BSML 기반 능동 트리거 규칙을 이용한 염기서열정보관리시스템의 구현

(Implementation of an Information Management System for Nucleotide Sequences based on BSML using Active Trigger Rules)

박 성 희 † 정 광 수 †† 류 근 호 †††

(Sung Hee Park) (Kwang Su Jung) (Keun Ho Ryu)

요 약 유전체 서열을 포함하는 생물정보는 지속적으로 변화하며 이질적이고 다양하다는 특성을 갖는다. 이러한 생물 정보의 특성을 반영한 관리시스템이 요구되지만 현재 대부분의 기존 생물정보 데이타베이스는 생물 데이타에 대한 저장소로만 이용된다. 따라서 이 논문에서는 생물학 연구실 수준에서 시퀀싱 실험을 통해 생산되거나 다양한 공개용 데이타베이스로부터 수집된 염기 서열 데이타를 파일 포맷 변환, 편집, 저장 및 검색을 수행하는 서열정보관리 시스템을 제시한다. 이질적인 서열 포맷간의 파일 변환을 위하여 XML기반 BSML을 공통 포맷으로 이용한다. 서열 저장관리에서는 동일한 DNA 조각에 대한 서열 구성의 변경정보를 저장하기 위해 서열 버전을 정의하고 능동 트리거 규칙을 이용하여 변경 정보 검출 및 생성 방법을 보여준다. 트리거 기능을 이용하여 서열의 변경 정보를 자동적으로 데이타베이스에서 저장관리 할 수 있음을 보이고 성능을 평가하였다.

키워드 : 생명정보학, 염기 서열, 서열 관리, 서열 버전, BSML

Abstract Characteristics of biological data including genome sequences are heterogeneous and various. Although the need of management systems for genome sequencing which should reflect biological characteristics has been raised, most current biological databases provide restricted function as repositories for biological data. Therefore, this paper describes a management system of nucleotide sequences at the level of biological laboratories. It includes format transformation, editing, storing and retrieval for collected nucleotide sequences from public databases, and handles sequence produced by experiments. It uses BSML based on XML as a common format in order to extract data fields and transfer heterogeneous sequence formats. To manage sequences and their changes, version management system for originated DNA is required so as to detect transformed new sequencing appearance and trigger database update. Our experimental results show that applying active trigger rules to manage changes of sequences can automatically store changes of sequences into databases.

Key words : Bioinformatics, Nucleotide Sequence, Sequence Management, Sequence Version, BSML

1. 서 론

2000년에 초본이 발표된 인간 게놈 프로젝트는 생명체로부터 유전체 서열 획득을 위한 시퀀싱 기술의 발전을 가져왔다. 이러한 결과로 웹을 통한 대량의 생물학적 서열 데이타가 배포되고 생물학 실험실에서도 생물체의 염기 서열을 자체적으로 생산하게 되었다. 그러나 대부분의 생물학 연구실에서는 생산된 서열 데이타를 저장관리할 수 있는 소프트웨어가 존재하지 않아 파일형태로 디스크에 저장해 두고 이용한다. 생물학 연구실에서

· 이 연구는 2002년도 KISTI 바이오인포메틱스 연구센터와 2004년도 KISTEP 연구비 지원에 의하여 수행되었음

† 비 회 원 : 충북대학교 전자계산학과
shpark@dblab.chungbuk.ac.kr
†† 비 회 원 : 충북대학교 전자계산학과
ksjung@dblab.chungbuk.ac.kr
††† 종신회원 : 충북대학교 전기전자컴퓨터공학부 교수
khryu@dblab.chungbuk.ac.kr

논문접수 : 2003년 10월 7일
심사완료 : 2004년 10월 19일

생산된 서열 데이타의 진보적인 분석과 응용을 위해서는 자체 생산된 서열 데이타와 비교 분석을 위해 외부의 공개용 생물데이타베이스로부터 수집된 서열 데이타를 동시에 저장 관리할 수 있는 염기서열정보관리시스템이 필요하다.

염기 서열을 포함하는 생물정보는 데이타의 특성이 다른 데이타와 다르며 이러한 특성을 반영한 관리시스템이 요구된다. 아래는 염기서열정보관리시스템에서 고려해야 할 염기 서열 데이타의 특성을 나열하였다.

- 생물학적 진화 및 실험 환경 요인으로 서열 변경이 자주 발생한다[1-4].

ABI같은 시퀀싱 기계로부터 얻어진 베이스 구성 정보로부터 어셈블리 및 편집 과정을 거쳐 완전한 하나의 서열을 얻기까지 많은 프로그램들이 이용된다. 이러한 프로그램 상의 알려지지 않은 오류 및 생물학적 진화 과정상의 변이 등에 의해 서열이 변경될 수 있다. 따라서 비록 동일한 DNA 분자라 해도 다시 시퀀싱 실험을 하면 서열의 염기 구성이 달라질 수 있다. 현재로써는 이러한 동일한 DNA 분자에 서로 다른 베이스 구성을 갖는 두 개의 서열 중 어느 서열이 정확한 서열인지 구분이 명확하지 않다. 서열 데이타의 저장관리 시 동일한 DNA 서열의 베이스 변경정보도 함께 저장되어야 한다.

- DNA 염기 서열은 A, T, G, C의 네 개 문자의 반복으로 이루어진 비정형의 문자열이다[1,5].

서열 데이타에 대한 정확한 길이가 서로 다른 유전체마다 다르며 규칙성이 없는 비정형 데이타 타입이다. 사용자에 의해 비정형 서열 데이타가 변형되지 않고 일치성있게 데이타베이스에서 저장관리하기 위한 데이타베이스 디자인이 요구된다.

- 이질적인 데이타 포맷을 이용하는 분석 프로그램간서열 데이타 교환을 위한 공통 포맷이 요구된다[4,6-9].

서열 데이타는 이것을 표현하는 데이타 포맷과 데이타베이스 및 분석 프로그램마다 서로 다른 포맷을 이용한다. 서열 데이타 포맷은 서열 관련 정보의 검색 및 생물학적 지식을 추출, 데이타간의 연결 및 연산을 쉽게 할 수 있는 데이타 기술 포맷이어야 한다. 또한 장기적인 관점에서 서열 정보 분석을 위해서 분석 프로그램 및 데이타 변화를 수용할 수 있는 공통 모델로써 역할[10-12]을 해야 한다.

- 실험을 통해 얻어진 원시 데이타에 대한 해석을 위한 추가적인 주석 정보를 가지며 여러 단계의 분석과정을 통해 많은 유도된 데이타가 창출된다.

전 세계적 규모로 유전체 서열 통합데이타베이스 검색을 지원하는 미국의 NCBI(National Center for Biotechnology Information)[13], 일본의 DDBJ(DNA Data-Bank of Japan)[14], 유럽의 EMBL(European Molecular Biology Laboratory)[10]은 염기 서열 데이타 저장 및 검색 서비스와 분석 S/W를 제공한다. NCBI의 GenBank[4,8,15]는 공통포맷으로 ASN.1 포맷을 이용하고 서열의 변경을 서열 레코드 식별자에 함께 표현한다. 초기에 서열의 변경 정보를 지원하지 않아 서열 레코드 갱신에 따른 서열의 변경 정보를 식별할 수 없었다. 또한 ANS.1 포맷을 이용하여 서열포맷에 대한 질의를 지원하지 않으며 확장성이 없다는 단점을 갖는다. 현재는 xml 기반의 BSML(Bioinformatics Sequence Markup Language)을 정의하여 서열 및 관련 정보를 표현하려는 시도를 하고 있다. 영국의 Sanger 연구소에서도 시퀀싱된 염기 서열을 생산하며 생산된 인간 염기 서열을 저장하기 위한 데이타베이스로 객체지향 데이타베이스를 확장한 AceDB를 자체적으로 운영한다. 시퀀싱의 마지막 단계에서 서열을 편집 및 질을 평가하고 데이타베이스에 저장하기 위한 소프트웨어로 Staden Package[14,16-18]를 이용한다. 그러나 Staden Package 역시 파일 상태로 서열 데이타를 저장하고 포맷은 일반 텍스트 파일을 이용하여 서열 포맷에 대한 질의와 파일 변환이 자유롭지 못하다는 단점을 갖는다.

이 논문에서는 생물학 연구실 수준에서 시퀀싱 실험을 통해 얻어지고 공개용 데이타베이스로부터 수집된 염기 서열 데이타의 포맷을 자유롭게 변환하여 생물학자가 원하는 필드를 추출 및 서열 편집한 후 데이타베이스에 저장 및 검색할 수 있는 서열 정보관리시스템을 제시한다. 제안하는 서열정보관리시스템은 XML 마크업 언어인 BSML을 서열 데이타 표현을 위한 공통 포맷으로 사용하고 복잡하고 계층적인 서열관련 생물학적 지식 및 서열 데이타를 표현한다. 새로운 서열의 디자인과 시퀀싱된 서열의 편집 및 검증을 위한 서열 연산을 제시하며 이를 통해 서열 데이타의 일치성을 유지한다. 또한 서열의 저장 관리는 동일한 DNA조각에 대한 서열 변경 정보를 유지하기 위해 서열 버전을 정의하고 이를 관리할 수 있는 메커니즘의 제시와 이를 능동 데이타베이스의 능동 트리거 규칙[11]을 이용하여 자동적으로 실행할 수 있음을 보여준다.

이 논문의 구성은 다음과 같다. 2절에서는 시퀀싱된 염기 서열 처리 과정에 대한 개요와 현재 사용되는 서열저장관리시스템들의 서열 저장 기법과 서열 파일 포맷을 분석한다. 3절에서는 제안하는 서열정보시스템의 구조에 대해 기술한다. 4절과 5절에서는 버전서열 관리와 서열 포맷 변환기에 대해 상세히 기술한다. 6절에서는 시스템의 구현결과 및 기존 시스템과 비교평가하고 7절에서 결론을 맺는다.

2. 관련연구

이 절에서는 생물학적 배경지식으로 염기 서열 시퀀싱 실험을 통하여 얻어진 염기 서열 젤 이미지 파일로부터 일치된 서열을 얻기까지의 과정을 소개한다. 또한 시퀀싱된 데이타 처리 및 서열 관리에 관련된 연구를 미국의 NCBI와 영국의 Sanger Center를 중심으로 소개한다.

2.1 생물학적 배경 : 시퀀싱 실험으로 생산된 서열 데이타 처리 과정

ABI 370 또는 377 시퀀싱 기계로부터 얻어진 서열에 대한 실험 데이타는 다음과 같은 일련의 과정을 거쳐서 양질의 일치서열(consensus sequence)을 얻는다. 아래 그림 1은 Sanger Center에서 shotgun 시퀀싱으로 얻어진 서열 데이타 처리 과정[8,18]을 기술한다.

• 시퀀싱 젤 이미지 처리

이 과정은 젤 이미지로부터 젤에 로드된 실험 샘플에 일치하는 DNA 서열과 젤 이미지에서 각 베이스의 추적경로(lane tracking)를 찾아내고 염기 베이스 결정(base calling)을 정확하게 처리하기 위한 일련의 과정이다. 결정된 염기 베이스 정보를 포함한 SCF(Standard Comressed File Format From DNA Sequencing Instrument) 형식의 추적(trace) 파일이 생성된다.

• 서열 전처리

이 과정은 서열 어셈블리를 위한 준비과정이다. 이 과정에서는 시퀀싱의 인공산물(벡터 서열, 바이러스 및 숙주 오염물질)을 식별 및 제거하고 서열의 중요한 특징을 기술한다. 고유의 실험 파일이 생성된다.

• DNA 조각 서열 어셈블리

중첩된 DNA을 연결하여 일치된 서열을 생성하는 과정이다. 실험파일을 텍스트 기반의 CAF(Common Assembly Format)파일 형식으로 변환한다. 서열 어셈블러, 어셈블 후처리, 어셈블리 편집기 모듈로 구성된다. Sanger Center는 어셈블리를 위한 Phrap과 편집을 위한 GAP4 프로그램을 통합하였다.

• 편집

어셈블리 과정 동안 중첩된 컨티그가 누락되거나 어셈블리 후에 새롭게 생겨난 컨티그를 식별하고 컨티그의 종단을 편집한다. 또한 일치서열의 품질을 하락시키는 부가적인 서열 데이타를 편집함으로써 일치 서열 안에 있는 모든 모호한 베이스를 결정한다.

• 품질 관리 및 평가

품질 관리(Quality Control) 및 평가(Assesment)는 하나의 샘플에 대한 시퀀싱 프로젝트가 끝나게 될 때 품질의 수준을 검사한다.

2.2 서열정보관리시스템

전세계적으로 염기 서열을 저장 관리하는 기관으로는 영국의 Sanger Center와 미국의 NCBI가 대표적이다. 이 절에서는 이러한 생명 정보 연구 기관에서 염기 서열 관리를 위해 이용하는 데이타베이스와 관련시스템에 대해서 소개한다.

Sanger 센터는 서열분석 프로그램으로 유사성 검색을 위한 BLAST[19,20]와 유전자를 발굴을 위한 GeneFinding을 이용한다. 시퀀싱된 서열을 관리 및 분석하기 위한 소프트웨어로는 Staden Package[14,16-18]가 있다.

Staden Package는 시퀀싱된 실험 파일에 대한 뷰어인 Trev, 참조 데이타와 추적 데이타의 돌연변이 정보를 보여주는 trace_diff, 서열의 어셈블리 및 어셈블리 후 컨티그를 편집할 수 있는 GAP4, 추적 데이타로부터 조각 어셈블리를 할 수 있도록 데이타를 준비하는 Pregap4, 어셈블리 후 얻어지는 일치 서열에 대한 유사성 검색 및 연산을 제공하는 Spin으로 구성된다.

Standen Package는 서열 파일에 포함된 서열 데이타에 대한 분석을 지원하지만 분석된 서열 데이타를 데이타베이스에 저장 및 검색을 지원하지 않는다.

NCBI[13]는 DNA 서열 데이타베이스인 GenBank[4, 8,15], EST서열 데이타베이스 dbEST, 단백질 분자 구조 모델링 데이타베이스 MMDB, 인간 유전자 카탈로그

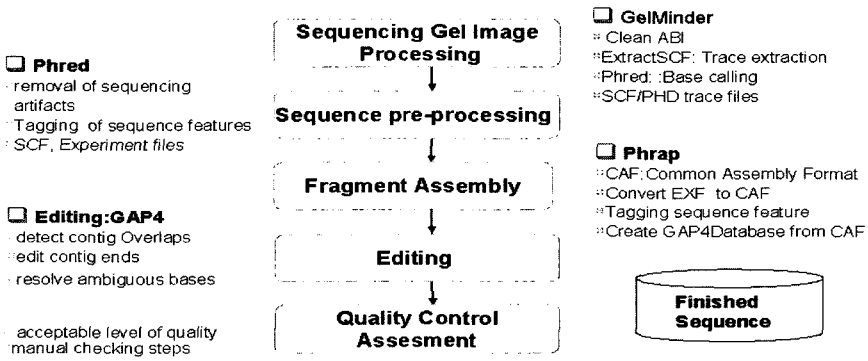

그림 1 시퀀싱 서열 데이타 처리 과정

와 유전적 변이에 대한 데이타베이스 OMIN, 문헌정보 데이타베이스 PUBMED를 유지하고 있다. 이러한 이질적인 생명 데이타를 통합하기 위해 데이타 사이에 링크가 연결되어 있다.

NCBI는 1990년부터 NCBI의 공통 포맷[4,21,22]으로 ASN.1 포맷을 이용하며 ASN.1은 단순한 파일포맷이기보다는 생물학적 데이타를 표현하는 데이타 모델로서 이용된다. ASN.1을 이용하여 서열, 물리적인 지도, 계통분류 정보, 분자정보와 문헌정보 등을 표현하고 이러한 정보를 포함한다. ASN.1의 사용을 위해서는 데이타를 파싱하고 인코딩하기 때문에 ASN.1메세지를 위한 전용 파서가 요구되고 데이타에 대한 질의를 지원하지 않으며 확장성이 없다는 단점을 갖는다. 현재는 염기 서열 데이타를 XML 포맷으로 표현한 BioML(Biopolymer Markup Language)을 이용한다.

Genbank는 초기부터 지금까지 데이타베이스의 서열 레코드를 관리하기 위해 서열 레코드의 식별자[4,18]를 변경하며 사용하고 있다. 초기 단계에는 생명체의 이름과 유전자 이름을 혼합하여 사용하였다. 그러나 시간이 지남에 따라 유전적 Locus 이름의 변경으로 인해서 식별자가 불안정해졌다. 따라서 현재는 실험자들이 NCBI에 서열을 제출할 때 Accession Number를 할당받도록 하고 서열 레코드의 식별자로 Accession Number를 사용한다. Accession Number는 Locus 이름보다 안정적이나 Accession으로 검색 시 변경된 서열에 대한 검색을 할 수 없었다. 그래서 서열 레코드의 갱신 시 변경된 서열 데이타를 식별하기 위해 GI(GenInfo Identifier)를 이용한다. GI는 서열이 GenBank의 ID(integrate DB) 통합 데이타베이스에 로딩될 때 서열에 할당되며 특정 서열이 갱신되면 동일한 Accession number에 GI를 새롭게 할당받는다. 이렇게 함으로써 서열 데이타의 변경에 대한 이력을 관리할 수 있다.

2.3 서열 데이타 포맷

공개용 서열 데이타베이스에서 배포되는 염기 파일은 주로 각 데이타베이스마다 고유한 포맷을 갖는 플랫파일이다. 이러한 플랫파일은 GenBank[15], EMBL[10], Stanford 대학의 Intelligenetics Sequence Format[23]과 Genetic Computer Group의 GCG 포맷[23]이 일반적이다. DNA서열의 유사성 분석을 위해서는 Pearson에 의해서 제시된 FASTA[24]가 가장 많이 사용되며 각 서열 분석 프로그램마다 서로 다른 포맷을 사용한다. 사용되는 플랫파일의 문제점은 1) DNA 서열의 분석보다는 서열 데이타의 배포에 초점이 맞추어져 있어 특정 시점에 따라 포맷이 다름, 2) 필드와 필드의 값 구분이 불분명하여 복잡한 공백처리가 요구됨, 3) 필드의 값이 모호하고 필드의 의미와 데이타 타입이 불일치하다는 것이다. 기술된 문제점으로 인해 분석을 위한 플랫파일의 사용은 자동프로그램보다는 생물학자가 개입된 데이타 추출 작업이 요구되며 플랫 파일 파싱시 많은 시간이 소모된다.

서로 다른 플랫 파일 간의 포맷 변환은 변환하려는 파일간의 일대일 매핑 관계에 따른 파서를 구현한다. 즉, NCBI 에서 그들의 데이타를 다른 포맷으로 변환할 때 NCBI 포맷과 변환하려는 파일 포맷마다 매핑 관계를 분석하여 파서를 구현하여야 한다. 따라서 다음과 같은 단점이 존재한다 1) 변환하려는 두 개의 포맷마다 하나의 파서를 구현해야한다 2) 플랫 폼이 바뀌었거나 소스 데이타의 포맷이 바뀌었을 때 거의 재활용을 못하고 매핑 정보를 수정하고 그에 따라서 파서를 다시 구현하여야 한다. 생물학자가 자유롭게 서열 포맷으로 필드 추출하고 포맷 변환을 위해 재활용 성이 높은 포맷 변환 기술이 필요하다.

따라서 서열 데이타 및 생물분자의 표현과 교환을 위해 XML을 적용이 필수적이다. 염기 및 단백질 정보에 대한 XML을 적용한 마크업 언어로는 BSML(Bioinformatic Sequence Markup Language)[12], BioML[25] 등이 존재한다.

Spitzner에 의해 제시된 BSML[12]은 97년부터 NHGRI(National Human Genome Research Institute)에서 유전체 연구 정보를 교환하기 위해 개발된 XML기반 생물 서열 마크업언어이다. BSML은 DNA, RNA 및 단백질 서열에 대한 서열 데이타, 문헌 및 관련된 유전자 발현정보를 포함한다. 유전체, 염색체, 조절부위, 유전자, 전사 등과 같은 다양한 생물학적 계층 수준의 서열에 대한 정보를 표현할 수 있고 가시화를 위한 부분을 정의할 수 있다. BSML을 기반으로한 Genomic WorkSpace[26] 소프트웨어는 GenBank같은 공용 데이타베이스로부터 유전체 서열을 검색하고 BSML 문서로 변환하여 서열 및 주석정보를 편집하고 BSML 문서로 저장 및 검색을 지원하는 소프트웨어이다.

이외에도 GenBank, EMBL, DDBJ, Swiss-Prot같은 데이타베이스에서도 NCBI_BLAST, Clustal multiple alignment와 같은 분석 도구들 사이에서 데이타 변환을 위해 BSML 변화기를 개발하여 이용한다. 또한 XML언어가 갖는 다른 시스템간의 데이타 교환과 데이타 공유를 위한 응용 프로그램 개발을 위한 공통 인터페이스 및 질의 언어와 같은 개방형 프레임워크을 제공한다는 커다란 장점은 계속해서 생명정보학 분야에서 XML의 활용이 가속화 될 것이다.

3. 서열정보관리시스템구조

이 절에서는 염기서열정보관리시스템[7]의 구조에 대

해서 상세히 설명한다. 3.1절에서는 서열 데이타를 저장 관리하는 서열편집기, 서열포맷변환기[6], 저장 관리기[2,3]의 구조에 대해 설명한다. 3.2절에서는 3.1절에 제시된 구성 요소 중 서열 편집기의 서열 연산처리기에서 수행되는 서열연산에 대해 설명한다.

3.1 시스템 구조

서열정보관리시스템은 시퀀싱 실험을 얻어진 일치 서열을 편집 및 검증하고 이러한 파일을 데이타베이스에 저장관리 한다. XML을 기반하여 서열 데이타를 표현 및 연산하며 그림 2처럼 서열 편집 관리기와 서열 저장 관리기, 서열 포맷 변환기로 구성된다.

• 서열 편집 관리기

서열 편집 관리기는 생물학자에게 서열 및 주석정보를 생성, 수정과 삭제 연산을 통해 편집할 수 있는 기능을 제공한다. 이를 위해 서열의 베이스 구성에 대한 기본적인 통계정보 제공과 베이스 수정 및 서열에 대한 분석을 위한 연산을 수행한다. 포맷변환기를 통해서 변환된 XML 파일을 서열 뷰어 모듈을 통해서 서열을 전시하고 서열에 대한 편집 연산을 수행하도록 한다. 서열 편집 관리기는 서열 뷰어, 서열 연산처리기, 서열 주석기로 구성된다.

- 서열 뷰어는 대화상자를 통하여 실험파일로부터 선택되거나 데이타베이스에서 검색된 서열을 전시한다. 서열의 A, T, G, C베이스 구성을 계산하여 함께 전시된다. 사용자는 서열 데이타에 대해 텍스트 편집을 할 수 없으며 염기베이스를 추가, 삭제 및 수정을 위해서는 서열 연산을 이용해야 한다.
- 서열 연산처리기는 선택된 서열에 대해 서열의 배열이나 순서를 변경할 수 있는 연산을 수행한다. 서열 연산 처리기에서는 염기 구성(Base Composition), 부분 서열 검색(Search Subsequence), 보수 서열 생성(*Complement Sequence*), T와 U 염기 베이스 변환(convert t into u), 회전(rotate)연산과 버전(version) 연산을 수행한다.
- 서열 주석기는 현재 편집중인 서열 데이타에 필요한 주석정보를 추가 및 갱신한 후 데이타베이스에 추가된 주석정보를 함께 저장해주는 기능이다. 현재 주석데이타는 단백질 ID, 프로젝트 ID, 서열타입, 소스생명체, 실험자 이름 및 서열 데이타에 대한 생물학적 정보와 인용문헌에 대한 정보이다.

• 서열 저장 관리기

서열 저장 관리기는 서열 저장기와 서열 정보검색기와 버전 관리기로 구성된다. 서열 버전 관리 모듈은 새로운 서열이 데이타베이스에 입력될 때 버전을 검사 및 생성한다. 서열 저장기는 새로운 서열 정보의 입력하고 서열 정보 검색기는 검색어를 입력받아 검색한다.

- 염기 서열 정보는 바이너리 타입으로 데이타베이스에 저장된다. 데이타베이스 테이블은 시퀀싱에 관련된 테이블 1개와 서열의 생물학적 기능 및 참고문헌 정보 같은 주석정보 관련 16개 테이블과 서열관련 3개의 테이블로 구성된다. 서열관련 테이블은 서열 테이블과 연산을 통해 유도된 서열을 저장하는 유도서열 테이블과 버전서열을 저장하는 테이블로 구성된다.
- 서열 정보 검색기는 서열 매칭에 의한 서열 정보 검색 및 주석 정보 검색 모듈이다.

서열 버전 관리기는 서열의 변경에 대한 이력정보를 관리하기 위하여 능동 트리거 규칙을 이용하여 서열 버전의 검출 및 생성하는 모듈이다. 그림 3은 서열 정보의

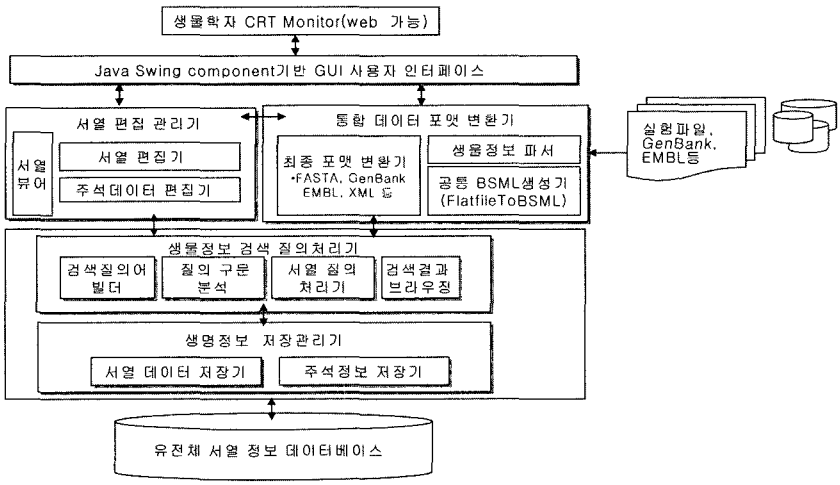

그림 2 서열정보관리시스템 구조

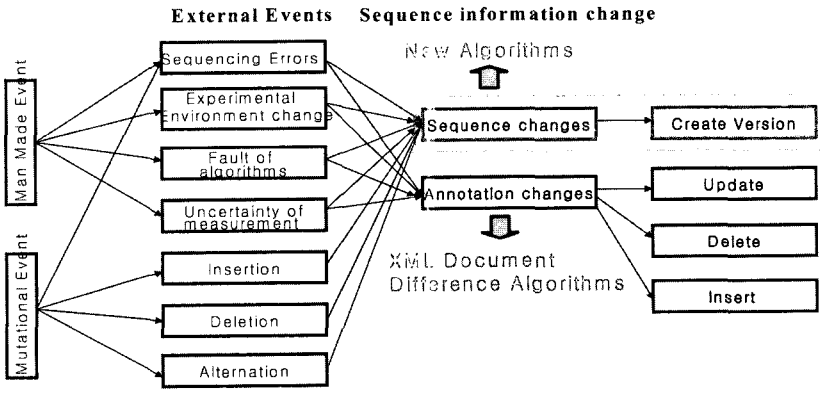

그림 3 서열 변경 요인 및 관리 기법

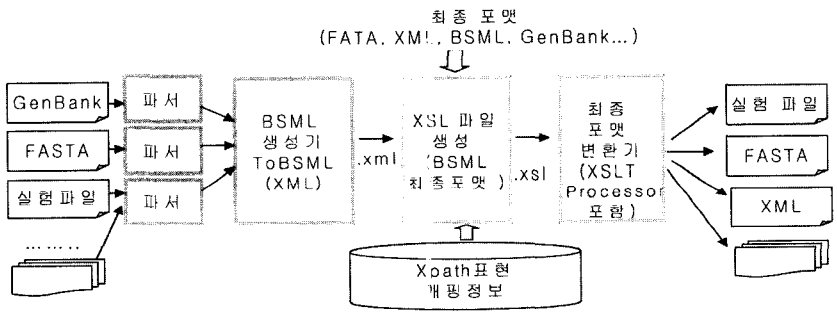

그림 4 서열 포맷 변환 메커니즘

변경 원인과 그에 따른 관리를 나타내고 이에 대한 상세한 설명은 4절에 기술한다.

• 서열 포맷 변환기

서열 포맷변환기는 이질적 염기 서열 포맷으로부터 필드를 자유롭게 추출하고 생물학자간의 데이타 교환을 위하여 XML을 공통 포맷 및 모델로써 이용한다. 그러므로 서로 다른 포맷들 간의 n:n 포맷 변환을 1:n 변환으로 감소한다. 또한 염기 서열 분석 및 관리 시 XML 질의언어, 데이타 모델과 저장관리 기술등과 같은 XML 기술을 적용할 수 있다. 특히 XML 기반 유전체 정보 표현을 위한 표준으로 이용되는 BSML버전 3.0의 DTD를 따르도록 XML 서열 데이타 포맷을 정의하였다. 생물학 실험실에서 시퀀싱 실험으로부터 얻어진 실험파일 또는 공개용 데이타베이스의 플랫 파일을 BSML로 변환하고 변환된 BSML을 FASTA, GenBank, EMBL, BSML포맷으로 변환을 지원한다. 서열 포맷 변환기는 그림 4와 같이 BSML 문서 변환기(EXFileToBSML), 매핑 정보 저장기(MappingInfo)와 포맷생성기 (GenTargetFormat) 모듈로 구성된다. 그림 4의 세부 모듈에 대한 상세한 설명은 아래와 같다.

- BSML 문서 변환기 : 실험파일 및 공개용 유전체 서열 파일을 파싱하여 서열 및 관련 데이타를 추출하고 이를 XML 기반의 BSML 문서로 작성하는 모듈이다.
- 매핑정보 저장기 : BSML문서의 DTD와 변환하려는 포맷들간의 매핑 정보를 XPath로 작성하고 변환하려는 포맷에 대한 XSL파일을 기술한다. 이러한 정보는 포맷 변환을 위한 메타데이타로 데이타베이스에 저장된다.
- 포맷 생성기 : 사용자의 포맷변환 요구가 있을 때 BSML문서에 대해서 변환하려는 포맷의 XSL파일을 데이타베이스로부터 검색하여 XML문서에 적용하여 파일 변환을 수행하는 모듈이다.

3.2 서열 연산

이 절에서는 서열 편집을 위해 서열 연산처리기에서 실행하는 염기 구성(Base Composition), 부분 서열 검색(Search Subsequence), 보수 서열 생성(Complement Sequence), T와 U 염기 베이스 변환(convert t into u), 회전(rotate)연산을 예를 들어 상세히 설명한다. ECAE198 염기 서열의 일부를 연산에 대한 예제로 사용한다.

• 염기 구성

서열을 구성하는 염기의 구성 비율을 계산하는 연산이다. 전체 서열 길이에 해당 염기 베이스의 출현빈도를 퍼센트로 나타낸다. 그림 5의 (a)는 DNA 서열 ECAE-

198의 염기 베이스 구성 연산의 결과이다.

• 부분 서열 검색

이 연산은 현재 편집되고 있는 서열에서 특정 부분의 시작과 끝부분 위치를 지정하여 새로운 서열 엔트리로 생성한다. 그림 5의 (b)는 ECAE198 서열에 대해 6번부터 25번까지의 부분 서열 검색 연산의 수행 결과로 얻은 부분 염기 서열을 보여준다.

• 보수 서열 생성

이 연산은 현재 편집 중 인 DNA서열의 상보 서열 생성을 수행하는 연산한다. 우선적으로 A-T, G-C의 염기베이스를 변환하여 서열의 역(reverse) 서열을 만든다. 이 역 서열은 5'에서 3'의 서열 방향을 갖으므로 다시 반대 방향으로 서열을 읽어서 3'에서 5'방향의 상보 서열을 생성한다. 그림 5의 (c)는 ECAE198 서열에 대한 상보서열 연산결과를 보여준다.

• T와 U 염기 베이스 변환

DNA가 RNA로 전사될 때 DNA의 염기 중 티민(T)이 RNA에의 우라실(U)로 변경된다. 이 상호 변환 연산은 DNA의 염기 T를 RNA의 염기 U로 변환해 주는 연산이다. 따라서 이 연산을 통해 DNA염기에 대한 RNA 염기 서열을 얻을 수 있다. 아래 그림 5의 (d)는 DNA 서열 ECAE198의 상호전환 연산을 수행한 결과이다.

• 회전 연산

회전 연산은 서열이 시작되는 위치를 새롭게 지정하기 위한 연산이다. 이 연산은 서열의 염기 구성은 변함없고 서열을 구성하는 염기 서열의 순서가 변경된다. 이러한 회전 연산은 생물학자에게 어셈블리 후 일치 서열을 편집할 때 shotgun 시퀀싱 동안에 반복되는 부분이 잘못 어셈블리 되거나 어셈블리 상의 에러를 찾을 수 있도록 도와준다.

회전 연산은 사용자가 지정한 위치를 입력으로 받아 현재 편집 중인 서열의 지정된 위치에서 시작하여 앞부분을 뒤로 보내는 회전 작업을 실행하여 준다. 이 연산을 통하여 새롭게 얻어진 서열도 역시 데이타베이스에 저장된다. 아래 그림 6은 DNA 서열 ECAE198의 49번째 베이스 염기를 시작으로 회전 서열을 구하는 예제이다.

ECAE198 DNA서열의 원본 서열:(1~60)

```
0123456789012345678901234567890123456789012345678901234567890123456
gtagaaagcaccgacaatactcctggcatgggcgttaaagctcacaggatggagattctt

10        20        30        40        50        60
```

(a) ECAE198의 염기서열 구성 결과

■ A:30% C:21.67% G:26.66% T: 21.67%

(b) Set Subsequence 연산 결과

■ aagcaccgacaatactcctg

6 10 25

(c) 보수 서열(complement sequence) 연산 결과

■ aagaatctccatcctgtgagctttaacgcccatgccaggagtattgtcggtgctttctac

10 20 30 40 50 60

(d) 상호 전환(convert t into u)

■ guagaaagcaccgacaauacuccuggcaugggcguuaaagcucacaggauggagauucuu

10 20 30 40 50 60

그림 5 ECAE198 서열에 대한 연산 예

ECAE198 DNA서열의 원본 서열:(1~60)

```
gtagaaagcaccgacaatactcctggcatgggcgttaaagctcacaggatggagattctt

10        20        30        40        50        60
```

(a) 회전 서열(Rotate sequence)

atggagattcttgtagaaagcaccgacaatactcctggcatgggcgttaaagctcacagg

10 20 30 40 50 60

그림 6 회전 서열 연산 예

4. 능동 트리거를 이용한 서열 버전 관리

동일한 DNA 조각에 대해서 시퀀싱 실험을 통해 염기 서열을 얻을 때 실험을 할 때마다 서로 다른 염기 서열 구성을 보일 수 있다. 이러한 동일한 염기 서열 조각에 대한 서열 변경 원인은 생물학적 측면에서 진화적인 돌연변이 및 변이가 될 수 있다. 인위적인 측면에서는 시퀀싱 실험에서 환경 요인의 변화 또는 시퀀싱 기계로부터 일치된 서열을 얻기까지 시퀀싱 과정에 관여하는 프로그램들의 알고리즘 상의 오류 같은 요인에 기인한다. 그러나 현재까지는 이러한 변경된 서열 중 어떠한 서열이 정확한 서열인지 불확실하며 그 원인 또한 확실하지 않다. 따라서 이 절에서는 서열 저장관리기 모듈 중 서열의 변경정보를 관리하는 서열 버전 관리 기법에 대해서 상세히 소개한다.

서열 버전 관리는 원본 서열과 새로운 서열과 차이점을 발견하는 서열 버전 검출 기법과 새로운 서열 버전을 데이타베이스에 저장하기 위한 서열 버전 생성 연산을 포함한다. 버전관리연산은 새로운 서열이 입력될 때 자동적으로 처리하기 위해 능동데이타베이스의 트리거 규칙을 이용한다. 최종적으로 버전 정보를 XML로 표현하여 서열에 대한 이력정보를 서로 다른 시스템과 교환할 수 있도록 한다.

4.1 서열 버전 정의

동일한 DNA 조각에 대해 시퀀싱 실험때 마다 얻어지는 서열 구성이 변경된 서열을 서열 버전이라고 부르고 이를 다음과 같이 정의하였다.

한번의 시퀀싱 실험에서 DNA 샘플로부터 얻어진 염기 서열의 집합을 SS라 하면, SS는 염기 서열의 유한 집합이며 SS={S_1, S_2,…, S_l}로 나타낸다. 이때 l은 서열 총 개수이며 l=|SS|이고 SS에 포함된 서열 S_i가 $S_i \in SS$ 일 때 염기 서열은 다음과 같이 정의된다.

정의 1. 임의의 염기 서열 Si 는 Si={a_1, $a_2 \cdots a_k \cdots a_n$}이고 a_k는 {A, G, C, T} 염기 베이스 중 하나로 표현될 수 있는 문자이고 n은 서열의 길이로 n=|S_i|이 된다. 단 S_i는 서열을 유일하게 식별할 수 있는 서열 식별자를 가지며 서열 식별자는 Sid(S_i)로 표현된다. 서열 식별자는 Sid(S_i)는 염색체상의 위치와 Locus id를 포함하며 서열을 유일하게 식별할 수 있는 값이다.

정의 2. 동일한 시퀀싱 실험방식으로 다시 동일한 DNA 샘플 조각으로부터 얻어진 시퀀싱된 서열의 집합을 SS′, 서열 Si'={a_1', $a_2' \cdots a_j' \cdots a_m'$}와 $Sid(S_i')$가 주어지면, S_i와 서열 S_i'사이는 S_i'-Version(S_i)가 성립하고 S_i'를 S_i의 버전서열이라 정의한다. 서열 S_i과 서열 버전 S_i'는 다음의 조건을 만족해야 한다.

즉, $Sid(S_i)$-$Sid(S_i')$, Length(S)=Length(S_i') 또는 Length(S)≠Length(S_i'), Order(S)≠Order(S_i')이고 Composition(S)≠Composition(S_i')이다. 이때 D는 서열 데이타베이스이고 $SS \subseteq D$ 이고 $S_i' \notin D$이다.

정의 3. 원본 서열 S_i 의 서열 버전 S_i'={a_1', $a_2' \cdots a_j' \cdots a_m'$}는 원본 서열과 local alignment 수행 후 차이(Difference : Diff)만을 저장 및 표현한다. Diff는 Diff(S_i', S_i)={a_1', $a_2' \cdots a_j' \cdots a_k'$} 2차원 점의 집합이다. 이때 0 < k < m (m=max(length(S_i), length(S_i'))이고 a_j'=(x, y)로 구성된 2차원 점이다. x는 원본 서열 Si와 다른 염기 베이스의 서열 내의 위치이다. y는 원본 서열 Si와 다른 염기 베이스를 bit로 매핑한 값이다. 즉, 하나의 염기 베이스는 2bit로 표현하고 염기 서열의 일치 관계를 연산할 때는 네 개의 염기베이스를 1byte로 표현하여 이용한다. 즉, 00=A, 01=C, 10=G, 11=T로 나타낸다. 검색을 빠르게 하기 위해 서열에 대한 저장은 편집연산에 의해서 유도된 서열과 버전서열과 원본 서열을 서로 다른 테이블로 분리하여 저장한다. 서열 데이타는 데이타베이스에 저장 시 variable character, binary와 blob 타입 중의 하나로 저장할 수 있다. binary와 blob 타입으로 저장할 경우 SQL에서 지원하는 스트링 매치 기능을 이용할 수 없지만 빠른 검색 속도를 보인다. variable character 타입으로 저장할 경우애는 데이타 베이스관리시스템 마다 가변 문자 타입의 최대 허용길이가 다르므로 이를 고려해야 한다. 즉, MSSQL 서버는 8K까지 ORALCE은 최대 4k까지 지원한다. 여기에서는 서열 데이타를 가변 문자 타입으로 저장하여 SQL의 스트링 매치 기능을 이용한다.

4.2 능동 트리거를 이용한 서열 버전 검출 및 생성 기법

이 절에서는 앞 절 4.1에서 설명한 바와 같이 서열이 저장되었을 경우 새로운 서열이 입력될 때 능동 데이타베이스의 트리거를 이용하여 자동적으로 서열 버전을 검출하고 서열 버전을 생성하는 알고리즘에 대해 설명한다.

• 버전 검출

트리거를 이용한 새로운 서열 버전에 대한 검출 및 생성은 다음과 같은 과정을 통해서 이루어진다.

서열 입력단계 : 새롭게 서열 데이타 Si'가 데이타베이스에 입력된다. 이 단계에서는 서열 입력이라는 데이타베이스의 insertion 연산이 트리거의 이벤트(Event)로 발생한다.

Source finding 단계 : Si'가 특정 서열의 버전이 될 가능성을 조사한다.

데이타베이스의 서열 테이블에 존재하는 서열 중 입력된 서열 Si'과 동일한 서열 식별자 $Sid(Si)$ = $Sid(Si')$를 갖는 원본 서열 Si을 찾아낸다. 이 단계는 서열 입력단계에서 이벤트가 발생된 트리거에 대해 동일한

식별자를 갖는 서열을 찾아내기 위한 조건(Condition)을 평가한다. 동일한 식별자를 갖는 원본서열 *Si*을 찾은 경우에는 다음 단계인 Change detection단계로 넘어가고 그렇지 않은 경우에는 *Si'*는 새로운 서열로 데이타베이스에 입력되는 트리거의 액션이 발생한다.

Change detection단계 : 서열 테이블에 동일한 서열 식별자를 갖는 원본 서열 *Si*가 존재하면 *Si'*가 원본 서열 *Si*와 차이가 있는지 검사한다. 즉, *Si'*과 *Si* 사이에 정의 2를 만족하는지 검사한다. 이 단계 역시 서열 입력 단계에서 발생된 트리거의 이벤트에 대해 원본 서열과 입력서열과의 차이 여부를 조사하는 트리거의 조건(Condition)절에 해당된다.

i) 만일 정의 2를 만족하지 않으면 *Si'*는 원본 서열 *Si*와 차이가 존재하지 않고 *Si*의 버전서열이 될 수 없다. 따라서 이 원본 서열 Si에 대해 서열 주석이 변경되었으면 주석을 갱신 할 수 있도록 한다. Change detection을 위한 트리거 조건 평가에 대한 주석 데이타 갱신이나 아무런 연산도 발생하지 않는 트리거 액션이 발생한다.

ii) 정의 2를 만족하면, *Si'*는 원본 서열 *Si*과는 차이가 존재하지만 *Si* 가지는 버전서열들과 차이가 있는지 여부를 알 수 없고 *Si'*는 다음 단계인 iteration 단계로 넘어간다. *Si'*이 정의 2를 만족하는 지의 평가하기 위해서는 두 서열 *Si* 와 *Si'* 간에 서열 데이타의 변경이 있는지 여부를 서열의 배열 순서Order(*Si*)≠Order(*Si'*) 또는 베이스 구성Composition(*Si*)≠Composition(*S'i'*)을 검사하여 결정한다. Binary로 표현된 서열에 대해 XOR 연산을 통해서 베이스 구성이 일치하는지 알 수 있다. *Si'*이 정의 2를 만족하면 *Si*과 *Si'* 사이에 Diff(*Si'*, *Si*)를 생성한다.

Iteration 단계 : *Si* 가지는 모든 버전서열과 *Si'*사이에 변경이 존재하는지 검사하여 최종적으로 버전여부를 판단한다. 정의 3에서 서열 버전은 원본 서열에 대해 Diff 차이 정보로 표현 및 저장되므로, 이 단계에서는 *Si'*의 Diff(*Si'*, *Si*)와 *Si*가 가지는 모든 버전들과의 Diff 정보를 비교해서 결정된다. Diff'가 *Si*의 서열 버전과 일치하지 않으면 *Si'* 는 *Si* 의 새로운 버전서열이 된다.

CreateVersion 단계 : *Si'*는 *Si*의 새 서열 버전으로써 서열 버전 테이블(VersionSeq)에 입력되는 insert 연산과 서열테이블(Sequence)에 *Si*의 버전번호가 1증가되는 갱신 연산되는 트리거 액션이 발생된다. 이 때 *Si'*에 대한 버전정보는 정의 3에 있는 Diff정보가 데이타베이스에 저장된다.

그림 7은 서열 버전 검출을 위해 SQL(Structured Query Language)을 이용한 트리거에 대한 정의이다.

4.3 서열 버전정보의 XML 표현

서열 정보와 함께 서열 버전 정보를 교환하고 분석 프로그램에 제공하기 위해 생명정보데이타 표현을 위한 defacto 표준인 BSML을 이용하여 버전 정보를 표현한다. BSML3.0 버전의 DTD에 양립하도록 BSML의 엘리먼트 중 Segment-set과 Segment를 이용하여 원본 서열과 서열 버전 사이의 서열 차이를 표현하고 원본 서열이 갖는 다른 주석 및 실험정보를 공유하도록 한다. Segment-set은 정의되는 서열과 다른 서열의 특정 부분이나 조각과의 관계를 정의하는 엘리먼트이고 Segment는 Segment-엘리먼트에서 정의된 관계에 대하여 서열의 부분이나 조각을 나타내는 엘리먼트이다. 여기서 Segment set은 원본 서열을 나타내고 Segment는 서열 버전과 원본 서열 사이의 다른 부분인 Diff 를 나타낸다.

그림 8의 예제처럼 서열 SQ2002060300001에 대한 정보를 BSML의 DTD에 맞게 Sequence 엘리먼트에 정의한다. Sequence 엘리먼트에 대한 상세한 설명은 다음 5절에서 설명한다. Sequence 엘리먼트는 서열SQ-2002060300001에 대한 가장 최신 정보를 포함한 최근의 버전 2에 대해 기술한다. Segment-set 엘리먼트의 id 속성은 원본 서열의 식별자를 나타낸다. seg 엘리먼트의 id 속성 값을 식별자로 갖는 서열과 segment 엘리먼트로 정의될 조각 서열과의 관계를 나타낸다. 보통 BSML은 segment, equivalent, copy, translated, expressed, gapped, aligned와 같은 관계를 정의해서 사용한다. 여기서는 원본 서열과 서열 버전 관계 중 Segment-set은 원본 서열을 나타내므로 origin으로 표시한다.

Segment 엘리먼트는 Segment-set 엘리먼트에 기술된 원본 서열과 서열 버전의 차이를 기술한다. 특히, 원본 서열이 버전서열과 일치하지 않은 서열 부분의 정보를 나타내며 이러한 정보 표현을 위한 속성을 갖는다. Segment 엘리먼트의 id 속성은 서열 버전 중 원본 서열과 변경이 있는 원본서열 부분에 대한 식별자이다. seg role은 차이를 나타내는 부분의 역할로서 여기서는 "difference region"값을 갖는다. seg id는 현재 segment 엘리먼트에서 나타내는 버전서열의 식별자를 나타낸다. seg start와 seg end는 원본 서열과 서열 버전과의 차이를 나태는 부분의 서열 버전상의 위치를 나타낸다. Attribute 엘리먼트는 segment 엘리먼트에 관련된 정보 중 DTD에 포함되지 않은 정보를 사용자가 정의해서 사용할 수 있는 엘리먼트로서 name 속성과 name 속성에 대한 데이타 값을 나타내는 content 속성을 갖는다. name 속성의 값 중 type은 변경이 있는 부분이 어떠한 변이를 나타내는지를 표현한다. 여기서는 원본서열에 대해 버전서열의 1001번부터 1003번 염기 베이스가 생물학적으로 insertion 변이가 일어나고 3001번에서 점 변

```
CREATE OR REPLACE TRIGGER DetectingVerionSequence
AFTER INSERT ON TempSeq //A Triger event occurs when a new sequence is inserted
REFERENCING
  OLD ROW as oldrow
  NEW ROW as newrow
FOR EACH ROW
DECLARE  // Declaration variables
  Maxvid INTEGER;Duple INTEGER;Seqtrue INTEGER;Vertrue INTEGER;Firstver INTEGER;
BEGIN
  SELECT COUNT(*) INTO Duple FROM Sequence
  WHERE Sid = newrow.Sid AND Sseq= newrow.Sseq;
IF Duple=0 THEN //A condition of the trigger to check whether there are duplicated identifier
  SELECT COUNT(*) INTO Seqtrue FROM Sequence   WHERE Sid = newrow.Sid;
IF Seqtrue = 0 THEN // A condition of the trigger to check the first insertion of the input sequence
    INSERT INTO Sequence VALUES  (newrow.Sid, newrow.SPid, newrow.Sseq,
    newrow.Slength, newrow.Stype, newrow.Sdate, newrow.Smachine, 0, 0, newrow.SsPos,
    newrow.SePos, newrow.SnumA, newrow.SnumT, newrow.SnumC, newrow.SnumG);
ELSE  // A condition of the trigger  to check whether the input sequence may be version or not
Varchar  Diff;
Diff=Difference(Newrow.Seq, OldRow.Seq)
SELECT COUNT(*) INTO Firstver FROM VersionSeq
WHERE VSid = newrow.Sid;
  IF Firstver = 0 THEN // A condition of the trigger to be the first version of a sequence
    INSERT INTO VersionSeq // Action of  the trigger to insert the first version of the sequence
    VALUES (newrow.Sid,1, newrow.SPid, Diff, newrow.Slength, newrow.Sdate);
   ELSE
     SELECT COUNT(*) INTO Vertrue FROM VersionSeq
     WHERE VSid = newrow.Sid AND VerDiff = Diff;
     IF Verture = 0 THEN // A Condition of the trigger to be a new version of the sequence
        SELECT MAX(Vid) INTO Maxvid FROM VersionSeq WHERE VSid= newrow.Sid;
INSERT INTO VersionSeq // Action of  the trigger to insert the first version
VALUES (newrow.Sid,Maxvid+1, newrow.SPid, Diff, newrow.Slength, newrow.Sdate);
    ELSE // A condition of the trigger to be the same base composition with existing   versions
       RAISE_APPLICATION_ERROR(-20000, ' WARNING! DUPLICATED VERSION!!!' );
END IF;
   END IF
ELSE // A condition of  the trigger  when a existing sequence is inserted into a table SEQUENCE
  RAISE_APPLICATION_ERROR(-20000,'WARNING! DUPLECATED ID AND SEQUENCE!');
END IF;
END;
```

그림 7 서열 버전 검출 및 생성을 위한 트리거

이 중 대치가 발생한 것을 의미한다. name 속성의 값 중 position과 content "1001"은 원본 서열과 서열 버전이 다른 부분의 위치를 나타내며 base와 "01"은 position에 정한 위치의 버전서열의 염기 베이스를 비트로 나타낸 것이다. 만일 원본 서열과 버전서열이 다른 부분이 하나의 염기 베이스만이 다르다면 seg start와 seg end가 갖게 된다.

5. XML기반 서열 포맷 변환 관리기

2.3절에서 소개한 내용처럼 생명정보학분야의 염기 서열 데이타 관련 내용을 표현하기 위한 데이타 포맷은 이질적이고 다양하다. 5.1절에서는 시스템에서 사용하고 있는 염기 서열 데이타 포맷으로 실험 파일 포맷과 FASTA 포맷, BASML 기반 XML포맷을 기술한다. 그

```
<Sequence id="SQ2002060300001.2" ic-acckey="AB003468" title="Cloning vector pAP3neo DNA"
 length="5350" topology="linear" representation="raw">
      <attribute name="version number" content="2"></Sequence>
<Segment-set id=" SQ2002060300001.1" seg-set-type="origin">
   <Segment title="Difference Region1" id=" SQ2002060300001.1.1" seg-role="version" seg-
     id="SQ2002060300001.2" seg-role="different region" seg-start="1001" seg-end="1002">
    <attribute name="type" conent="insert"/>
    <attribute name="position" content="1001">
    <attribute name="base" content="01">
    <attribute name="position" content="1002">
    <attribute name="base" content="00">
</Segment>
  <Segment title="Difference Region2" id="SQ2002060300001.1.2" seg-source-type="version" seg-id="
    SQ2002060300001.2" seg-role="segment" seg-start="3001" seg-end="3001" >
    <attribute name="type" content="substitution"/>
    <attribute name="position" content="3001" >
    <attribute name="base" content="01" >
 </Segment>
</Segment-set>
```

그림 8 BSML을 이용한 버전 정보 예

```
>SQ2002060300001|1|Cloning vector pAP3neo DNA, complete sequence |DNA| Cloning vector
pAP3neo |5350
GGTACCTTCTGAGGCGGAAAGAACCAGCCGGATCCCTCGAGGGATCCAGACATGATAAGATAC
ATTGTTCCGACCCTGCCGCTTACCGGATACCTG*
```

그림 9 FASTA 포맷 예

리고 5.2 절에서는 이러한 실험파일로부터 FASTA 포맷이나 다른 공용 데이타베이스에서 사용하는 포맷으로 변환을 위한 메커니즘을 상세히 설명한다.

5.1 염기 서열 데이타 포맷 정의

대표적인 염기 서열 데이타베이스인 GenBank와 EMBL에서 각자의 플랫 파일 포맷을 가지며 대표적인 서열유사성검색시스템인 BLAST에서는 FASTA라는 데이타 포맷을 사용하고 있다. 이 절에서는 제안 염기서열정보관리시스템에서 사용한 FASTA 포맷과 XML 파일을 기술한다.

• FASTA 포맷

FASTA 포맷은 서열 유사성 검색 및 다중 서열 정열 같은 서열 분석프로그램에서 사용하는 포맷이다. 하나의 서열 엔트리는 ">" 부터 시작하고 이것은 서열에 대한 주석 라인을 나타내다. "|"로 주석 필드간의 구분자로 사용한다. 주석라인의 다음 라인부터 서열 데이타가 온다. 그리고 "*"로 서열의 끝을 나타낸다. FASTA 포맷의 주석 라인에 서열 식별자, 서열 버전, DNA 분자 이름, 분자 종류, 생명체 및 서열 길이 필드 값으로 표현하고 다음 라인에는 서열 데이타를 나타낸다. 그림 9는 FASTA 포맷의 예이다.

• XML 포맷

BSML3.0을 기준으로 할 때 BSML문서는 Definition, Research, Display와 같은 상위 엘리먼트와 이들의 하위 엘리먼트로 구성된다. 서열 데이타는 Definitions 엘리먼트의 하위 엘리먼트로 기술되며 가시화에 필요한 정보는 display 엘리먼트에 기술될 수 있다. Definitions 엘리먼트는 염기 서열 정보를 기술하기 위한 하위 엘리먼트를 가지며 이러한 서열 정보 기술과 관련된 DTD는 그림 10과 같이 정의된다.

Sequences엘리먼트는 서열의 집합으로 Sequence 엘리먼트의 반복으로 이루어진다. 각 서열에 대한 정보는 Sequence 엘리먼트에 포함되며 하위 엘리먼트로 서열 데이타를 기술하는 Seq-data와 서열에 대한 참고문헌 및 유전자 정보를 기술하는 Feature-tables 엘리먼트를 포함한다. 또한 BSML의 각 엘리먼트는 해당 엘리먼트에 포함되지 않은 내용을 기술하기 위한 Attribute 엘리먼트를 가지며 이 Attribute 엘리먼트의 name과 content 속성의 쌍으로서 표현할 수 있다. 여기서는 서열이 갖는 총 버전의 수를 표현하기 위해 Attribute 엘리먼트를 다음과 같이 Sequence 엘리먼트의 하위 엘리먼트로 정의한다. <attribute name="version number" content="2">

서열 데이타는 Seq-dat와 Seq-data-import 엘리먼트에 의해 기술되는데 Seq-data-import는 다른 BSML 문서에 정의된 서열정보를 참조할 때 URL을 이용하여 서열정보에 대한 링크를 할 수 있다. Modification 엘리

```
<!ELEMENT Definitions(Attribute*, Genomes?, Sequences?, Isoforms?, Sets?,Tables?, Networks?) >
<!ELEMENT Sequences(Attribute*, (Sequence|Sequence-import)*, Segment- set*, Resource*, links) >
<!ELEMENT Sequence(Attribute*, Feature-tables?, (Seq-data|Seq-data-import)?, Numbering?,
Modification*, Resource*,links)>
<!ELEMENT Feature-table (Attribute*, (Reference|Feature|Digest-set)*, Resource*, links) >
<!ELEMENT Seq-data (#PCDATA) >
<!ELEMENT Seq-data-import(Attribute*) >
<!ELEMENT Modification(Attribute*,Resource*, links) >
<!ELEMENT Numbering EMPTY >
<!ELEMENT Segment-set(Attribute*, Segment+,Resource*, links) >
<!ELEMENT Segment(Attribute*, Resource*, links) >
```

그림 10 BSML DTD중 서열 정보 관련 엘리먼트

먼트는 서열의 베이스가 변경된 정보를 나타내기 위한 엘리먼트이다. Numbering 엘리먼트는 서열의 상대적인 위치를 나타낼 수 있다. 즉, 염색체 상에서의 clone의 위치와 clone 상에서의 서열의 위치를 숫자로서 표현할 수 있다.

5.2 XML적용 서열 파일 변환 메커니즘

이 절에서는 XML 기술에 기반하여 염기 서열 실험 파일로부터 Genbank, EMBL, FASTA와 BSML 형태로 변환하는 과정을 상세히 기술한다. 일반적으로 XML 문서는 문서의 구조를 나타내며 이 XML문서에 대한 가시화 정보를 XSL(Extensible Stylesheet Language) 파일에 정의하여 XML문서를 원하는 포맷으로 전시할 수 있다. 포맷 변환에서 이러한 XML문서의 포맷 변환 원리를 적용한다.

BSML의 DTD를 염기 서열 데이타의 공통 포맷을 위한 통합 포맷으로 활용한다. 우선적으로 BSML의 DTD와 변환하려는 대상 포맷인 GenBank, FASTA, EMBL 포맷들간의 매핑정보를 XPath로 작성하여 그림 11과 같이 데이타베이스의 테이블에 저장한다. 변환하려는 대상 파일은 BSML DTD 기반한 XML문서에 대한 스타일쉬트 파일로 나타낸다. 따라서 BSML문서로 표현된 공통 포맷에 작성된 매핑정보를 이용하여 변환하려는 대상파일로의 XSL파일을 생성하여 이를 BSML문서에 적용한다.

즉, BSML을 Genbank 파일로 변환을 위해서는 GenBank.xsl파일을 생성해야 하며 이것을 위해서 BSML 문서에서 XPath에 해당하는 엘리먼트를 테이블의 왼쪽에 위치한 GenBank 포맷의 필드로 인코딩한다.

파일변환에 대한 문서 요청이 있을 경우 실험파일을 BSML문서로 변환하고 변화하려는 대상 스타일 시트 파일인 XSL에 대한 경로를 BSML문서에 삽입한다. 데이타베이스로부터 해당 XSL파일을 검색하여 BSML 파

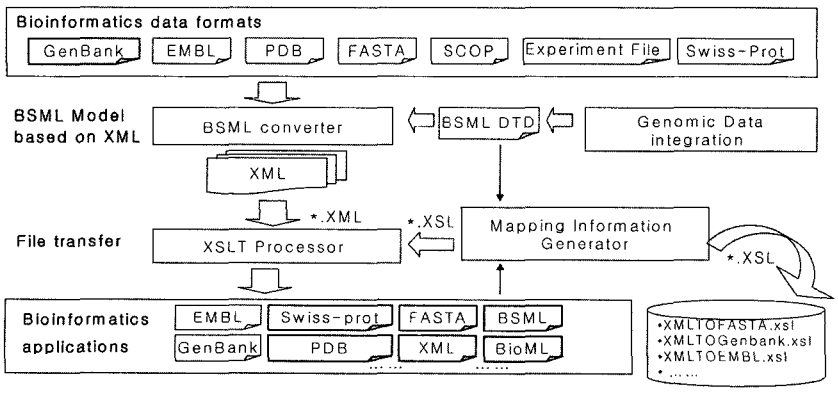

그림 11 BSML 기반 서열 포맷 변환 메커니즘

일과 XSL파일을 동일 디렉토리 상에 위치시키면 된다. 이와 같은 XML을 이용한 파일 변환 메커니즘은 그림 11에 나타낸다. EMBL과 GenBank는 염기 서열 데이타를 상호 교환하여 서열 데이타를 공유하기 때문에 플랫 파일 필드의 의미가 동일하다. 여기서는 그림 12처럼 GenBank포맷과 BSML 엘리먼트 간의 매핑정보를 나타낸다.

XML에 기반한 BSML 문서와 그림 9의 FASTA문서간의 XSL을 그림 13과 같이 나타낸다. 이 XSL문서는 웹브라우저에서 전시 될 수 있도록 문서를 HTML형태로 작성한다.

6. 구현 및 평가

이 절에서는 지금까지 기술한 서열정보관리시스템의 구현 결과와 기존의 염기서열정보 관리시스템과 비교 평가하여 기술한다.

7. 구현 질의 및 결과

서열정보관리시스템의 구현 환경은 염기 서열 정보를 저장하기 위해 객체 관계형 데이타베이스시스템인 Oracle 8.1.7을 이용하였고 Oracle 기반의 SQL3에서 지원하는 트리거 기능을 이용하여 버전 관리를 위한 트리거를 구현하였다. 구현언어로는 플랫폼 독립적인 Java 언어를 이용하였고 Java언어에서 제공하는 Sun Microsoft 사의 JAXP(JAVA API for XML Processing) 1.2 XML 파서로 이용하였다. 구현 사항을 예로 들어 설명하면 다음과 같다.

예 1. DNA 서열 ECAE198을 포함하고 있는 서열 파일이 디스크 상에 텍스트 파일 형태로 존재한다고 가정하자. 생물학자는 제안시스템을 통하여 DNA 서열 ECAE198의 염기서열의 베이스 구성을 보기를 원한다. 이 경우 생물학자는 다음과 같은 질의를 할 수 있다. DNA 서열 EACE의 염기 서열의 베이스 구성을 전시하시오.

이 경우 시스템은 파일-파일 열기 메뉴를 선택하게 되면 디스크상에 서열 파일을 열어 그림 14와 같은 정보를 전시한다. 내부적으로 서열 전시 연산과 베이스구성 연산을 수행하여 DNA 서열 ECAE198에서 아데닌(A), 티민(T), 구아닌(G)과 사이토신(C)의 구성 비율과 함께 서열을 보여준다.

예 2. 화면에 전시된 DNA 서열 ECAE198에 대해 생물학자는 서열 연산을 이용해서 서열을 편집, 분석 및

GenBank Flat file		BSML
Field		XPath
Locus		/Bsml/Definitions/Sequences/Sequence
	Locusname	/Bsml/Definitions/Sequences/Sequence@title
	Length	/Bsml/Definitions/Sequences/Sequence@length
	Type	/Bsml/Definitions/Sequences/Sequence@molecule
	Topology	/Bsml/Definitions/Sequences/Sequence@topology
Definition		/Bsml/Definitions/Sequences/Sequence@comment
Accession		/Bsml/Definitions/Sequences/Sequence@ic-acckey
Version		/Bsml/Definitions/Sequences/Sequence/Attribute@name
Source	Organism	/Sequence/attribute@name
Reference	number, range	/Sequence/Feature-tables/Feature-table/reference@title
	Authors	/reference/RefAuthors
	Title	/reference/RefTitle
	Journal	/reference/RefJournal
	Medline	/reference@dbxref
Features	FeatureKey	Sequence/Feature-tables/Feature-table/Feature
		Sequence/Feature-tables/Feature-table/Feature@title(value-type) Title or value-type: allele, attenuator, C_region, CAAT_signal, CDS,conflict ,Dloop,D_segment,enhancer,exon,gene,GC_signal,DNA, etc.
	Location	/Feature/interval-loc or /Feature/site-loc
	Qualifier	/Feature/qualifier
Origin		/BSML/Definitions/Sequences/Sequence/Seq-data

그림 12 BSML과 GenBank 포맷간의 매핑 정보

비교할 수 있다. 이 질의 예에서는 생물학자가 관심 있는 서열의 특정한 일부분을 검색, 서열에 대한 보수 서열과 특정 베이스를 중심으로 회전 연산등을 수행하는 것을 보인다. 수행할 질의는 DNA 서열 ECAE198에 대해서 Search Subsequence, complement sequence, rotate연산을 수행하는 것이다.

첫 번째 서열은 ECAE 198서열에 대해서 0번부터 20번까지의 부분서열을 선택하는 setRange 연산을 수행한

```
<?xml version="1.0" encoding="euc-kr"?>
<xsl:stylesheet version="1.0"  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                               xmlns:fo="http://www.w3.org/1999/XSL/Format">
<xsl:template match="/">
 <html>
          <head>
          <title>FASTA Format Generator</title>
          <body><center><b><font face="굴림"><font color="#3366FF"><font size="15">
          <strong>HTML형식의 FASTA Format</strong>
          </font></font></font></b></center><br></br>
                    <xsl:apply-templates/>
          </body>
          </head>
</html>
</xsl:template>
<xsl:template match="sequence">
    <xsl:value-of select="@id"/>|
   <xsl:value-of select="@version"/>|
   <xsl:value-of select="@molecule"/>|
   <xsl:value-of select="@molecule"/>|
   <xsl:value-of select="@length"/>|
   <xsl:for-each select="attribute">
   <xsl:value-of select="@content"/>|
   </xsl:for-each>
   <br></br>
   <xsl:value-of select="seq-data"/>
   <br></br>
   </xsl:template>
</xsl:stylesheet>
```

그림 13 FASTA 변환을 위한 XSL

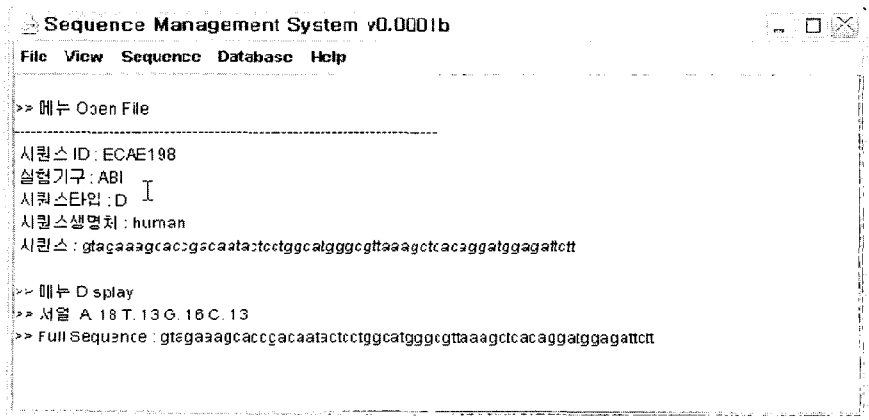

그림 14 염기 서열 베이스 구성 연산과 서열 전시 결과

결과이다. 두 번째 서열은 서열에 대한 5'에서 3' 방향의 상보 서열(complete seuqnce)을 수행한 결과이다. 마지막은 20번째 서열을 중심으로 rotate 연산을 수행한 회전 서열이다.

예 3. 질의 2에서 수행된 연산 결과를 FASTA 형식으로 작성하여 유사성 검색이나 다른 분석프로그램에서 이용할 수 있다. 이번 질의에서는 2번 서열 연산들을 통해서 생성된 서열을 FASTA 형식의 웹 문서로 표현을 수행할 것이다.

예 2에서 수행된 연산 결과를 BSML로 변환하기 위해 TOXML file 메뉴를 선택한다. 이 메뉴의 수행결과는 그림 16 같은 BSML문서를 생성한다. TOFASTA file메뉴를 선택하여 BSML문서를 FASTA 형식으로 전시하기 위한 XSL파일을 XML문서가 존재하는 동일한 디렉토리에 생성된다. 이 연산 후 XML문서를 브라우저에서 보면 그림 17과 같은 결과를 볼 수 있다.

예 4. 현재 편집 창에서 편집이 끝난 서열을 데이타베이스에 입력하게 되면 내부적으로 이 서열이 데이타베이스에 이미 존재하는 서열의 버전서열인지 아니면 새롭게 데이타베이스에 존재하는지를 자동적으로 검사하게 된다. 이 예제에서는 기존에 데이타베이스에 존재하는 서열에 대해서 실험을 통해 서열의 베이스 구성이 변경을 발견하고 이를 데이타베이스에 버전 서열로 저장하고 해당서열의 가지는 모든 버전 서열을 검색하여 XML문서로 저장하기 위한 예이다. 이에 대한 질의는 서열 ECAE198의 버전서열을 삽입하고 검색하여 XML 파일로 저장하시오.

ECAE 198의 서열 버전 파일을 선택하여 서열 입력 메뉴를 통해서 데이타베이스에 입력하면 트리거를 통하여 자동적으로 버전 검사여부를 하여 서열버전 테이블에 저장한다. 그림 18은 버전 삽입 후 서열 식별자를 이용하여 서열 버전을 검색한 결과이며 그림 19는 그림 8처럼 BSML로 표현한 XML문서이다.

6.2 실험

그림 20은 질의로 입력된 서열과 일치하는 데이타베이스의 서열을 검색하기 위한 실행 시간을 측정한 것이

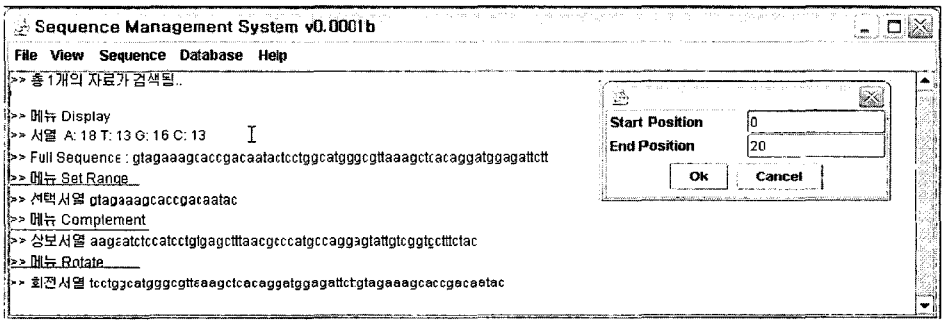

그림 15 Search Subsequence, complement sequence, rotate 연산처리 결과

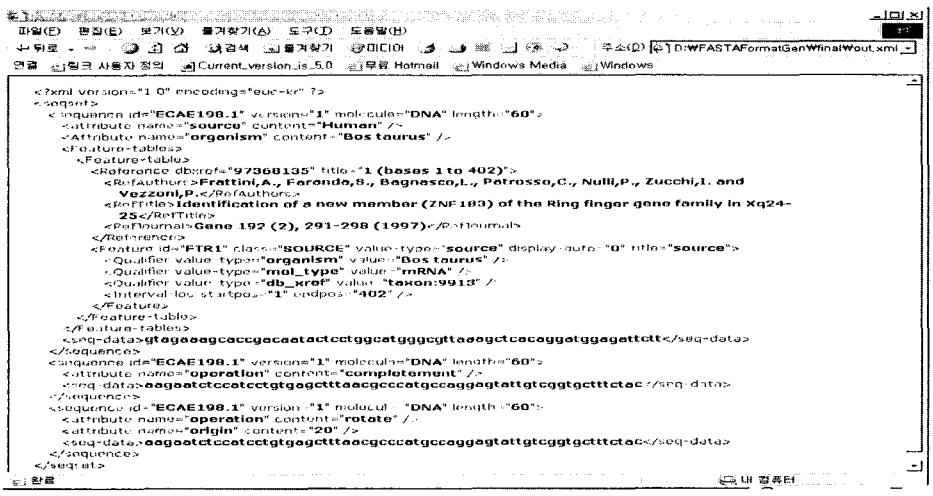

그림 16 서열 연산 결과에 대한 XML 문서

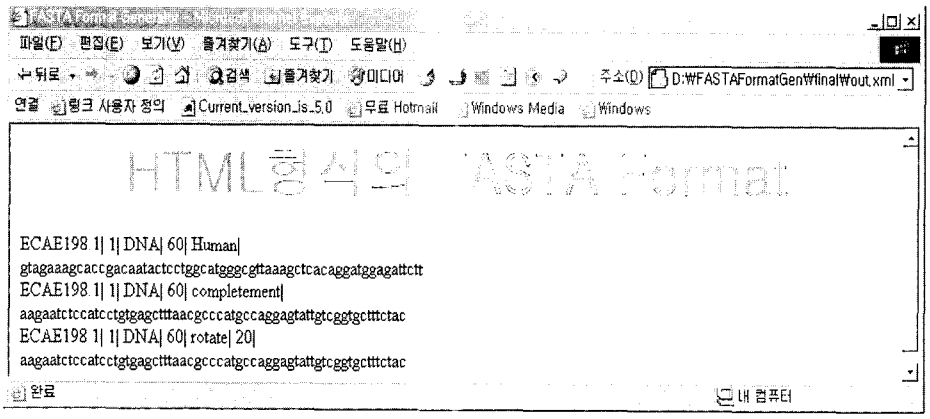

그림 17 XML문서에 FASTA.XSL파일이 적용 결과

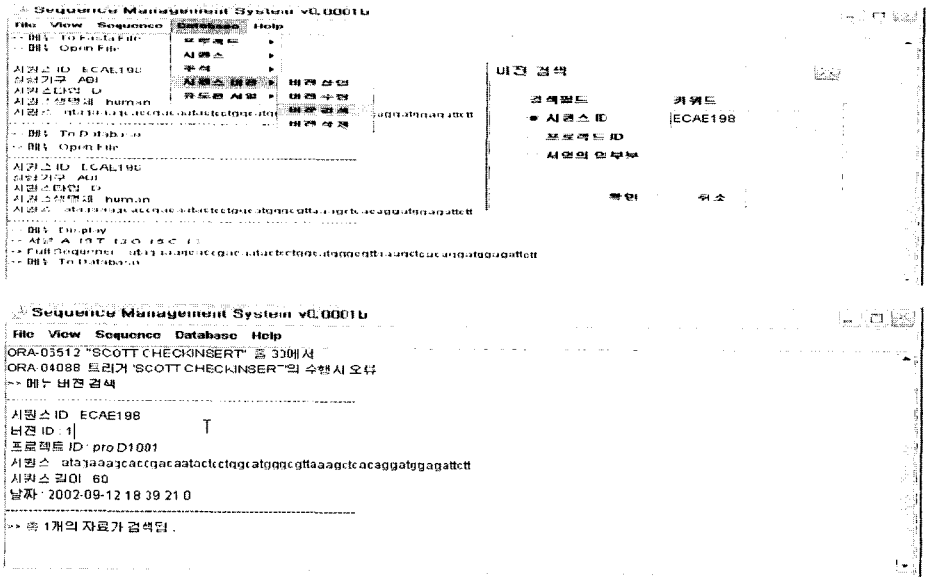

그림 18 ECAE의 서열 버전 삽입 인터페이스 및 검색 결과

다. 이 질의 실행을 위해서 서열을 가변 문자 타입으로 저장하고 like연산 자를 이용하여 서열을 2000개씩 증가하여 10,000개의 서열에 대해서 실험하였다. 데이타베이스에 저장된 평균 서열의 사이즈는 4Kbyte이고 전체 데이타베이스 사이즈는 1.431Gbytes이다.

그림 20에서의 평균 시간은 서열의 수가 증가할수록 선형적으로 증가하는 추세를 보인다. 대부분의 검색 시스템에서 검색결과의 응답 시간을 2초 이내로 잡은 것과 비교할 경우 서열에 대한 검색 시간을 가변 문자로 저장할 경우 검색시스템에서 수용 가능한 결과임을 나타낸다.

그림 21은 새로운 서열이 데이타베이스에 저장될 경우 버전 서열을 검출하는 알고리즘에 대한 실행시간 결과를 보여준다. 이 실험에서 서열 버전을 검출하는 방법을 제안 시스템에서 제시한 1) 트리거를 이용한 방법과 2) 트리거를 이용하지 않고 구현한 경우로 나누어 평가하였다. 트리거를 이용하지 않은 경우는 4.2절에서 제시한 서열 버전 검출알고리즘을 서열 입력 프로그램에서 구현하여 평가하였다.

그림 21같이 트리거를 이용한 방법은 응용프로그램에서 직접 버전을 검출한 알고리즘을 구현한 경우보다 더 좋은 성능을 보이며 특히, 데이타베이스의 서열의 수가 증가함에 따라 서열 버전 검출 시간이 일정하게 유지되고 있음을 알 수 있다. 이것은 입력된 서열과 데이타베

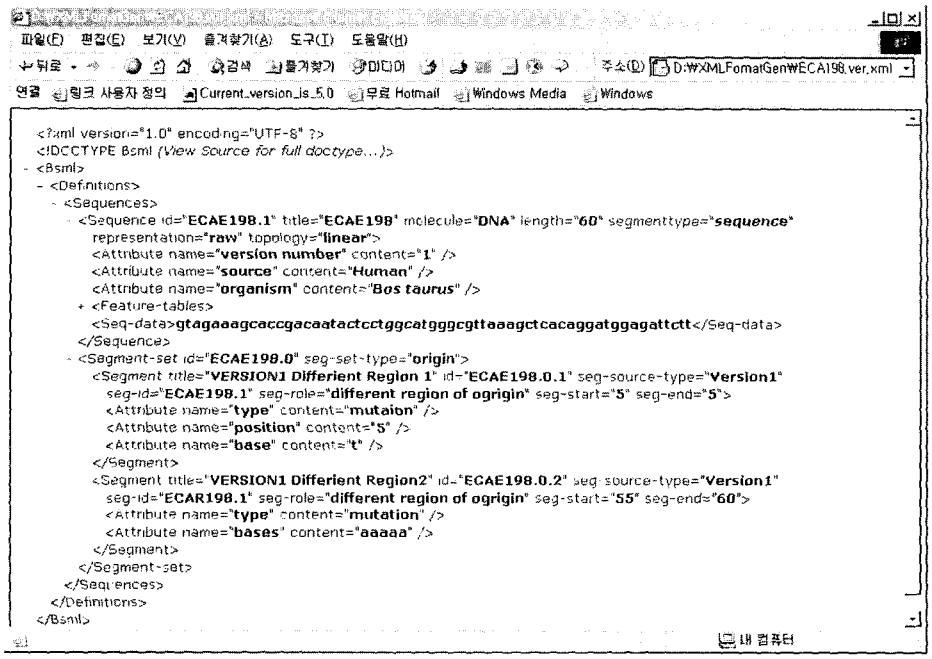

그림 19 검색된 버전 결과를 XML로 생성한 결과

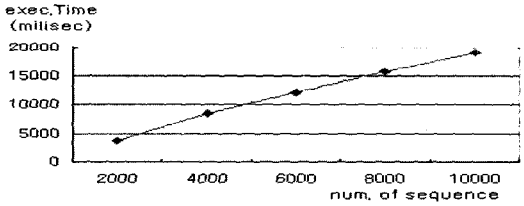

그림 20 서열 검색 실행 시간

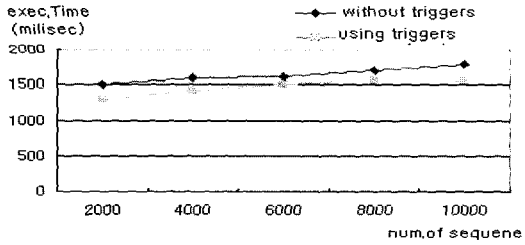

그림 21 서열 버전 검출 기법 실행 시간

이스에 존재하는 서열을 비교하여 버전 서열인지를 판단하기 위해서 서열 테이블에 접근할 때 서열 식별자를 이용하기 때문에 서열의 검색시간이 일정하게 유지된다.

6.3 구현 평가

2000년 HGP이후 데이타 량의 증가로 염기 서열 데이타의 분석뿐만 아니라 분석을 효율적으로 지원하기 위한 서열데이타관리시스템에 대한 요구가 증가되었다. 따라서, 이 논문에서는 서론에서 제시한 새로운 이슈를 반영하여 서열데이타관리시스템을 구현하였다. 제안시스템의 평가는 Sanger Center의 Staden Package, NCBI의 GenBank와 BSML을 기반으로 하고 있는 Genomic Recentrics사의 Genomic Workspace를 비교하여 표 1과 같이 나타낸다.

제안 시스템이 기존 시스템인 Staden Package, GenBank와 Genomic Workspace와 다른 점은 다음과 같

표 1 기존서열관리시스템과 비교

항목	Staden Package	GenBank	Genomic Workspace	제안시스템
서열 조작	6개 서열 연산	연산 지원 안함	연산 지원 안함	6개 서열 연산 지원
파일 포맷	EMBL 포맷	GenBank/ ASN.1/ BSML	BSML	BSML DTD에 기반한 XML 포맷
서열저장	파일	DBMS	BSML문서 및 텍스트	DBMS
서열변경	지원하지 않음	변경 이력 지원	지원하지 않음	변경 이력 지원
파일 변환	지원하지 않음	지원하지 않음	일부(BSML)	XML기반 변환지원 (XML, BSML, FASTA, GenBank)

다. 첫째, 서열 저장 시에 실험을 통해 생성되는 서열 데이타의 변경과 진화적인 변이의 추이를 함께 저장관리 할 수 있도록 하나의 서열에 대한 서열 버전을 저장한다. 둘째, 공개용 데이타베이스의 플랫 파일 및 생물학 연구실의 실험 파일로부터 생물학자가 원하는 필드를 자유롭게 파싱하여 추출하고 다른 포맷으로 변환이 가능하다. 특히, 이를 통하여 서열 분석을 위한 기본 포맷인 FASTA 포맷을 생성할 수 있다.

- 서열 버전의 관리는 SNPs(Single Nucleotide Polymorphisms) 서열 저장관리에 적용될 수 있다. SNPs 서열은 점 변이가 있는 서열들이다. SNPs 서열들은 생물학적으로 상동 관계에 있으며 서열의 구성이 거의 유사하다. 따라서 이러한 SNPs 서열들을 서열 버전으로서 저장관리 할 수 있다.
- 서열연산을 통해서 실험이나 분석을 위한 새로운 서열을 설계하거나 생성 가능하다. PCR(Polymerase chain reaction)은 DNA 분자를 시퀀싱 하기 위하여 DNA 분자의 특정한 부분을 선택하여 증폭하는 실험이다. 이때 사용되는 template DNA 의 이중 나선 중 한쪽 나선의 상보 서열이 형성된다. 이러한 반복적인 합성을 통해서 template DNA가 증폭된다. 이러한 새로운 합성을 하기 위해서 잘 알려진 Primer를 이용한다. Primer를 디자인하기 위해서 제안시스템의 서열 연산이 적용될 수 있다.
- 서열 분석을 효율적 지원하기 위해 시스템에서 서열 정보를 XML로 표현하고 다른 생물학적 포맷으로 변환을 지원한다. BSML DTD를 기반하여 XML로 표현하였기 때문에 다양한 분석 프로그램에서 사용하는 서로 다른 포맷으로 변환이 용이하며 데이타의 교환이 가능하다. 특히 대부분의 서열 유사성 검색 및 분석 프로그램은 FASTA 포맷을 사용하므로 서열분석 프로그램에서는 FASTA 포맷으로 변환이 필수적이다. 이 논문에서는 데이타베이스에서 검색된 서열을 FASTA 포맷으로의 변환 및 연산을 통해서 유도된 서열도 FASTA 형식으로 변환이 가능하다는 장점을 갖는다.

7. 결 론

HGP를 통하여 염기 서열에 대한 시퀀싱 기술이 보급되어 국내에서도 생명체의 염기 서열 데이타를 대량 생산하고 있다. 데이타 량의 증가로 염기 서열 데이타의 분석뿐만 아니라 분석을 효율적으로 지원하기 위한 서열데이타관리시스템에 대한 요구가 증가되었다.

따라서 이 연구에서는 시퀀싱된 염기 서열 데이타의 효율적인 관리를 위해 염기 서열 정보의 편집, 저장, 검색과 서열 파일 포맷 생성 및 변환을 수행하는 서열 정보관리 시스템을 설계하고 구현하였다. 서열 포맷 변환기를 통하여 서열 분석프로그램 간의 데이타 교환을 위해 XML기반 BSML을 이용하여 이질적인 서열 파일 포맷으로 변환할 수 있음을 보여주었다. 또한 시퀀싱 실험에서 다양한 원인에 의해 발생되는 염기 서열의 변경 정보를 트리거를 이용하여 자동적으로 관리 수 있음을 보여주었다.

이 연구는 염기 서열뿐만 아니라 복잡하고 이질적인 생물학 데이타를 관리하기 위한 개방형의 시스템 구조를 갖는다. 따라서 이 연구 결과를 돌연변이 연구를 위한 SNPs 데이타의 관리에 적용 가능하다. 또한 국내의 생물학 실험실에서 자체적으로 시퀀싱을 통해 생산된 서열 데이타를 효율적으로 저장관리 할 수 있게 활용될 수 있다. 향후 연구로는 진화적 관계 분석을 위한 Pylogenetic tree 분석 기법 및 사용자를 위한 편집기능 등을 추가하여 사용자 정의의 서열 분석 연구를 위한 통합관리시스템의 설계하는 것이다.

참 고 문 헌

[1] S. H. Park, K. H. Ryu, H. S. Son, A Protein Structural Information Management Based on Spatial Concepts and Active Trigger Rules, DEXA03 : Database and Expert Systems Applications, LNCS2736 : 413-422, 2003.

[2] S. H. Park, K. H. Ryu, B.J. Jeong, H. S. Son, Version Management of a genomic sequence database using active rules and temporal concepts, ISMB 03', Australia, Jun 29, July 3, 2003.

[3] K. S. Jung, S. H. Park, K. H. Ryu, H. S. Son, Sequence Version Management System based on Trigger, Korean Society for Bioinformatics Annual Meeting, Vol.1, pp. 134-141, 2002.

[4] J. Ostell, S.J. Wheelan, J.A. Kans, The NCBI data model. Chapter 2 in Bioinformatics : A Practical Guide to the Analysis of Genes and Proteins, 2nd ed., edited by Baxevanis, A.D. and Ouellette, B.F.F. New York : John Wiley & Sons, pp. 19-43, 2001.

[5] R. Elmasri, S. B. Navathe, "Fundamentals of Database Systems," Addison-Wesley, 2000.

[6] R. H. Li, S. H. Park, K. S. Jeong, K. H. Ryu, Integrated data modeling of protein structures using a fact constellation model based on a XML mediated warehouse system, ISMB 03', Australia, Jun 29-July 3, 2003.

[7] S. H. Park, Y. Han, K. H. Ryu, "Building Genime and Protein Sequence Information Management System," 7th KOSTI Workshop on Korean Infrastructure for Science and Technology Information, pp. 234-247, 2002.

[8] A.D Baxevanis, B.F.F. Ouellette, Bioinformatics : A Practical Guide to the Analysis of Genes and

Proteins, pp. 45-59, Wiley-Liss, Inc, 2001.
[9] S. I. Letovsky, Bioinformatics Databases and Systems, Kluwer Academic Publishers, 2000.
[10] G. Stoesser, W. Baker, A. V.D Broek, E. Camon, M. Garcia-Pastor, C. Kanz, T. Kulikova, V. Lombard, R. Lopez, H. Parkinson, N. Redaschi, P. Sterk, P. Stoehr, M. Ann T., "The EMBL nucleotide sequence database," Nucl. Acids. Res. Vol.29, pp. 17-21, 2001.
[11] J. Widom, S. Ceri, Introduction to Active Database Systems. Active Database Systems : Triggers and Rules For Advanced Database Processing, Morgan Kaufmann (1996)1-41.
[12] J. Spitzner, Bioinformatics Sequence Markup Language Manual, LabBook Inc., 1997.
[13] D. L. Wheeler, D. M. C. A. E. Lash, D. D. Leipe, T. L. Madden, J. U. Pontius, G. D. Schuler, L. M. Schriml, T. A. Tatusova, L. Wagner, B. A. Rapp, Database resources of the National Center for Biotechnology Information : 2002 update, Nucl. Acids. Res. Vol : 30. pp. 13-16, 2002.
[14] R. Staden, K. F. Beal, J. K. Bonfield The Staden Package, 1998. Computer Methods in Molecular Biology, pages 115-130, vol. 132 : Bioinformatics Methods and Protocols Eds Stephen Misener and Steve A. Krawetz. The Humana Press Inc., Totowa, NJ 07512.
[15] D. A. Benson, I. K. Mizrachi, D. J. Lipman, J. Ostell, B. A. Rapp, D. L. Wheeler "GenBank" Nucl. Acids. Res. Vol : 30, pp. 17-20, 2002.
[16] B. James, K. Beal, K. F. Betts, J. Matthew, S. Rodger. Trev : a DNA trace editor and viewer. Bioinformatics Vol.18, pp. 194-195, 2002.
[17] J. Bonfiled, K. F. Beal , M. Jordan, Y. Cheng, R. Staden, The Staden Package Manual, Medical Research Council Labortory of Molecular Biology, 2001.
[18] R. Staden, D. P. Judge, J. K. Bonfield SEQUENCE ASSEMBLY AND FINISHING. A Practical Guide to the Analysis of Genes and Proteins. Second Edition Eds. Andreas D. Baxevanis and B. F. Francis Ouellette. John Wiley & Sons, New York, NY, USA, 2001.
[19] Altschul, S. F., Carrol, R. J., and Lipman, D. J.(1990). Basic local alignment search tool. J. Mol. Biol., Vol. 215, pp. 403, 1990.
[20] Karlin, S. & Altschul, S.F, "Methods for assessing the statistical significance of molecular sequence features by using general scorling schemes," Proc. Natl. Acad. Sci. USA 87, 1990.
[21] F. Achard, G. Vaysseix, XML, bioinformatics and data integration, Society Technical Committee on Data Engineering, 1999.
[22] J. Ostell, The NCBI software tools. In Nucleic Acid and Protein Analysis : A Practical Approach, M. Bishop and C. Rawlings, Eds. Oxford: IRL Press, pp. 31-43, 1996.
[23] D. W. Mount, "Bioinformatics : Sequence and Genome Analysis," Cold Spring Harbor Laboratory Press, 2001.
[24] Pearson W.R., Lipman D.J., "Improved tools for biological sequence comparison," Proc. Narl. Acad. Sci. vol 85 pp. 2444-2448, 1988.
[25] D. Fenyo, The Biopolymer Markup Language, Oxford University Press, 1999.
[26] The Genomic Workspace User Manual 4.0, Technical Memo, Rescentris, Ltd., 2003.

박 성 희

1996년 충북대학교 도시공학과 졸업(공학사). 1998년 한국전자통신 연구원 컴퓨터 소프트웨어 연구소 위촉 연구원. 2001년 충북대학교 대학원 전자계산학과 석사 졸업(이학석사). 2003년~충북대학교 대학원 전자계산학과 박사수료. 관심분야는 Bioinformatics, 생명정보 데이타 통합, 단백질 구조 예측, XML 데이타베이스

정 광 수

2001년 충북대학교 화학공학부 졸업(공학사). 2004년 충북대학교 정보산업공학과 석사 졸업(공학석사). 2004년~충북대학교 대학원 전자계산학과 박사과정 중 관심분야는 Bioinformatics, 단백질 서열 및 구조, 생명정보 데이타베이스

류 근 호

1976년 숭실대학교 전산학과 졸업(이학사). 1980년 연세대학교 산업 대학원 전산전공(공학석사). 1988년 연세대학교 대학원 전산전공(공학박사). 1976년~1986년 육군 군수 지원사 전산실(ROTC 장교), 한국전자통신연구소(연구원), 한국 방송대학교 전산학과(조교수) 근무. 1989년~1991년 University of Arizona, Research Staff(TempIS 연구원, Temporal DB). 1986년~현재 충북대학교 전기전자 컴퓨터 공학부 교수. 관심분야는 시간 데이타베이스, 시공간 데이타베이스, 지식기반 정보검색 시스템, Temporal GIS, 데이타 마이닝 및 데이타베이스 보안, Bioinformatics 등