 노드로 표시된다. 그리고 두 개의 노드는 대응하는 2차 구조 원소들의 길이와 각도에 기반 하여 선으로 연결된다. 그러므로 그래프는 두 개의 구조에서 대응하는 같은 종류의 2차구조의 위상 및 연결 관계를 대표한다. 이 시작점에서 VAST는 정렬을 래지듀 레벨까지 확장 가능하다.

TOPS는 구조에 대한 2차원 위상다이어그램 서비스를 제공하는 것으로부터 시작되었다. TOPS의 흥미로운 점은 폴드에 대한 2차구조 요소와 그들의 수소-본드 패턴을 사용하여 데이터베이스에 대한 검색이 빠르다는 것이다. 또한 CATH나 SCOP데이터베이스의 대표적인 구조와 비교할 수 있다는 장점을 가지고 있다.

3. 플랫 파일 포맷 및 데이터베이스 설계

이 장에서는 구조 데이터를 추출하여 저장한 새로운 플랫파일포맷과 분석을 위한 단백질의 주석정보데이터베이스에 대해 설명한다.

3.1 PDB데이터베이스 플랫파일 분석

Protein Data Bank(PDB)는 실험으로 얻어진 생물학적 거대분자의 저장소로서 데이터는 원자좌표, 서열정보, 실험정보, 생물학적 구성 그리고 관련도서목록 등 정보들이 포함된다. 이러한 정보들은 9가지로 분류할 수 있고 거기에는 다음과 같은 세부 레코드들이 포함되어 있다.

■ ***Title 부분***

Title 부분에서는 생물학적 거대분자에 대한 기술을 기록한다.

- HEADER: idCode필드로서 PDB 엔트리를 정의하고 엔트리의 분류, 그리고 PDB에 저장된 날짜를 기록한다.
- OBSLTE: 배포에서 회수된 엔트리에서 볼 수 있다. PDB배포에서 어떤 다른 데이터에 의해 대체되였다는 것을 표시한다.
- TITLE: 엔트리에 대한 실험이나 분석의 주제가 들어간다.
- CAVEAT: 엔트리에 존재하는 엄중한 오류에 대해 경고한다.
- COMPND: 엔트리의 거대분자성분을 기술한다. 거대분자는 여러 개의 토큰으로 묘사되었다.
- SOURCE: 엔트리에 있는 생물학적 거대분자의 생물학적 혹은 화학적 성분을 기술한다.
- KEYWDS: 엔트리에 대한 분류 및 인덱싱할 수 있는 데이터를 가지고 있다.
- EXPDTA: 실험에 대한 데이터를 기록한다.
- AUTHOR: 엔트리에 대해 책임을 가지고 있는 사람들을 기술한다.
- REVDAT: 엔트리가 배포되어서부터 진행한 수정에 대한 정보를 가지고 있다.
- SPRSDE: 주어진 엔트리에 의해 버려진 엔트리의 정보를 가지고 있다.
- JRNL: 배포되는 좌표정보를 얻은 실험에 대해 기술한 문헌을 기술한다.
- PUBL: 저널이 아닌 출판정보를 가지고 있다.
- REFN: 인코딩된 참조정보를 가지고 있다.
- REMARK: 실험에 대한 상세 해석과 주석 및 코멘트 정보를 가지고 있다.

■ *1차구조 부분*

- DBREF: 엔트리에 대한 서열정보를 가지고 있는 데이터베이스 사이트의 링크를 제공한다.
- SEQADV: DBREF에 참조된 서열데이터베이스데이터와 PDB ATOM레코드에 있는 데이터사이의 관계를 표기한다.
- SEQRES: 잔기의 아미노산서열 혹은 핵산서열정보를 가지고 있다.
- MODRES: 단백질이나 핵산에 대한 수정정보를 가지고 있다.

■ *이질적인 데이터부분*

- HET: 표준적이 아닌 잔기에 대해 기술하는데 예를 들면 저애물질, 용매분자, 이온 등이다.
- HETNAM: hetID에 근거하여 성분의 화학적 이름을 기술한다.
- HETSYN: HET 그룹을 찾는데 도움을 주기 위한 내용을 가지고 있다.
- FORMUL: 화학방정식을 제시한다.

■ *2차구조 부분*

- HELIX: helix에 속하는 위치를 정의하기 위하여 사용된다. 전체 길이와 함께 helix의 시작위치와 끝위치가 정해진다.
- SHEET: 분자에 존재하는 sheet를 정의할 때 사용된다. sheet의 시작과 끝위치가 표기된다.
- TURN: 2차구조 조각들을 이어주는 굽힘 부분을 정의한다.

■ *좌표 부분*

- MODEL: 한개 좌표 엔트리에 여러 개의 구조가 기술되었으면 모델 일련번호를 붙여준다.
- ATOM: 일반적인 잔기의 원자좌표, 그리고 점유 및 온도계수를 기술한다.
- SIGATM: ATOM과 HEATM의 편차 값을 기술한다.

이러한 레코드들에 포함된 정보를 분석한 후 우리는 자동적으로 이러한 레코드들과 레코드에 기재된 데이터를 인식하여 우리가 설계한 새로운 구조데이터플랫파일 포맷으로 저장하고 기타 정보들 사이의 연관성에 따라 작성된 테이블에 저장할 파서들을 구축하였다. 새로운 구조파일 포맷과 데이터베이스 스키마는 다음절에서 소개된다.

3.2 구조정보 추출 및 플랫파일포맷 설계

기존의 구조비교 알고리즘은 모두 PDB플랫파일을 입력포맷으로 이용하였다. PDB 플랫파일 포맷은 구조데이터뿐만 아니라 기타의 많은 문헌정보와 모든 원자의 위치정보를 가지고 있다. 그러나 문헌데이터는 구조 비교 연산에 필요없고 또한 모든 원자들의 좌표정보도 모두 필요한 것은 아니다. 대부분의 구조정렬 알고리즘은 Ca원자의 정보만을 사용하는데 한개 아미노산은 8, 9개의 원자(Ca는 한개만)로 구성되었기 때문에 사실상 7/8의 원자좌표데이터는 필요 없다. 우리의 구조비교 알고리즘도 Ca원자의 좌표정보만 계산하는 방법을 사용하였기 때문에 이 논문에서 우리는 구조검색에 필요한 데이터들만 추출하여 새로운 플랫파일 포맷을 설계하였다. 이렇게 함으로서 불필요한 데이터의 처리로 인한 검색시간의 지연을 현저히 줄일 수 있고 저장 공간도 많이 절약 될 수 있다. 우리가 원본의 PDB 플랫파일에서 취한 레코드들은 HEADER, HELIX, SHEET, MODEL, ATOM, TER, ENDMDL, END등이다. 비교연산에서는 2차구조 정보를 필요로 하기 때문에 2차구조 정보가 들어있지 않는 데이터들에 대해서는 DSSP 2차구조 정의 프로그램을 이용하여 2차구조 부분을 찾은 후 새로운 플랫파일 생성시에 엔트리에 추가하였다. 새로운 플랫파일포맷에 대한 필드정보는 다음과 같다.

〈표 1〉 구조데이터 플랫파일의 필드정보

레코드 이름	필드 위치	데이터 타입	데이터	주석
HEADER	1 - 6	record name	"HEADER"	
	11 - 50	String (40)	classification	분자를 분류
	51 - 59	Date	depDate	저장한 날자
	63 - 66	IDcode	idCode	식별자 코드
COMPND	1 - 6	record name	"COMPND"	
	11 - 70	specification	compound	분자구성에 대한 서술
HELIX	1 - 6	record name	"HELIX"	
	16 - 18	residue name	initResName	시작 잔기의 이름
	20	character	initChainID	체인 식별자
	22 - 25	integer	initSeqNum	시작 잔기의 서열번호
	28 - 30	residue name	endResName	helix종말 잔기 이름
	32	character	endChainID	체인 식별자
	34 37	integer	endSeqNum	종말 잔기의 서열번호
SHEET	1 - 6	record name	"SHEET"	
	22	character	initChainID	체인 식별자
	23 - 26	integer	initSeqNum	시작 잔기의 서열번호
	33	character	endChainID	체인 식별자
	34 - 37	integer	endSeqNum	종말 잔기의 서열번호
MODEL	1 - 6	record name	"MODEL"	
ATOM	1 6	record name	"ATOM"	
	13 16	atom	name	원자 이름
	18 - 20	residue name	resName	잔기 이름
	22	character	chainID	체인 식별자
	23 - 26	integer	resSeq	잔기 서열번호
	31 - 38	real(8.3)	x	직각좌표계 x좌표 위치

〈표 1〉 구조데이터 플랫파일의 필드정보(계속)

레코드 이름	필드 위치	데이터 타입	데이터	주석
ATOM	39 - 46	real(8.3)	y	직각좌표계 y좌표 위치
	47 - 54	real(8.3)	z	직각좌표계 z좌표 위치
TER	1 - 6	record name	"TER"	
ENDMDL	1 - 6	record name	"ENDMDL"	
END	1 - 6	record name	"END"	

여기에서 식별자 코드와 단백질 이름은 비교연산을 진행한 후 관련데이터에 대한 질의를 하기 위해 필요하고, 2차 구조 정보는 2차 구조 데이터의 벡터 구조 정보를 위하여 필요하다. 원자 좌표정보에서도 오직 Ca원자의 x, y, z좌표 값과 잔기이름 그리고 서열순서만 추출하여 저장되었다. 다음 그림은 새로운 플랫파일의 샘플데이터이다.

3.3 단백질 주석정보 데이터베이스

이 논문에서는 다른 구조비교 검색엔진과 달리 구조비교의 결과와 함께 비교된 단백질의 주석 정보도 출력 인터페이스에 보여줌으로써 단백질구조 유사 부분뿐만 아니라 주석정보에 대해서도 비교 분석 할 수 있도록 하였다. 그러므로 단백질의 주석데이터는 예전의 플랫파일 형식이 아닌 데이터베이스형태로 저장되어 데이터베이스 질의를 통하여 이러한 관련데이터를 추출한다. 단백질 주석정보의 데이터베이스 스키마는 다음과 같이 구성되었는데 단백질의 식별자 코드와 이름, 저자등 정보를 이용하여 주석정보를 검색 할 수 있다. 현재 주석데이터베이스의 데이터는 PDB엔트리에 있는 데이터를 추출하여 저장되었는데 릴레이션 명세는 다음과 같다.

- Protein: IdCode와 실험이나 분석의 주제 및 저장된 날짜정보
- Compound: 엔트리의 거대분자에 대한 정보
- Source: 엔트리에 있는 거대분자의 생물학적 혹은 화학적 발원에 관련된 정보
- Keyword: 엔트리에 대한 분류 및 인덱싱할 수 있는 데이터정보
- Author: 저자에 대한 정보
- Citation: 실험에 대해 기술한 문헌정보
- CiteAuthor: 참조된 저널이나 논문의 저자 정보
- Sequence: 아미노산 서열 혹은 핵산서열 정보
- Remark: 실험에 대한 상세 해석과 코멘트 정보
- Heterogen: 엔트리에 있는 표준적이 아닌 잔기에 대한 기술정보
- Features: 중요한 의미를 가지는 위치정보

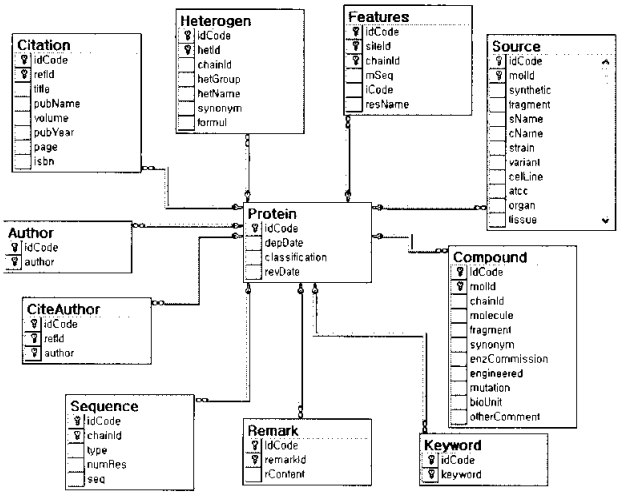
〈표 2〉 주석데이터베이스 스키마

Protein	idCode, depDate, classification, revDate
Compound	idCode, molId, chainId, molecule, fragment, synonym, enzCommission, engineered, mutation, bioUnit, otherComment
Source	idCode, molId, synthetic, fragment, sName, cName, strain, variant, celLine, atcc, organ, tissue, cell, organelle, secretion, celuarLoc, plasmid, gene, es, esStrain, esVariant, esCeline, esatccNum, esOrgan, esTissue, esCell, esOrganel, escellLoc, esVectortype, esVector, plasmid, esGene, otherComment
Keyword	idCode, keyword
Author	idCode, author
Citation	idCode, refId, title, pubname, volume, year, page, isbn
CiteAuthor	idCode, refId, author
Sequence	idCode, chainId, type, numRes, seq
Remark	idCode, remarkId, rContent
Heterogen	idCode, hetId, chainId, group, hetName, synonym, formul
Features	idcode, siteId, chainId, seq, iCode, resName

```
12345678901234567890123456789012345678901234567890
HEADER    MUSCLE PROTEIN                          02-JUN-93   1MYS
HELIX     GLY    A     86       GLY    A     94
HELIX     GLY    B     86       GLY    B     94
SHEET     THR    A     107      ARG    A     110
SHEET     ILE    A     96       THR    A     99-1
SHEET     ARG    A     87       SER    A     91-1
SHEET     TRP    A     71       ASP    A     75-1
MODEL     1
ATOM      CA     VAL   A   25   31.132  16.439  58.160
ATOM      CA     ALA   A   26   29.520  15.059  59.174
AtOM      CA     LEU   A   27   32.433  16.336  57.540
MODEL     2
ATOM      CA     GLN   B   96   29.909  16.996  55.922
ATOM      CA     CYS   B   97   28.870  17.401  57.336
ENDMDL
END
```

〈그림 1〉 구조정보 플랫파일 샘플

단백질 주석데이터베이스를 위해 설계된 E-R 다이어그램은 다음 그림과 같다.



(그림 2) 주석 데이터베이스 E-R다이어그램

4. 단백질 구조비교 시스템 설계

이 장에서는 우리가 구축한 전반 시스템의 구조와 시스템에서 사용한 단백질의 구조비교 연산에 대해서 설명한다.

4.1 시스템 구조

시스템은 주로 구조데이터 플랫파일, 주석정보데이터베이스, 데이터베이스 구축모듈 및 입출력 인터페이스로 구성되었다. 데이터베이스 구축모듈은 새로 배포된 PDB 엔트리들을 주기적으로 다운로드하여 구조데이터를 추출하고 새로운 플랫파일형식에 맞추어 구조데이터베이스를 구축한다. 다음 필요한 주석데이터는 주석정보데이터베이스에 입력된다. 따라서 입출력 인터페이스는 사용자의 질의를 받아서 구조비교와 주석데이터검색을 진행한 후 다시 웹을 통하여 전시한다.

4.2 단백질 구조비교 알고리즘

우리의 구조비교방법은 α-helix와 β-strand를 벡터 형식으로 표현하고 한 그룹의 일곱 가지 점수 정의 함수를 이용하여 한 쌍의 벡터를 두개의 단백질로부터 비교한다. 얻어진 값은 Dynamic Programming을 이용하여 최적의 지역정렬을 두 개의 벡터에서 찾아낸다. 이 알고리즘의 두 번째 단계는 원자의 좌표를 이용하여 맨 처음 정렬된 벡터부터 시작하여 반복적으로 RMSD값 을 줄여 가는 것이다.

여기에서 2차 구조의 시작 방향를 결정하여 기능적으로 더 좋은 평가를 하는 것이다. 우리가 base alignment를 진행하는 것은 α-helix와 β-strand이다. 모든 종류의 α-helix(α, π, 3/10, lefthand)를 한 개 클래스로 분류 하였다. 초반의 구조 Superposition을 얻기 위하여, 우리는 먼저 두 단백질의 각 helix와 strand들을 독립적인 벡터로 표현되고, 이 벡터는 Ca원자의 좌표를 이용하여 구해진다. 레지듀 i에서 시작하고 레지듀 j에서 끝나는 helix에 대해 아래의 공식이 벡터 시작과 끝점을 표현하는데 사용된다.

$$X_{origin} = (0.74 * X_i + X_{i+1} + X_{i+2} + 0.74 * X_{i+3})/3.48$$

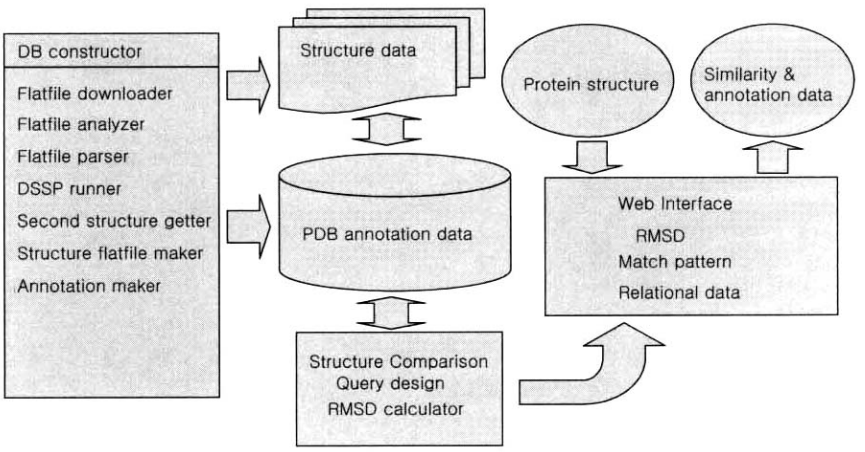
$$X_{end} = (0.74 * X_j + X_{j-1} + X_{j-2} + 0.74 * X_{j-3})/3.48$$

위의 공식을 사용함으로써 Helix에서 한쪽으로 치우치지 않게 중심을 찾아준다. 4개 이하의 레지듀를 가지는 helix는 helix로 보지 않고 한 개 레지듀의 확장으로 보고 0이 아닌 벡터로 인정하여준다. 마찬가지로 다음 공식은 레지듀 i와 j 사이의 Sheet의 시작과 끝점을 결정할 때 사용한다.

$$X_{origin} = (X_i + X_{i+1})/2$$

$$X_{end} = (X_j + X_{j-1})/2$$

이리하여 두 단백질을 H 또는 S벡터로 감소시켰다. 다음 우리는 Dynamic Programming을 사용하여 매치가 되는 가장 긴 서열을 찾는다. 알고리즘에서 사용되는 Scoring Func-



(그림 3) 시스템 구조도

tion은 이 두 벡터의 유사성을 표현하는 점수를 반환한다. 우리는 처음에 Alignment가 연관된 방향과 변환에 관계없이 계산하기 위하여 Orientation Independent와 Orientation dependant의 두 가지 Scoring Function을 정의하였다.

Orientation Independent 점수는 두 단백질에서 선택된 벡터내부의 각도와 거리를 비교하여 얻어진다. Orientation Dependent점수는 예를 들면 단백질 A의 벡터 i와 벡터 k사이의 각도를 단백질 B의 벡터 p와 비교하여 orientation dependant 점수를 생성한다.

우리가 정의한 seven score function은 아래에 열거된다. 이 함수로 비교되는 벡터들은 그림 3과 같다. 실선으로 표시된 부분만이 벡터를 표시한다. 점선은 가운데 벡터로서 두 2차 구조 사이 결합하는 시작과 끝점만 표현한다. 아래에 보여준 함수들의 합은 벡터 k와 r사이의 점수를 미리 정렬된 두 벡터 i와 p의 점수와 합하여 최종 유사성 값을 준다.

Orientation Independent Scores:

$$S_1 = S\{|\, angle(i,k) - angle(p,r)\,|\}$$

$$S_2 = S\{|\, angle(i,j) - angle(p,q)\,|\}$$

$$S_3 = S\{|\, angle(j,k) - angle(q,r)\,|\}$$

$$S_4 = S\{|\, dis\tan ce(i,k) - dis\tan ce(p,r)\,|\}$$

$$S_5 = S\{|\, length(k) - length(r)\,|\}$$

Orientation Dependent Scores:

$$S_6 = S\{angle(k,r)\}$$

$$S_7 = S\{dis\tan ce(k,r)\}$$

두 벡터사이의 거리는 대응하는 벡터의 시작, 중간, 마지막 점의 거리 평균을 구하는 것으로 구해진다.

각도는 두개 대표적인 유닛 벡터의 Dot Product의 역 코싸인 값을 구함으로써 얻어진다. 위의 공식에서 M parameter는 비교되는 속성의 민음직한 무게 계수와 같은 작용을 한다. Seven Score Function에서는 여러 가지 M, do 값을 이용하여 비교가 되였다. 예를 들면 최고점수5 (함수 S7)를 주는 것으로 길이가 강조되고 각도 점수는 10(함수 S6)을 주어서 계산된다. 점수들은 <표 3>에 나타나 있다.

〈표 3〉 일곱 가지 점수 부여 값

	S1	S2	S3	S4	S5	S6	S7
M	10	4	4	2	5	10	5
do	45	5	5	A	7	45	A

우리는 위에서 정의한 Scoring Function을 이용하여서 Dynamic Programming을 수행하여 두 단백질의 벡터를 정렬하였다. 사용된 Score Function에 근거하여 우리는 Orientation Independent와 Orientation Dependent의 결과를 얻을 수 있다. 우리는 smith-waterman 알고리즘의 변형을 사용하여 2차 구조 벡터를 정렬한다. 그리고 모든 벡터를 감안하여 갭 패널티를 0으로 주었다. 우리는 Smith-Waterman 알고리즘을 임의의 시점에서도 감소하지 않도록 설계하였다. 그리하여 i, j위치에서 계산된 점수가 p, q에서 계산된 점수보다 작으면 i, j는 0으로 설정된다. 아래의 (그림 4)는 Local Secondary structure superposition 단계로써 위의 2차 구조 정렬기술을 사용하여 두 단백질의 방향에 관계없는 첫 Superposition을 찾아준다. 기존의 방법은 자주 틀린 superposition을 반환하므로 하위-최적화 알고리즘을 이용하여 가장 적합한 정렬을 찾는다. 이 문제를 해결하기 위하여 다음 단계를 대신 실행한다.

1. 모든 질의 단백질의 벡터에 대해서 대응하는 벡터를 찾는다. 두 쌍의 벡터는 반드시 선택되어야 한다.
2. 위에서 선택된 4개의 벡터를 가지고 두 개를 가장 잘 맞추어주는 Transformation Matrix를 찾아준다. 이 Matrix를 전체 질의 단백질에 응용한다.
3. 2번에서 찾은 시작 Alignment에 대하여 Orientation Independent, Orientation Dependent 두 개를 사용하여 Dynamic Programming을 한다. 최고 점수만 필요하기 때문에 Alignment를 찾는 Trace-back은 이루어지지 않는다.
4. Transformation은 3번에서 얻은 점수로 전체 단백질에 대해 진행한다.
5. Dynamic Programming에 Orientation Independent와 Orientation Dependent를 이용하여 두 단백질을 다시 정렬한다.
6. Step5에서 정렬된 각 2차구조 Element들의 대상 단백질로 정렬하는 가장 작은 RMSD의 Transformation Matrix를 찾는다.
7. Transformation을 응용하고 RMSD가 수렴할때까지 5,6,7번을 반복한다.

(그림 4) Superposition을 찾는 단계

최초의 정렬을 찾은 (그림 4)의 단계를 Global Optimal Alignment라고 하면 다음 (그림 5)의 세 과정를 통하여 두 단백질 원자들 사이의 RMSD를 감소시킨다.

1. 질의 단백질의 모든 원자에 대하여 가장 가까운 원자를대상 단백질에서 찾는다.
2. Transformation을 하여서 RMSD를 감소시킨다.
3. RMSD가 수렴할때까지 step1,2를 반복한다. 여기서 Alignment Space는 6차원인데 3개는 Rotation이고, 각도 3개는 Transformation이다.

(그림 5) Atomic Superposition 단계

RMSD값이 정렬의 품질을 평가하는 기준이긴 하지만 정렬에서 한 Alignment와 그 다음 Alignment의 RMSD값 차이는 매우 클 수 있다. 따라서 우리는 RMSD값과 잘 정렬된 원자 두 가지를 사용하여 정렬을 평가하기 위하여 (그림 6)의 두 가지 테스트를 수행하여 정렬이 잘 되였는지를 평가한다.

1. 모든 단백질A의 원자에 대하여 단백질B에서 가장 가까운 원자를 찾는다. 모든 B의 원자의 가장 가까운 원자를 찾는다. (T에 의해 정렬된 원자만 고려한다.) 가장 가까운 원자로 찾은 원자 쌍을 모두 선택한다. 예를 들어 i가 A에 있고 p가 B에 있는데 i가 p를 가장 가까운 이웃으로 p가 i를 가장 가까운 이웃으로 찾았을 때이다.
2. 1번에서 co-linearity와 상반되는 모든 원자 쌍을 제거한다. 여기에서 선택된 atom은 핵심 정렬로서 인정받게 된다. 이 알고리즘의 복잡도는 두 부분으로 나눌수 있다. step1은 O((max(n,m))*n*m). 여기서 m, n은 두개 단백질에 있는 2차구조의 수이다. step2, 3는 O(n*m)여기서 m, n은 두 단백질의 Ca개수이다.

(그림 6) Core superposition 단계

5. 구현 및 평가

이 장에서는 우리가 구현한 새로운 플랫파일 포맷 생성기, 문헌정보 데이터베이스 그리고 구조비교 시스템에 대해 설명하고 기타 구조비교시스템과의 비교 평가에 대해 기술한다.

5.1 구현환경

이 논문에서 소개하는 시스템의 구현환경과 프로그래밍환경은 다음과 같다. CPU는 P3 1.8GHZ, 메모리는 512MB를 사용하였다. 플랫폼은 Wow Linux 6.0을 설치하였고 DBMS는 MySQL v4.0.14를 사용하였다. 프로그래밍 환경으로 Java API버전 J2sdk1.4.2을 사용하였고 여기에 프로그래밍 인터페이스 JCreater LE를 설치하였다. 프로그래밍에 사용된 언어는 Java, C, SQL이다.

5.2 구현 결과

5.2.1 플랫파일 파서

플랫파일 파서의 역할은 기존의 PDB플랫파일에서 2차구조와 3차구조등 우리의 구조비교 알고리즘에 필요한 레코드들을 추출하여 새로운 구조데이터 플랫파일을 만드는데 있다. 이러한 파서는 플랫파일에서 HELIX, SHEET 및 ATOM 레코드를 식별하고 거기에 대응하는 원자 좌표정보, 즉 Ca 원자의 좌표정보, 그리고 2차구조의 시작 위치와 끝 위치를 선택하여 새로운 플랫파일에 저장하는데 사용된다. 여기서 PDB플랫파일에 2차구조의 위치가 기술되지 않은 경우가 있는데 우리는 DSSP프로그램을 사용하여 Ca원자들의 위치로부터 단백질 내에 존재하는 2차구조정보를 저의한후 이러한 결과를 플랫파일에 추가하는 방법으로 모든 단백질구조 플랫파일에 2차구조정보가 포함되게 하였다. 플랫파일 인식알고리즘은 다음과 같다.

```
입력: PDB 플랫파일
출력: IDcode.ent
01 : pdb엔트리파일을 연다.
02 : while ((s = in.readLine()) != null){
03 :     if(s.substring(0,6).compareTo("HEADER")==0){
04 :         header에서 식별자코드와 분자이름을 추출하여 출력파일에 쓴다.
05 :     }else if(s.substring(0,6).compareTo("COMPND")==0){
06 :         구성정보를 추출하여 출력파일에 쓴다.
07 :     }else if(s.substring(0,6).compareTo("HELIX")==0){
08 :         2차구조 위치를 추출하여 출력파일에 쓴다.
09 :         존재하지 않을 경우 DSSP프로그램을 이용하여 2차구조정보를 얻는다.
10 :     }else if(s.substring(0,4).compareTo("ATOM")==0&&s.substring(13,15).compareTo("CA")==0){
11 :       Ca원자의 좌표정보를 추출하여 출력파일에 쓴다.
12 :     ...
13 :     }
14 : }
15 : in.close();
16 : out.close();
```

(알고리즘 1) 플랫파일파서 알고리즘

```
입력: PDB 플랫파일
출력: DB테이블에 입력
01 : myConn = DriverManager.getConnection(myURL, myID, myPasswd);
02 : PDB엔트리파일을 연다.
03 : while ((s = in.readLine()) != null){
03 :     HEADER, COMPND등 레코드에서 주석정보를 추출한다.
04 :     주석정보를 데이터베이스 스키마에 따라 분류한다.
05 :     DbConnector c = new DbConnector();
06 :     c.makeConnection();
07 :     c.upLoad( 데이터베이스 질의어 );
08 :     c.takeDown();
09 : }
10 : 열린 파일을 닫는다.
```

(알고리즘 2) 주석데이터베이스구축 알고리즘

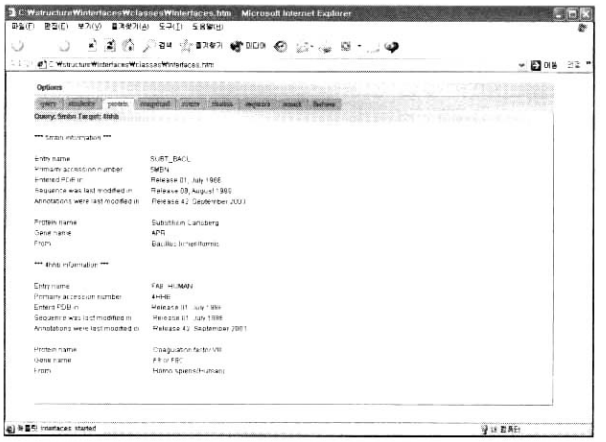
5.2.2 주석정보데이터베이스 구축

기존의 단백질 구조비교시스템에서는 단백질 구조정보와 연관된 문헌정보는 관계형 데이터베이스로 구축되지 않았었다. 그리하여 구조비교를 거친 후 기타 정보들을 검색하기 위하여 단백질의 플랫파일들을 검색하여 그 안에서 필요한 정보들을 읽어야만 했다. 이러한 과정의 번거로움과 그중에서 유용한 정보를 손실할 수 있는 단점을 고려하여 우리는 단백질과 관련된 문헌정보도 분류를 거쳐 데이터베이스 스키마를 작성하고 관련정보를 데이터베이스화함으로써 단백질의 구조비교와 동시에 연관된 단백질의 특성정보를 분석할 수 있도록 하였다. 주석데이터베이스 구축 알고리즘은 다음과 같다.

5.2.3 입출력 인터페이스

우리의 시스템은 주석데이터를 관계형 데이터베이스에 저장하고 구조비교 알고리즘과 연동하는 구조를 사용하였기 때문에 다른 구조비교 시스템보다 체계적인 분석환경을 제공한다. 사용자가 웹 환경을 이용하여 질의한 구조는 데이터베이스내의 단백질구조와 비교를 진행한 후 (그림 7)과 같은 유사성 검색결과를 보여준다.

(그림 7) 구조유사성 검색 결과



(그림 8) 비교된 단백질의 문헌데이터

(그림 7)의 유사성결과 화면에서 RMSD값과 매치되는 아미노산 잔기들을 볼 수 있다. 단백질들과 연관된 주석데이터는 동시에 데이터베이스 질의어를 통하여 주석데이터베이스에서 추출된 후 (그림 8)과 같이 여러 페이지로 분류하여 보여준다. 이렇게 함으로써 두 분자의 구조뿐만 아니라 주석데이터에 대해서도 비교분석하고 분자의 기능과 특성을 예측하는데 활용할 수 있다.

5.3 성능평가

우리의 시스템에 대한 성능평가는 구조비교의 정확성과 구조비교 시간의 효율성으로 나누어 진행하였다. 따라서 우리는 기존의 LOCK시스템과 비교평가를 진행하였고 기타의 구조비교 시스템과도 비교를 진행할 수 있도록 [4]에서 제시한 데이터와 같은 데이터를 사용하였다. 질의 데이터는 <표 5>의 3개의 샘플 1mbd, 1tph-2, 8fab-A를 사용하였는데 이들을 선택한 목적은 공간 구조상 완전히 틀린 세 가지 형태를 하고 있기 때문에 시스템의 정확성에 대해 더 잘 평가할 수 있기 때문이다. 대상 데이터로는 SCOP 데이터베이스에서 같은 클래스에 속하는 685개의 데이터를 사용하여 구조적으로 비슷한(true positive) 또는 연관 없는(false positive)로 분류할 수 있었다.

〈표 4〉 성능평가에 사용된 질의 데이터정보

인식 코드	길이	Scop 클래스분류	Scop 폴드분류
1mbd	153	All-α	Globin-like
1tph-2	245	α and β	α/β (TIM) barrel
8fab A	208	All-β	Immunoglobulin-like

90%의 cutoff값을 가지고 대상데이터에 대해 검색한 결과는 <표 5>와 같다.

〈표 5〉 cutoff value 90%로 검색한 결과

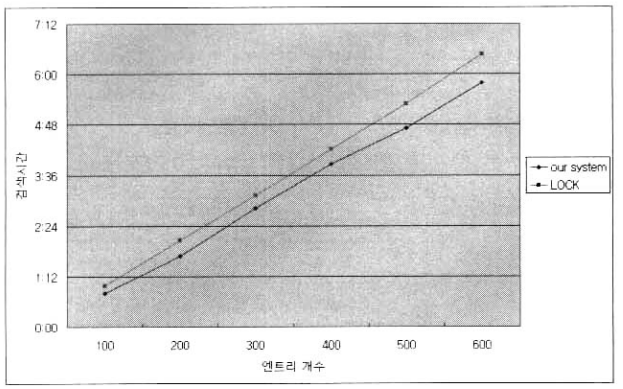
질의 단백질	검색된 개수	오류 개수	검색된 단백질
1mbd	11	1	1mbd, 1bab-B, 2fal, 1eca, 2hbg, 1hlb, 1ash, 3sdh-A, 2gdm, 1cpc-A, 1cpc-B
1tph-2	51	5	1tph-2, 5tim-A, 1nsj, 1fba-A, 1jul, 1dhp-A, 2tys-A, 2tmd-A, 1nal-1, 1ltd-A, 1oyc, 1nar, 1pkm, 1cnv, 1ctn, 1ceo, 1luc-A, 1dor-A, 1gow-A, 1ece-A, 1dos-A, 2myr, 1qba, 1pud, 1ghr, 2acq, 1sft-A, 1req A, 1byb, 2ebn, 2mnr, 1edg, 5rub-A, 2amg, 1psc A, 1ucw-A, 1tcm-A, 1bgl-A, 1xyz-A, 2chr, 1qap-A, 2aaa, 1fkx, 2kau-C, 4xia-A, 4enl, 1smd, 1gym, 1djx-A, 1nfp, 1bpl-A
8fab-A	38	4	8fab-A, 1osp-L, 1fc2-D, 1dlh-B, 1dlh-A, 8fab-B, 1mhc-A, 1bec, 1fru-A, 2ncm, 1tlk, 1zxq, 1vsc-A, 1tcr-A, 1neu, 1hng-A, 1fnf, 1ebp-A, 1bgl-A, 1cid, 1fie-A, 4kbp-A, 3hhr-C, 3dpa, 2hft, 1svb, 1ctn, 1cfb, 1cdy, 1clc, 1oxy, 1jcv, 1gof, 1nfk A, 1msp-A, 1tcm-A, 1edh-A, 1qba

〈표 6〉 cutoff value 90%에서 각 시스템의 오류검색 개수

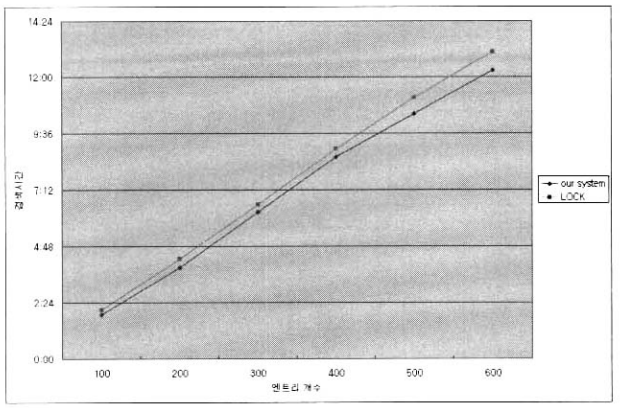
	DALI	STRUCTAL	VAST	MINAREA	LOCK	3dSEARCH
1mbd	0	1	2	0	0	2
1tph-2	1	1	0	330	1	0
8fab-A	3	159	1	573	4	185

같은 Cut-off 값을 가지고 기존의 구조비교 시스템들을 비교한 결과는 [4]에 다음과 같이 제시 되었는데 결과에서 우리는 새로운 시스템이 기존의 LOCK 시스템과 동등의 검색 정확성을 제공하고 DALI와 같이 구조비교시스템에서 가장 좋은 성능을 보여주고 있다는 것을 알 수 있다.

다음 우리는 LOCK시스템과 검색시간의 효율성에 대하여 평가하였다. 그래프에서 All a구조로 구성된 1mbd의 대상데이터가 증가함에 따라 변하는 검색시간을 비교하였다. 우리의 시스템은 전반 테스트과정에 LOCK시스템보다 빠른 검색시간을 사용하였다.

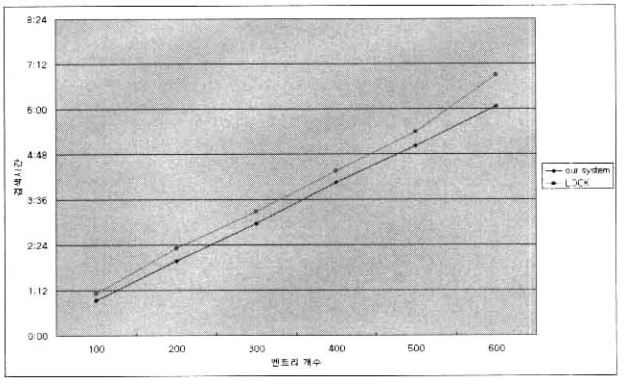


(그림 9) 1mbd의 검색시간 비교



(그림 10) 1tph-2의 검색시간 비교

1tph-2는 기타 두 가지 질의 데이터보다 복잡한 α and β가 공유하는 형태를 취하였는데 이 검색은 다른 두 가지 샘플보다 1배정도 많은 검색시간을 소모하였다. 하지만 우리의 시스템은 여전히 LOCK보다 빠를 검색성능을 보여주었다.



(그림 11) 8fab-A의 검색시간 비교

전부 β구조로 구성된 8fab-A의 검색시간비용은 대체로 1mbd와 비슷하였다. 그래프를 통하여 우리는 우리의 검색시스템이 LOCK보다 적은 검색시간을 소모한다는 것을 알수 있다. 하지만 전반 검색과정에서 데이터의 양이 증가함에 따라 검색시간도 선형으로 증가하는 것을 볼수 있다. 이는 구조비교를 진행함에 있어서 순서적인 비교를 하여야 하기 때문이다.

6. 결 론

서열이나 구조패턴은 기능적 혹은 구조적으로 연관된 여러 개의 단백질을 분류하는데 사용할 수 있다. 이러한 패턴의 연구는 서열, 구조와 기능사이의 연관을 이해하고 더 나아가서 생체의 기능을 이해하는데 있어 아주 중요하다. 하나의 새로운 단백질 서열을 발견하면 데이터베이스 유사성 검색은 이와 유사한 단백질을 정의할 수 있다. 따라서 이러한 유사성이 아주 강하면 우리는 그 새로운 단백질도 데이터베이스 내에 있는 단백질과 유사한 구조, 심지어는 비슷한 기능을 가지고 있다고 추정할 수 있다.

이 논문에서 우리는 웹 기반으로 단백질의 구조를 구조데이터베이스와 비교하여 새로운 구조가 이미 알려진 임의의 구조와 유사성을 가지고 있는지 또는 새로운 폴드가 있는지를 검색해주고 따라서 기존 단백질의 문헌정보도 제공하는 시스템을 구축하였다.

이 시스템은 PDB로부터 단백질의 플랫파일을 다운로드하고 분석을 통하여 엔트리에 있는 각 레코드를 분석하고 데이터를 인식하는 시간과 저장 공간을 절약 위하여 2차 구조와 3차구조등의 구조비교프로그램에 필수적인 데이터는 추출하여 새로운 플랫파일형태로 저장하고 기타 단백질연구

에 필요한 데이터는 데이터베이스 스키마에 맞추어 데이터베이스를 구축하였다. 그리하여 구조비교와 단백질연구에 필요한 데이터는 보다 빠르게 검색되고 사용자친목적으로 보여 질 수 있다. 향후연구로는 단백질 구조에서 매치되는 부분을 그래픽적으로 보여줌으로서 보다 직관적으로 분석할 수 있도록 시스템을 업그레이드 하고, 단백질 구조 비교 결과를 이용하여 다양한 입력된 서열 간의 클래스 형성 정보를 웹을 통하여 서비스 할 예정이다.

참 고 문 헌

[1] N.P.Brown, C.A.Orengo, W.R.Taylor, "A protein structure comparison methodology", Computers Chem, Vol.20, pp. 359-380, 1996.

[2] L.Holm, C.Sander, "Protein structure comparison by alignment of distance matrices", Biol, Vol.233, pp.123-138, 1993.

[3] R.Brschweiler, "Efficient RMSD measures for the comparison of two molecular ensembles". PROTEINS: Structure, Function, and Genetics, Vol.50, pp.26-34, 2003.

[4] A.P.Singh, D.L.Brutlag, "Hierarchical protein structure superposition using both secondary structure and atomic representations", Bioinformatics, Vol.5, pp.284-293, 1997.

[5] I.N.Shindyalov, P.E.Bourne, "Protein structure alignment by incremental combinatorial extension(CE) of the optimal path", J.Mol.Biol, Vol.233, pp.123-138. 1998.

[6] C.I.Branden, J.Tooze, "Introduction to Protein Structure", Garland Publishing, 1991.

[7] A.M.Lesk, "Introduction to Protein Architecture: The Structural Biology of Proteins", Oxford Press, 2001.

[8] A.Bairoch, R.Apweiler, "The Swiss-Prot protein sequence data bank and its supplement TrEMBL in 2000", Nucleic Acids Res, Vol.28, pp.45-48, 2000.

[9] H.M.Berman, J.Westbrook, Z.Feng, G.Gilliland, T.N.Bhat, H.Weissig, I.N.Shindyalov, P.E.Bourne, "The Protein Data Bank", Nucleic Acids Research, Vol.28, pp.235-242, 2000.

[10] C.A.Orengo, A.D.Michie, S.Jones, D.T.Jones, M.B.Swindells, J.M.Thornton, "CATH - A Hierarchic Classification of Protein Domain Structures", Structure, Vol.5, pp.1093-1108, 1997.

[11] W.Kabsch, C.Sander, "Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features", Biopolymers, Vol.22, pp.2577-237, 1983.

[12] K.Kedem, L.P.Chew, R.Elber, "Unit-vectore RMS (URMS) as a tool to analyze molecular dynamics trajectories", Proteins: Structure, Function and Genetics, Vol.37, pp. 554-564, 1999.

[13] J.F.Gibrat, T.Madej, S.H.Bryant, "Surprising similarities in structure comparison", Curr Opin Struct Biol, Vol.6, pp. 377-385, 1996.

[14] D.Gilbert, D.Westhead, N.Nagano, J.Thornton. "Motif-based searching in TOPS protein topology databases", Bioinformatics, Vol.15, pp.317-326, 1999.

[15] R.Samudrala, J.Moult, "A graph-theoretic algorithm for comparative modeling of protein structure", J.Mol.Biol, Vol.279, pp.287-302, 1998.

[16] I.Eidhammer, I.Jonassen, "Protein structure comparison and structure patterns", ISMB2001 Tutorial, 2001.

[17] X.Pennec, N.Ayache, "A geometric algorithm to find small but highly similar 3D substructures in proteins", Bioinformatics, Vol.14, pp.516-522, 1998.

[18] L.P.Chew, D.Huttenlocher, K.Kedem, and J.Kleinberg, "Fast detection of common geometric substructure in proteins", Journal of Computational Biology, Vol.6, pp.313-325, 1999.

[19] G.M.Maggiora, D.C.Rohrer, J.Mestres, "Comparing protein structures: A Gaussian-based approach to the three-dimensional structural similarity of proteins", Journal of Molecular Graphics and modeling, Vol.19, pp.168-178, 2001.

[20] A.S.Yang, B.Honig, "An integrated approach to the analysis and modeling of protein sequences and structures. I. Protein structuralalignment and a quantitative measure for protein structural distance", J.Mol.Biol, Vol.301, pp.665-678, 2000.

[21] I.Jonassen, I.Eidhammer, D.Conklin, W.R.Taylor, "Structure motif discovery and mining the PDB", Bioinformatics, Vol.18, pp.362-367, 2002.

[22] I.N.Berezovsky, E.N.Trifonov, "Protein structure and folding: A new start", Journal of Biomolecular Structure & Dynamics, Vol.19, No.3, 2001.

[23] R.SnChez, U.Pieper, F.Melo, N.Eswar, M.A.Mart-Renom, M.S.Madhusudhan, N.Mirkovi and A.ali, "Protein structure modeling for structural genomics", Nature Structural Biology, Structural genomics supplements, 2000.

정 광 수

e-mail : ksjung@dblab.chungbuk.ac.kr

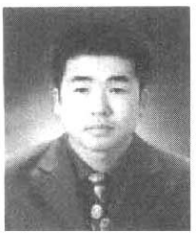
2001년 충북대학교 화학공학부(공학사)

2004년 충북대학교 정보산업공학과 (공학석사)

2004년~현재 충북대학교 대학원 전자계산학과 박사과정

관심분야 : Bioinformatics, 단백질 서열 및 구조, 생명정보 데이터베이스, 시스템 생물학, 구조 유전체학

한 욱

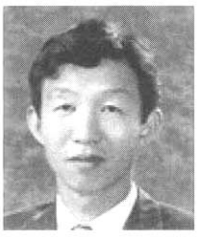


e-mail : wogia@dblab.chungbuk.ac.kr
2001년 연변과학기술대학 컴퓨터과학과 (이학사)
2004년 충북대학교 대학원 전자계산학과 (이학석사)
관심분야 : Bioinformatics, 생명정보 데이터 통합, 생명정보 데이터베이스

박 성 희

e-mail : shpark@dblab.chungbuk.ac.kr
1996년 충북대학교 도시공학과 졸업 (공학사)
1998년 한국전자통신연구원 컴퓨터소프트웨어연구소 위촉연구원
2001년 충북대학교 대학원 전자계산학과 (이학석사)
2003년 충북대학교 대학원 전자계산학과 박사수료
2004년 University of Glasgow, Bioinformatics Research Centre 방문연구원
관심분야 : Bioinformatics, Structural Bioinformatics, System Biology 단백질 구조 분류 및 비교, 단백질 상호작용 예측, 패턴발굴 및 매칭 Xml & Xml 데이터베이스, 시공간 데이터베이스

류 근 호



e-mail : khryu@dblab.chungbuk.ac.kr
1976년 숭실대학교 전산학과 (이학사)
1980년 연세대학교 공업대학원 전산전공 (공학석사)
1988년 연세대학교 대학원 전산전공 (공학박사)
1976년~1986년 육군 군수 지원사 전산실(ROTC 장교), 한국전자통신연구소(연구원), 한국방송대학교 전산학과(조교수) 근무
1989년~1991년 University of Arizona, Research Staff(TempIS 연구원, Temporal DB)
1986년~현재 충북대학교 전기전자컴퓨터공학부 교수
관심분야 : 시간 데이터베이스, 시공간 데이터베이스, Temporal GIS, 객체 및 지식 데이터베이스 시스템, 지식기반 정보검색 시스템, 데이터마이닝, 데이터베이스 보안 및 바이오인포매틱스 등