ISSN 1598-9798

데이타베이스연구

28권 제1호 2012년 4월

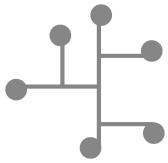
인간 유전자 영역의 SNP 추출을 위한 클라우드 컴퓨팅 알고리즘

A Cloud Computing Algorithm for the Detection of SNPs in Human Gene Regions

김덕근, 윤지희, 홍상균
Deokkeum Kim, Jeehee Yoon, Sangkyun Hong

데이터베이스 소사이어티
Database Society

사단법인 한국정보과학회
The Korean Institute of Information Scientists and Engineers



인간 유전자 영역의 SNP 추출을 위한 클라우드 컴퓨팅 알고리즘

A cloud computing algorithm for the detection of SNPs in human gene regions

김덕근(Deokkeum Kim)[1], 윤지희(Jeehee Yoon)[2], 홍상균(Sangkyun Hong)[3]

요 약

차세대 시퀀싱 기술 (Next Generation Sequencing, NGS)은 인간의 유전적 변이 및 질병 관련 연구를 위한 핵심 기술로 인식되고 있다. 그러나 NGS 데이터를 이용하여 유전적 변이를 정확히 추출하기 위해서는 우수한 퀄리티를 갖는 높은 커버리지의 데이터를 필요로 한다. 본 연구에서는 대용량의 RNA 리드 (read) 데이터를 이용하여 인간 유전자 영역에 존재하는 단일 염기 다형성 (Single Nucleotide Polymorphism, SNP)을 정확히 추출하는 클라우드 컴퓨팅 알고리즘인 CloudSNP를 제안한다. 제안된 알고리즘은 RNA 리드 데이터를 DNA 표준 서열과 mRNA 트랜스크립트 표준 서열에 정렬한 결과를 사용하며, 정렬된 대용량 데이터를 효율적으로 처리하기 위하여 하둡 기반의 맵리듀스 (Map/Reduce) 기법을 사용한다. CloudSNP는 정확한 SNP 검출을 위해 리드 데이터의 정렬 분포 및 시퀀싱 데이터에 포함되는 시퀀싱 에러율 등을 고려하여 유전자 영역에 존재하는 SNP를 추출한다. 또한 대용량의 RNA 리드 데이터와 SNP 추출 결과는 시각적 분석 도구인 Seq-Analyzer에 연결되어 SNP 추출 과정 및 근거를 시각적으로 확인/검증할 수 있도록 지원된다.

주제어: RNA 리드 서열, SNP, 클라우드 컴퓨팅, 맵리듀스

1 마크로젠

2 한림대학교 컴퓨터공학과, 교수, 교신저자

3 한림대학교 컴퓨터공학과, 박사과정

† 이 논문은 2011년도 한림대학교 교비 학술연구비(HRF-2011-302)에 의하여 연구되었음

+ 논문접수: 2011년 9월 8일, 심사완료: 2012년 3월 28일

Abstract

Next generation sequencing (NGS) is a powerful method increasing in popularity for the study of human variation and disease. However, accurate detection of genetic structural variations requires high-quality and high coverage sequencing data. This paper proposes a cloud-computing algorithm, CloudSNP that detects Single Nucleotide Polymorphisms (SNPs) in gene regions of the human genome from high-throughput RNA read data. The proposed method is based on parallel mapping of RNA read data to both genomic and transcriptomic reference sequences, and uses the Hadoop implementation of MapReduce for efficient analysis of large-scale alignment data. CloudSNP identifies SNPs in gene regions by carefully considering the alignment distribution and sequencing errors inherent in real data. The large-scale results containing the RNA reads and SNP calls can be imported directly into Seq-Analyzer, which is a visualization software to validate the SNP analysis results.

Keywords: RNA read sequence, SNP, Cloud Computing, MapReduce

1. 서 론

차세대 시퀀싱 (Next Generation Sequencing, NGS) 기술에 의하여 생성되는 대용량의 서열 정보는 DNA 리드 (read) 혹은 RNA 리드에 해당 한다[1]. 이러한 시퀀싱 데이터는 다양한 생물의 유전체에 존재하는 유전적 구조 변이 (genetic structural variation)를 추출/연구하는 데 사용될 수 있다. 특히 인간의 각 개인 유전체의 특이성을 발견하려는 시도의 하나로 DNA 서열 분석에 관한 연구는 비교적 초기부터 시작되었으며, DNA 시퀀싱 데이터를 이용한 구조적 변이 추출 방법 및 질병 관련성을 규명하기 위한 연구가 매우 활발히 진행되고 있다[2]. 유전적 구조 변이는 작은 규모의 구조 변이로 구별되는 삽입 (insertion), 삭제 (deletion), 전이 (inversion), 단일 염기 다형성 (Single Nucleotide Polymorphism, SNP)과 큰 규모의 구조 변이로 구별되는 유전체 단위반복 변이 (Copy Number Variation, CNV) 등이 있다. 이 중 일반적으로 잘 알려져 있는 SNP는 인간의 유전체 염기 서열 중에서 단일 염기의 차이를 보이는 유전적 변이를 말한다.

그러나 RNA 시퀀싱 분석에 관한 연구는 국내외적으로 아직 초기 단계에 있다고 할 수 있다. RNA는 세포의 유전 정보 발현에 관여하는 물질로 DNA의 유전 정보를 세포질까지 전달하는 역할을 수행한다. 즉, RNA 서열은 DNA 서열과 달리 발현된 유전자 영역만이 시퀀싱되어 얻어지므로 RNA 서열 분석에 의하여 검출되는 유전 변이 결과는 질병의 원인 분석 연구에 보다 중요한 자료로 활용된다.

대표적인 NGS 기술 보유 회사로는 Illumina[3], 454 Life Science[4], Applied Biosystems[5] 등이 있다. 이 들 회사의 차세대 시퀀싱 기술은 짧은 시간 내에 대용량의 리드를 생성하며, 생성된 리드는 30−400 bp (basepair)의 길이를 가지며, 실험 기술에 따라 single−end 혹은 paired−end 타입으로 구별된다. 예를 들어 Illumina의 Genome Analyzer는 열흘 동안 약 33 Gb (300 million 2×100 bp 리드에 해당함)의 대용량의 시퀀싱 데이터를 생성해낸다. 이와 같은 차세대 시퀀싱 데이터는 리드의 길이가 짧고 전체 용량은 매우 커서 이 들 데이터로부터 정확한 분석 결과를 추출하는 데에 많은 어려움이 있다. 차세대 시퀀싱 업체들이 데이터 분석을 위한 도구를 일부 제공하고 있으나 (Illumina의 CASAVA 1.6[3], ABI의 BioScope 3.0[5] 등), 이와 같은 대규모 유전체 데이터를 실시간에 효율적으로 처리, 분석할 수 있는 도구 개발에 관한 연구는 아직 매우 미흡한 상황이다.

본 연구에서는 클라우드 컴퓨팅 기술을 기반으로 높은 커버리지 (coverage)의 대용량 RNA 시퀀싱 (RNA−Seq로 약칭함) 데이터를 이용하여 SNP를 추출하는 병렬 SNP 추출 알고리즘 CloudSNP를 제안한다. 제안하는 방식에서는 RNA−Seq 데이터를 표준 서열인 DNA 서열과 mRNA 트랜스크립트 서열에 동시에 정렬하고, 다음 각 표준 서열의 정렬 정보를 기반으로 유전자 영역에서 SNP를 추출해낸다. DNA 서열을 표준 서열로 사용하는 경우, DNA 서열 전체 영역에 걸쳐 리드의 정렬 정보를 분석하여 SNP를 추출하기 때문에 아직 밝혀지지 않은 새로운 (novel) 구조의 트랜스크립트 상의 SNP를 추출하는 것이 가능하다. 또한 mRNA 트랜스크립트 서열을 표준 서열로 사용하는 경우, 정렬 대상인 표준 서열의 공간이 DNA 서열에 비하여 매우 작아지므로 (전체 DNA 영역의 2−3% 정도에 해당함) RNA 리드가 비교적 정확히 정렬되어, 유전자의 각 트랜스크립트 단위로 정확한 SNP 추출 결과를 얻을 수 있다.

제안된 알고리즘은 동시에 실행되는 다수의 맵리

듀스 (Map/Reduce)[6] 함수를 통해 표준 서열에 정렬된 리드의 결과를 병렬로 처리하며, 효과적인 분산 병렬 처리를 위해 데이터의 파티셔닝 기법을 사용한다. 이와 같이 처리된 클라우드 스케일의 대규모 RNA 데이터와 SNP 추출 결과는 로컬 데이터베이스에 저장하여 시각적 분석 도구인 Seq-Analyzer와 연계되어 그 결과를 추출, 검증할 수 있다. Seq-Analyzer는 SNP 추출 결과에 대한 표준 서열 정보, 정렬된 리드 데이터의 세부 정보 등을 함께 제시하여 SNP 추출 과정 및 근거를 시각적으로 확인/검증할 수 있도록 지원한다.

제안된 방식의 유용성을 보이기 위하여 실 데이터를 이용한 다양한 실험을 수행하였다. NCBI의 Gene Expression Omnibus (GEO) 데이터베이스에서 내려 받은 암환자 데이터와 정상인 데이터를 이용하여 클라우드 컴퓨팅 환경에서 실험을 수행하였으며, 그 결과를 통해 제안된 방식이 대규모 RNA-Seq 데이터로부터 유전자 영역에 존재하는 SNP를 효율적으로 추출하고 있음을 보인다.

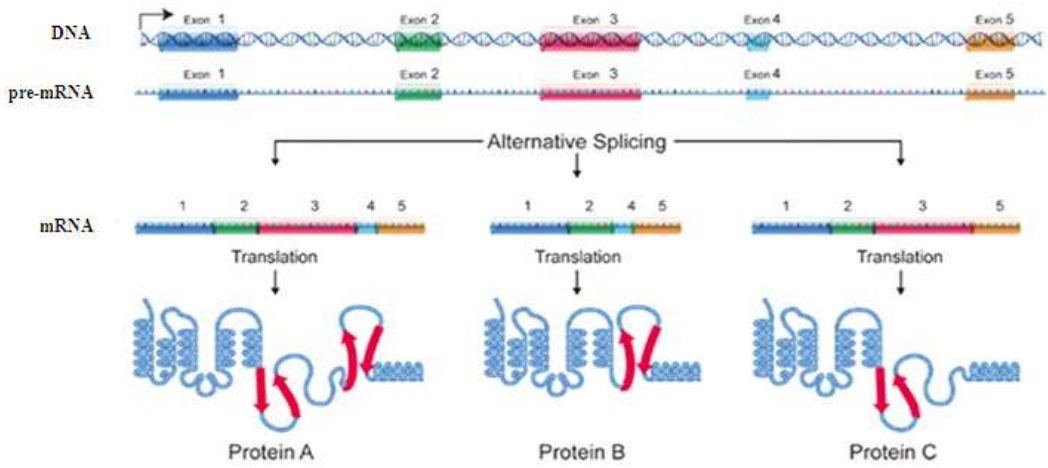
본 논문의 구성은 다음과 같다. 2장에서는 관련연구로, 클라우드 컴퓨팅 기술을 이용한 생물학 정보처리 현황과 RNA 데이터 분석 도구에 대해서 설명한다. 3장에서는 시스템 구성 방식과 CloudSNP 알고리즘을 보인다. 4장에서는 실험을 통해 제안하는 알고리즘의 유용성을 검증하고, Seq-Analyzer를 이용한 SNP 추출 결과에 대한 시각적 분석 방식을 설명한다. 마지막으로 5장에서는 본 논문을 요약하고, 결론을 내린다.

2. 관련연구

2.1 클라우드 컴퓨팅 기술을 이용한 생물학 정보처리

최근 생물정보학 분야에서는 대용량의 데이터를 분석하기 위해 클라우드 컴퓨팅을 기반으로 하는 분석 기법 및 도구 개발에 관한 연구를 진행하고 있다. 클라우드 컴퓨팅은 네트워크 기술을 활용하여 서로 다른 물리적인 위치에 존재하는 컴퓨터들의 리소스들을 가상화 기술로 통합하여 서비스 및 리소스를 제공하는 기술이다. 하둡 (Hadoop)은 클라우드 컴퓨팅 환경의 대표적인 프레임워크로 Apache™의 오픈소스 기반의 분산 컴퓨팅 플랫폼이다[7]. 하둡에는 대용량의 데이터 처리를 위한 하둡 분산 파일 시스템 (Hadoop Distributed File System, HDFS)과 데이터베이스인 HBase 등의 다양한 오픈 소스 프로그램이 제공되며, 분산/병렬 처리를 위한 맵리듀스 프로그래밍 모델을 제공한다.

클라우드 컴퓨팅을 이용한 유전 변이 검출 및 분석을 위한 도구로는 Crossbow[8], CloudBurst[9], CloudBLAST[10], Galaxy[11], Myrna[12] 등이 알려져 있다. CloudBurst는 기존의 서열 정렬 방법인 RMAP[13]을 맵리듀스 기법으로 개발한 것으로 표준 서열을 일정 길이의 시드 (seed)로 분할/생성하고 이와 동일한 크기로 리드를 분할하여 시드를 키로 하는 데이터를 산출한 뒤 시드와 동일한 리드의 단편을 모아서 표준 서열상에서 확장하여 정렬을 수행하는 방법을 사용한다. Crossbow[8]는 기존의 정렬 도구인 Bowtie[14]와 SNP 추출 도구인 SOAPsnp[15]를 결합하여 클라우드 컴퓨팅 환경에서 동작하도록 제작한 도구이다. 그러나 Crossbow는 맵 단계에서 Bowtie를 이용하여 리드들을 정렬하고, 리듀스 단계에서 SOAPsnp를 이용하여 정렬된 결과에 대한 SNP를 추출하고 있어, 병렬 수행의 핵심이 되는 맵 단계에서는 리드 정렬만이 이루어지고 있다. Myrna[12]는 RNA 리드 데이터를 이용하여 차등 발현 (differential expression) 유전자를 검



[그림 1] mRNA의 전사/번역 과정[16]

출하는 프로그램이다. 클라우드 스케일의 대용량 데이터 처리를 위하여 하둡 환경의 맵리듀스 프로그래밍 방식을 사용하였으며, Crossbow와 같이 정렬 도구로 Bowtie를 사용하였고, 통계연산을 위하여 통계 패키지인 알/바이오컨덕터 (R/Bioconductor)를 이용하였다. 그러나 이 들 클라우드 컴퓨팅 기반의 유전체 분석 방식에 관한 연구는 제한된 분야에 적용되고 있으며, 아직 연구가 초기 단계이기 때문에 다양한 목적과 분야에 적용 가능한 병렬 데이터 분석법 및 병렬 처리 도구 개발에 관한 연구가 시급한 실정이다.

2.2 RNA 데이터 분석

인간과 같은 진핵 생물의 유전자에는 엑손(exon)과 인트론 (intron) 영역이 존재하는데, 실제 mRNA로 전사 (transcription)되어 단백질 서열이 되는 부분이 엑손 영역이다. [그림 1]은 DNA의 유전자 정보가 mRNA로 전사 되어 단백질로 번역(translation)되는 과정을 그림으로 간단히 표현한 것이다. 즉, 유전 정보는 DNA에 의하여 암호화되어 저장되며, 전사 과정에 의하여 mRNA로 옮겨지고 다시 번역 과정에 의하여 단백질로 합성되어 유전 정보가 실체화된다. 여기에서 선택 스플라이싱(alternative splicing)은 mRNA의 전구체인 pre-mRNA가 mRNA로 전사될 때 pre-mRNA의 엑손 영역들이 여러 가지 유형으로 다시 연결되는 과정을 나타낸다[16].

서론에서 기술한 바와 같이 mRNA 분석에 의하여 검출되는 유전 변이 결과는 질병의 원인 분석 연구에 중요한 자료로 활용된다. 특히 정상인과 환자의 유전적 정보를 서로 비교해서 환자가 가진 변이만 골라낼 수 있다면 그 안에 질병의 원인이 된 변이가 있을 가능성이 높다고 예측할 수 있다. 전사된 유전자 구조를 판별하기 위해 일반적으로 사용하는 방법은 Expressed Sequence Tag (EST) 혹은 complementary DNA (cDNA)를 이용한 mRNA 분석 방법이다. 그러나 최근의 차세대 시퀀싱 기술을 기반으로 생성되는 대규모 RNA-Seq 데이터를 이용한 전사체 분석 방식은 기존의 마이크로어레이를

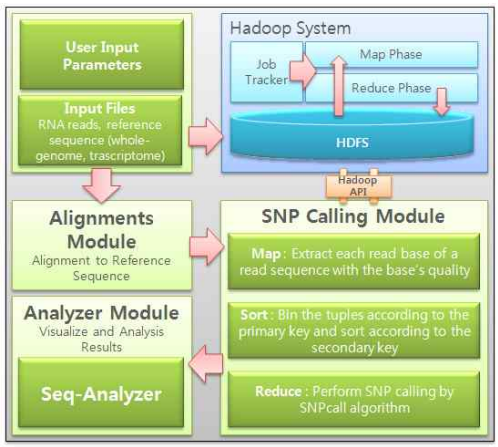
이용하는 방식 혹은 EST/cDNA를 이용한 방식에 비하여 저가에 보다 정확한 결과를 얻을 수 있다는 장점을 갖는다. 그러나 아직 RNA-Seq을 이용한 전사체 분석 방식에 대하여 장점만이 강조되었을 뿐이고, 구체적인 방법론에 대한 연구는 시작 단계이다.

최근 RNA-Seq 데이터로부터 선택 스플라이싱의 유형을 판별하기 위한 다양한 연구 방법론들이 제시되고 있으며, ERANGE[18], TopHat[19], SpliceMap[20], GSNAP[21] 등의 리드 정렬 도구들이 개발되어 있다. 이 도구들은 엑손의 접합 부분에서 생성된 리드인 접합 리드 (junction read)를 표준 서열에 자동 정렬시켜 엑손 경계에서 발생하는 스플라이스 영역을 추출하여 준다. 또한 Cufflinks[22]은 이와 같은 기능 외에도 RNA 서열을 조립(assembly)하여 유전자별 전사량을 추출하고, 이들 값을 기반으로 하여 주어진 샘플 사이의 차등 발현 유전자 등을 보고한다. 한편 참고 논문 [23]에서는 정상인과 백혈병 환자 샘플의 RNA 시퀀싱 데이터를 이용하여 단백질 코딩 영역에 존재하는 SNP를 추출하여 질병의 연관성을 비교, 분석하는 방법론을 제시하고 있다.

3. 클라우드 스케일의 SNP 추출

SNP는 인간의 유전체 염기서열 중에서 단일 염기의 차이를 보이는 유전적 변화 또는 변이를 말하는 것으로, 각 개인별로 검색된 SNP의 위치는 유전적 특성 혹은 질병의 발병 원인으로 해석될 수 있다. 본 장에서는 SNP 추출을 위한 클라우드 컴퓨팅 기반의 시스템 구성 방법과 하둡 환경의 맵리듀스 기법을 이용한 CloudSNP 알고리즘을 보인다.

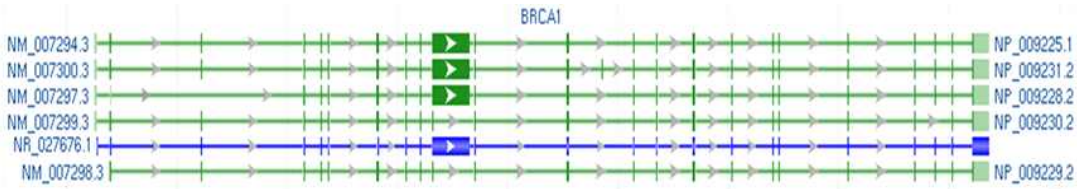
3.1 시스템 구성



[그림 2] 시스템 구성도

다음 [그림 2]에 전체적인 시스템 구성도를 보인다. 시스템 입력은 RNA 리드 (RNA-Seq 데이터), 표준 서열, 사용자 지정 매개변수로 이루어진다. 2장 2절에서 기술한 바와 같이 RNA-Seq 데이터에는 전체 유전체 영역 중 발현된 유전자의 엑손 영역의 정보만이 포함하므로 전체 유전체 영역의 정보를 보이는 DNA 리드 데이터에 비하여 RNA-Seq 데이터는 크기가 상대적으로 작다. 따라서 유전자 변이 분석을 수행하는 경우, RNA-Seq 데이터의 경우에는 수십 커버리지 이상의 높은 커버리지 데이터를 사용하는 것이 일반적이다.

본 시스템에서는 표준 서열로 DNA 서열과 mRNA 트랜스크립트 서열을 사용할 수 있다. DNA 표준 서열은 NCBI에서 제공되는 전체 게놈 (genome) 서열을 그대로 사용한다. mRNA 트랜스크립트 서열은 유전자 정보의 전사 과정에서 생성되는 모든 트랜스크립트 서열 (transcriptome)을 사용한다. [그림 1]에서 설명한 바와 같이 DNA의 각 엑손 영역은 선택 스플라이싱 과정에 의하여 여러 가지 유형으로 다시 연결되어 각각 서로 다른 트랜스크립트 서열로 전사되므로 각 유전자에 대한 모든 트랜스크립트 서



[그림 3] 유전자 BRCA1 영역에 보고되어 있는 6개의 트랜스크립트 구조[17]

열 정보를 통합하여 mRNA 트랜스크립트 서열을 생성한다. 예를 들어 다음 [그림 3]은 NCBI에서 제공하는 Build 36.3 염색체 17번 데이터에 있는 BRCA1 유전자 영역에 보고되어 있는 6개의 트랜스크립트 구조를 보인다. 여기에서 초록색 수직선으로 표시된 부분이 각 트랜스크립트에 있는 엑손 영역을 나타낸다. 즉, mRNA 트랜스크립트 서열에는 BRCA1 유전자 영역에 대하여 현재 보고되어 있는 6개의 트랜스크립트 서열이 저장되게 된다.

시스템의 정렬 모듈 (alignment module)에서는 입력된 RNA 리드 데이터를 사용자가 지정한 표준 서열에 정렬한다. 본 시스템에서는 SOAP 2[24]을 정렬 프로그램으로 사용하는 것을 가정하고 있으나, Bowtie, RMAP, BWA[25], MAQ[26] 등의 정렬 프로그램 혹은 CloudBurst, CloudBLAST 등의 클라우드 컴퓨팅 기반의 정렬 프로그램을 사용할 수 있다. 다음 SNP 검출 모듈 (SNP calling module)에서는 정렬 데이터를 입력 받아 본 연구에서 제안하는 CloudSNP 알고리즘을 통해 SNP 검출을 수행한다. 맵 함수에서는 입력 파일에서 리드 정렬 정보를 읽어 들여 표준 서열의 각 위치별로 정렬된 염기 정보를 수집/출력한다. 다음 리듀스 함수에서는 정렬된 맵 함수 결과를 입력으로 받아 표준 서열의 각 위치에 대한 SNP의 진위 여부를 판단하여 최종 결과를 반환한다. 분석 모듈 (analyzer module)은 RNA 리드 정렬 결과 및 SNP 추출 결과를 저장, 관리, 분석하는 서열 분석 도구인 Seq-Analyzer를 나타낸다. 대용량의 RNA 시퀀싱 데이터와 SNP 추출 결과는 로컬 데이터베이스에 저장된 후, Seq-Analyzer에 연결되어 SNP 추출 과정 및 근거를 시각적으로 확인/검증할 수 있도록 지원된다.

3.2 CloudSNP 알고리즘

SNP 추출을 위한 CloudSNP 알고리즘을 [Algorithm 1]에 보인다. 알고리즘의 동작 원리를 간단히 설명하면 다음과 같다.

맵 함수에서는 입력 파일의 각 줄의 리드 정렬 정보를 읽어 들여 표준 서열의 각 위치에 정렬된 염기 정보를 수집/출력하기 위하여 다음과 같은 과정을 수행한다. 우선 맵 함수는 리드 정렬 정보로부터 [염색체_ID, 리드가 정렬된 위치, 리드 서열, 리드 서열의 염기 품질]의 정보를 추출하여, [염색체_ID, 각 염기의 정렬된 위치, 파티션 번호]의 출력 키 (K2)를 만든다. 여기서 파티션 번호는 정렬 위치에 따라 파티셔닝 과정을 통해 계산된 파티션의 번호를 설정하게 된다.

파티셔닝 과정은 각 노드의 로드 밸런싱 (load balancing)과 SNP 추출 과정에서 표준 서열 참조 시에 표준 서열을 읽어 들이는 과정을 최적화하기 위하여 사용된다. 파티셔닝 과정을 간략히 설명하면 다음과 같다. 우선 하둡의 설정에 의해 지정되는 리듀스 함수의 개수로 표준 서열의 길이를 균등한 길이로 나눈다. 맵 함수에서는 정렬 결과를 염기 단위

```
[Algorithm 1] CloudSNP
Input : alignment_result, m_count_threshold, quality_threshold, zygosity_threshold
Output : set of SNPs
    MAP(K1, V1, K2, V2)
 1.     for each chrID, pos, read, quality in V1 do
 2.         part = calculatePartition(chrID, pos);
 3.         for each inc from 0 to (read.length-1) do
 4.             K2.set(chrID, pos+inc, part)
 5.             V2.set(read[inc], quality[inc])
 6.     return  List of <K2, V2>
    REDUCE(K2, List<V2>, K3, V3)
 7.     SNP = SNPcall(List<V2>, m_count_threshold, quality_threshold, zygosity_threshold);
 8.     if (SNP.result == 0) then
 9.         return;
10.     else
11.         K3.set(K2.chrID, K2.pos)
12.         V3.set(SNP.ref, SNP.readAlleles, SNP.zygosity, SNP.Score, SNP.totalCount, SNP.usedCount)
13.     return List of <K3, V3>
```

로 나누는 과정에서 표준 서열상의 위치에 따른 파티션 번호를 산출하여 출력키에 기록하게 된다. 이렇게 지정된 파티션 번호에 의해 동일 파티션에 해당하는 데이터가 동일 리듀스 함수로 전달한다. 각 리듀스는 표준 서열상의 일정 크기 영역을 처리하기 때문에 리듀스 함수간에 유사한 데이터량을 처리하게 된다.

또한 맵 함수는 표준 서열의 pos번째 위치에 정렬된 length의 길이를 가지는 리드에 대하여 [pos, pos+1, ... , pos+length-1]의 위치에 각각 정렬되는 염기와 품질 정보를 분리해 내는 작업을 수행하여 출력 값 (V2)으로 반환한다. 이때 다수의 키와 값의 출력을 산출하기 때문에 각 맵 함수의 결과는 출력 키 K2와 출력 값 V2의 쌍으로 이루어진 리스트 형식으로 산출되어 맵 합수의 결과로 반환된다. 맵 함수의 처리 과정을 예를 들어 설명하면, 염색체 1번의 1번 위치에 정렬된 리드가 길이는 2이고 염기 정보가 T, G이며 품질 점수가 각각 28, 30 인 경우 (이때 모든 위치는 파티션 0번이라고 가정), [1, 1, 0], [T, 28]와 [1, 2, 0], [G, 30]의 출력 키 K2와 출력 값 V2의 쌍이 리스트 형식으로 반환된다.

다음, 맵 함수의 결과를 이용하여 리듀스 함수에서는 SNP 추출을 위한 SNPcall 함수를 실행한다. SNPcall 함수의 수행 방식을 단계별로 설명하면 다음과 같다.

(Step 1) 키 값 에 의하여 정렬된 맵 함수 결과를 입력 받아 표준 서열의 해당 위치에 정렬된 염기 집합을 구성하고, 이들로부터 염기 정보 (염기의 종류별 개수, 각 염기의 품질)를 추출한다. 이때 염기의 종류는 A, C, G, T의 4가지이며 염기 품질은 프레드 품질 점수 (phred quality score)[27]로 주어진다. 프레드 품질 점수는 각 염기가 시퀀싱 되는 과정에서 에러 율 혹은 정확도를 나타내는 값으로 아스키 코드 (ASCII code)의 값으로 표현되며 프레드 품질 점수가 20인 경우 1%의 에러 율을 나타내며 정확도로는 99%라고 할 수 있다. 제안한 알고리즘에서는 정확한 SNP 검출을 위하여 정렬된 염기 집합으로부터 염기 정보를 추출하는 과정에서 품질이 주어진 임계값 (quality_threshold) 미만의 염기는 제외시킨다. 여기에서는 품질 임계값으로 20을 사용하는

것을 가정한다.

(Step 2) Step 1에서 수집된 염기 정보를 분석하여 표준 서열의 염기와 다른 종류의 단일 염기가 주어진 정렬 임계값 (m_count_threshold) 이상 정렬된 경우 해당 위치를 마킹한다. 정렬 임계값은 입력 데이터의 시퀀싱 커버리지와 사용자의 선택에 근거하여 결정되며, 낮은 커버리지의 영역에서 발생할 수 있는 오류를 최소화한다. 본 연구에서는 정렬 임계값으로 4 이상의 값을 사용하는 것을 가정한다. 해당 위치가 마킹된 경우에는 Step 3을 실행하고 마킹되지 못한 경우에는 함수를 종료한다.

(Step 3) 염기 집합의 각 염기에 대하여 다음과 같은 방식으로 염기 점수를 계산한다. 임의의 염기가 해당 위치에 n번 정렬되어 각각 Q_1, Q_2, ... , Q_n의 품질을 갖는 경우, 해당 염기의 품질 점수 S를 다음 [식 1]에 의하여 계산한다.

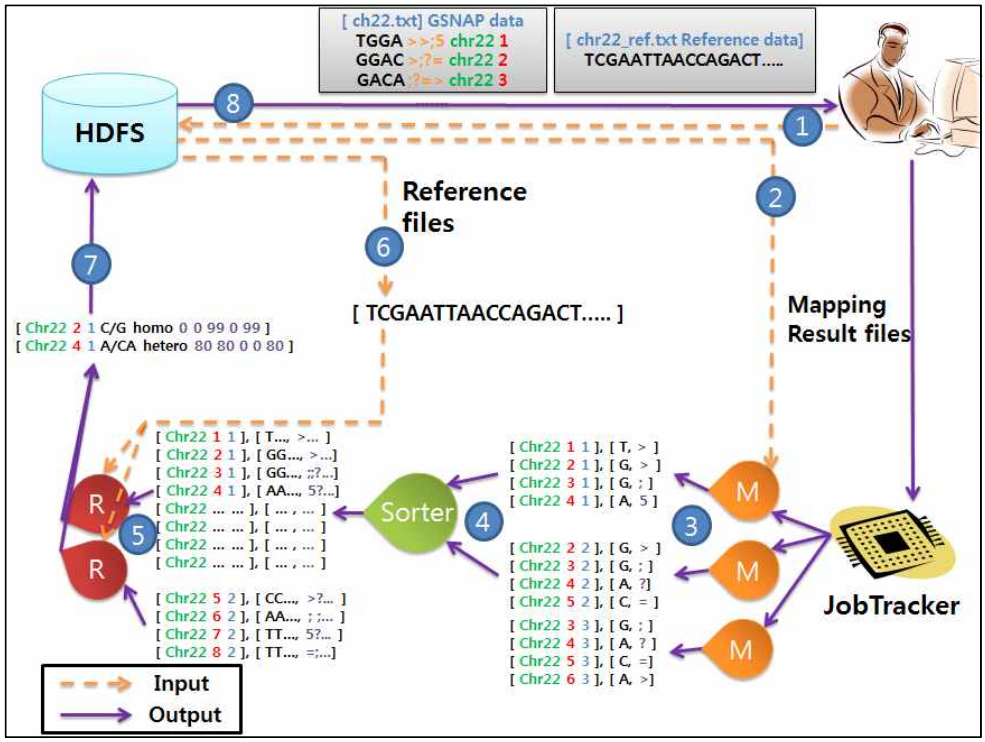
$$\sum_{=1}^{n} 1-10^{-10} \quad \text{..... [식 1]}$$

우선, 프레드 품질 점수 Q를 에러 확률 값 ()으로 변환한다. 이때 에러 확률 값이 작을수록 에러율이 낮다 (정확도가 높다)는 것을 의미하기 때문에 상수 1에서 확률 값을 뺀 값의 합으로 염기별 점수를 계산하게 된다. 이렇게 계산되어진 염기 점수를 통해 각 염기의 순위를 정하게 된다.

(Step 4) Step 3에 의해 산출된 각 염기의 점수와 순위를 통해 SNP를 검출한다. SNP는 반드시 하나의 다른 염기로 대신 되는 것이 아니라 경우에 따라 두 염기가 선택적으로 존재할 수 도 있다. 전자의 경우를 동형 접합 SNP (homozygous SNP), 후자의 경우를 이형 접합 SNP (heterozygous SNP)라고 한다. SNP 검출 알고리즘에서 특정 위치에 A, C, G, T의 염기들이 정렬될 때 품질 점수와 각 염기의 정렬 빈도에 의해 계산된 염기별 품질 점수를 이용하여 각 염기의 순위를 매기고 1위인 염기에 대하여 2위인 염기의 품질 점수가 접합자 구조 임계값 (zygosity_threshold) 이상의 값을 가지는 경우에 1, 2위의 염기를 이형 접합 SNP로 구분하게 된다. 반면 그 미만이며 1위의 염기가 표준 서열의 염기와 다를 경우 1위의 염기만을 동형 접합 SNP로 구분한다. 이 역시 사용자의 선택에 의하여 임계값이 변경 가능하며 CloudSNP에서는 기본값으로 9의 임계값을 사용한다.

이와 같은 과정에 의하여 SNPcall 함수에 의한 SNP 추출이 보고되는 경우, 마지막으로 리듀스 함수는 [염색체_ID, 위치]로 이루어진 출력 키 (K3)와 [표준 서열 염기, SNP 염기, 접합자구조, 전체 염기 수, 사용된 염기 수, 염기별 출현 횟수, SNP 염기 품질 점수]로 이루어진 출력 값 (V3)을 만들어 결과를 반환하게 된다.

[그림 4]는 제안된 알고리즘에 의한 데이터 처리 과정을 도식적으로 나타낸 것이다. 단계별 흐름은 다음과 같다. ① 사용자는 RNA 리드 정렬 파일과 표준 서열을 HDFS에 복사한다. ② 입력된 파일은 HDFS의 블록 크기와 파일의 크기에 따라서 파일이 자동으로 분할되어지고, 분할된 파일의 개수에 따라 맵 함수의 수가 결정되어 수행된다. ③ 맵 함수가 수행되어, 키 값에 calculatePartition 함수에 의해 파티션 번호가 삽입되어 리듀스 함수에서 처리되는 데이터가 일정하게 분산 되도록 한다. ④ 맵 함수에 의하여 산출된 결과 값은 같은 키 값을 가지는 값들끼리 모이게 되고, 모인 값들은 각각 Shuffle/Sort 과정을 거치게 된다. ⑤ Sort된 값들은 각각 리듀스 함수에 넘겨져 수행된다. 리듀스 함수는 사용자가 정의한 개수만큼 생성되며 그 개수에 따라 일정한 파티션 번호순으로 데이터를 처리하게 된다. ⑥ 리듀



[그림 4] 데이터 처리과정

[표 1] SNP 추출 결과의 예

Chr#	Pos#	Part#	Ref	SNP	Zygosity	A	C	G	T	Total	Used	Score
22	16040501	1604	C	G	Homo	0	0	67.98	0	72	68	67.98
22	16040556	1604	T	T/C	Hetero	0	8.0	0	11.0	33	19	11.00/8.00
22	16040646	1604	A	G/A	Hetero	11.97	0	20.98	0	78	33	20.98/11.97
22	16041178	1604	A	G/A	Hetero	12.99	0	34.99	0	80	48	34.99/12.99
22	16041347	1604	C	C/G	Hetero	0	19.99	17.99	0	50	38	19.99/17.99
22	16041764	1604	T	C	Homo	0	11.98	0	0	16	12	11.98

스 함수는 취합된 리드의 염기 정보와 표준 서열 정보를 이용하여 SNP 추출 연산을 수행한다. ⑦ 리듀스 함수를 통해 최종적으로 산출된 값들은 다시 HDFS에 저장된다. ⑧ 사용자들은 HDFS에 저장된 결과 값을 다시 로컬 시스템으로 복사하여 사용하게 된다.

[표 1]에 CloudSNP 알고리즘에 의한 SNP 추출 결과의 예를 보인다. 여기에서 4번째 행에 보이는 22번 염색체의 16041178번에 위치하는 SNP 추출 결과는 다음과 같이 해석된다. 레퍼런스 염기가 A이고 해당 위치에 48개의 유효한 염기가 정렬되었으며, 염기 G는 가장 높은 34.99의 점수를 가지며, 염기 A는 12.99의 점수를 갖는다. 따라서 이 들 점수 비율에 의하여 이형 접합으로 분류된다.

[표 2] Jurkat 샘플과 CD4+ 샘플의 엑손 영역에서의 SNP 추출 비교 결과

reference	DNA sequence (size: 2.9GB)		mRNA transcript sequence (size: 109MB)	
sample	Jurkat	CD4+	Jurkat	CD4+
number of reads	46,934,910	56,138,268	46,934,910	56,138,268
read mapping file size	3.27GB (unique match)	3.58GB (unique match)	5.98GB (all match)	5.26GB (all match)
SNP call	17,522	12,841	59,116	52,247

4. 성능평가 및 시각적 분석 도구

본 장에서는 실험을 통하여 제안하는 알고리즘의 성능을 분석하고, 시각적 분석 도구 Seq-Analyzer를 이용한 SNP 결과의 검증 및 분석 방식을 보인다.

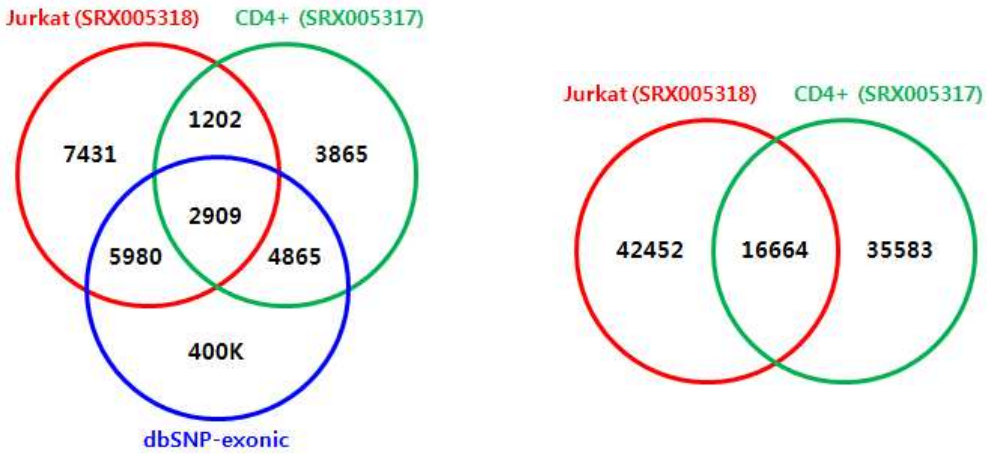
4.1 성능 평가

본 실험에서는 우선 CloudSNP 알고리즘의 SNP 추출 정확도를 검증하고, 다음 클라우드 컴퓨팅 환경에서 노드 수 증가에 따른 검색 속도 향상 지수를 평가한다. 실험 데이터로는 NCBI의 Build 36.3의 표준 서열과 NCBI의 GEO 데이터베이스 (http://www.ncbi.nlm.nih.gov/geo)[28]에서 내려받은 GSE16190의 수납 번호 (accession number)를 가진 시퀀싱 데이터를 이용하였다. GSE16190의 데이터 셋에는 급성 림프구성 백혈병 (Acute Lymphoblastic Leukemia, ALL) 질환을 가진 환자의 Jurkat T cell에서 추출한 mRNA와 건강한 사람의 CD4+ T cell에서 추출한 mRNA를 시퀀싱하여 얻어낸 두 샘플의 RNA-Seq 데이터가 포함되어 있다 (이후, 두 샘플의 데이터를 Jurkat와 CD4+로 약칭하여 부른다). 이 두 데이터는 Illumina GAⅡ를 통해 시퀀싱한 것으로 리드 길이는 36 bp이며, Jurkat (SRX005318)는 46,934,908개의 리드 데이터로 구성되어 있으며, CD4+ (SRX005317)는 56,138,265개의 리드 데이터로 구성되어 있다. 표준 서열로는 DNA 서열과 mRNA 트랜스크립트 서열을 사용한다.

실험을 위한 클라우드 컴퓨팅 환경으로는 CentOS 5.5와 Hadoop 0.20.2를 사용하였고, 각 노드는 2 GB의 주기억 장치와 200 GB의 보조 기억 장치, Intel Pentium IV 3 GHz의 중앙 처리 장치를 가진다. 개발 언어로는 자바를 사용하였다.

첫 번째 실험에서는 두 가지 샘플 데이터에 대한 SNP 추출 실험을 수행하여 제안된 알고리즘의 정확도를 평가하였다. 이 때 각 샘플에 대하여 두 가지 표준 서열을 이용한 실험을 각각 수행하였다.

다음 [표 2]는 CloudSNP에 의하여 두 샘플 데이터에서 추출한 SNP 추출 결과를 비교하여 나타낸 것이다. 우선 표 2의 왼쪽 데이터는 DNA 서열을 표준 서열로 사용하여 실험을 수행한 결과를 나타낸다. 여기에서는 SOAP을 이용한 리드 정렬을 수행하는 경우, 리드가 표준 서열에 중복 정렬되는 것을 허용하지 않고 유일하게 정렬된 (unique match) 리드 결과만을 사용한다. 또한 추출된 SNP 결과는 전체 DNA 영역 중 엑손 영역에서 추출한 SNP 결과만을 나타낸다. 표 2의 오른쪽 데이터는 mRNA 트랜스크립트 서열을 표준 서열로 사용하여 실험을 수행한 결과를 나타낸다. 하나의 유전자 영역에 대하여 서



(a) DNA 서열을 사용한 경우 (b) mRNA 트랜스크립트 서열을 사용한 경우

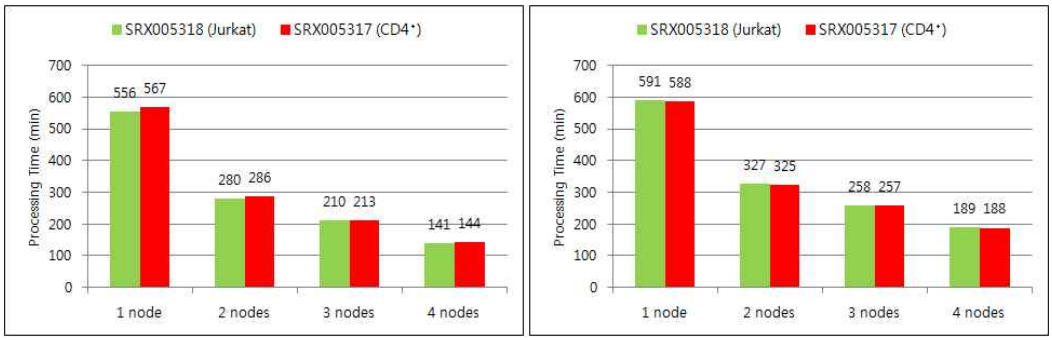
[그림 5] Jurkat 샘플과 CD4+ 샘플의 엑손 영역에서의 SNP 추출 결과

로 유사한 구조를 갖는 트랜스크립트가 복수로 존재할 가능성이 있다. 따라서 SOAP을 이용한 리드 정렬을 수행하는 경우, 표준 서열 영역에 중복으로 정렬된 리드의 사용을 허용한다 (all match). 또한 추출된 SNP 결과는 모든 트랜스크립트 상에서 발견된 SNP 결과를 나타낸다.

다음 [그림 5]는 이 들 SNP 추출 결과를 비교하여 밴 다이어그램으로 나타낸 것이다. 그림 5(a)는 DNA 서열을 사용한 경우를 나타내며, 그림 5(b)는 mRNA 트랜스크립트 서열을 사용한 경우를 나타낸다. 특히 DNA 서열을 사용한 그림 5(a)의 경우에는 대표적인 SNP 데이터베이스인 dbSNP (http://www.ncbi.nlm.nih.gov/projects/SNP)[29]에 보고된 SNP와 비교하였다. DNA 서열을 이용한 경우 ALL 질환을 가지는 Jurkat 샘플에서는 총 17,522개의 SNP가 검출되었으며, 이 중 dbSNP에 보고되지 않은 SNP는 8,633개이다. 정상인인 CD4+ 샘플은 총 12,841개의 SNP가 검출되었으며, 이 중 dbSNP에 보고되지 않은 SNP는 5,067개로 Jurkat 샘플에서 CD4+ 샘플보다 보고되지 않은 유의한 SNP가 더 많이 추출되고 있다.

또한 mRNA 트랜스크립트 서열을 사용한 경우에는 Jurkat 샘플에서 59,116개의 SNP가 추출되었으며, CD4+ 샘플에서는 52,247개의 SNP가 추출되었으며, DNA 서열을 사용한 경우에 비하여 약 3.3-4.0배 더 많은 SNP를 추출한 것이다. 이는 같은 유전자에 속한 서로 다른 트랜스크립트 내에서 동일한 SNP가 중복되어 검출되기 때문이다. 즉, DNA 서열상의 임의의 엑손 위치에서 하나의 SNP가 추출되는 경우, mRNA 트랜스크립트 서열에서는 두 개 이상의 트랜스크립트 상에서 SNP가 동시에 추출될 수 있다. 그러나 mRNA 트랜스크립트 서열을 사용하는 경우 유전자 엑손 영역 단위가 아닌 트랜스크립트 단위로 SNP 출현 여부를 정확히 판단할 수 있는 장점이 있다.

두 번째 실험에서는 로컬 클라우드 컴퓨팅 환경을 사용하여 클라우드 컴퓨팅에 참여하는 노드 수의 증가에 따른 성능 향상 효과를 평가하였다. 최적화된 실험을 위해 모든 컴퓨팅 노드를 하나의 네트워크에 연결하고 분산 환경에서의 CloudSNP의 병렬 처리



(a) DNA 서열을 사용한 경우 (b) mRNA 트랜스크립트 서열을 사용한 경우

[그림 6] 노드 수 증가에 따른 연산 속도 향상 비율

성능을 평가하기 위하여 실험의 입력 데이터는 각 노드에 중복 저장되도록 설정하였다. 단, 입력 데이터는 맵 함수에 의해 처리되어진 후 키 값에 따라 수행되는 리듀스 함수가 결정된 후, 서로 다른 노드에 전송되기 때문에 다시 노드 간에 데이터 교환이 이루어진다. 노드의 수는 단일 노드부터 4개의 노드까지 증가하며 실험을 수행하였으며, 제안하는 알고리즘의 수행되는 시간과 HDFS에 데이터를 입출력하는 시간을 포함한 전체 수행 시간을 측정하여 비교하였다.

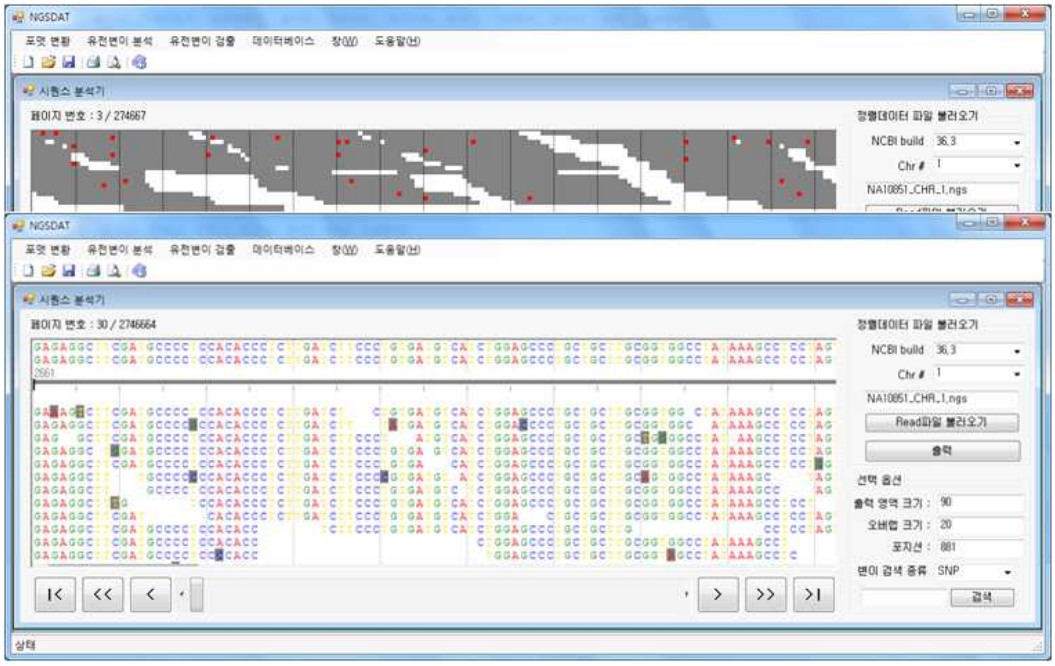
[그림 6]은 노드 수의 증가에 따른 제안한 CloudSNP 알고리즘의 처리 시간을 그래프로 표현한 것이다. 가로축은 노드수를 나타낸 것이며 세로축은 수행시간 (분 단위)을 나타낸 것이다. 그림 6(a)와 (b)는 각각 DNA 서열을 사용한 경우의 두 가지 샘플에 대한 SNP 추출 시간과 mRNA 트랜스크립트 서열을 사용한 경우의 두 가지 샘플에 대한 SNP 추출 시간을 비교하여 보인 결과이다.

실험 결과를 보면 DNA 서열을 사용한 경우에 비하여 mRNA 트랜스크립트 서열을 사용한 경우의 처리 시간이 더 길다. 이는 DNA 서열을 사용하였을 경우에는 유일하게 정렬된 경우 (unique match)만을 사용하기 때문에 3.27 GB와 3.58 GB의 정렬 결과 파일이 산출된데 비하여 mRNA 트랜스크립트 서열을 사용할 경우 중복 정렬 (all match)를 허용해야 하기 때문에 5.98 GB, 5.26 GB로 정렬 결과 파일의 크기가 상대적으로 더 크다. 따라서 SNP를 검출해야할 전체 데이터의 크기가 mRNA 트랜스크립트 서열을 사용한 경우가 더 크기 때문에 처리 시간이 더 길게 나오고 있다. 그러나 두 방식 모두 노드의 증가에 따라 처리 시간이 단축되고 있음을 알 수 있으며, 평균적으로 단일 노드의 처리 성능에 비하여 2, 3, 4 노드로 증가시킬 경우에 1.9, 2.47, 3.53배의 성능이 향상되고 있다.

4.2 시각적 분석 도구를 이용한 SNP 결과 검증

본 시스템에서는 SNP 결과 검증 및 분석을 지원하는 시각적 분석 도구인 Seq-Analyzer를 제공한다. Seq-Analyzer는 표준 서열에 정렬된 모든 리드들의 서열 정보를 구체적으로 확인/분석하는데 사용되고, 특히 SNP 검출 영역에 대한 리드 정렬 결과를 구체적으로 확인하여 SNP 결과를 검증하는데 사용될 수 있다. Seq-Analyzer는 기본적으로 리드의 염기 정보뿐만 아니라 리드의 품질 점수, 정렬 방향 등의 구체적 정보를 제공하고, dbSNP 등의 생물 정보

[그림 7] Seq-Analyzer의 리드 정렬 결과 분석 화면의 예



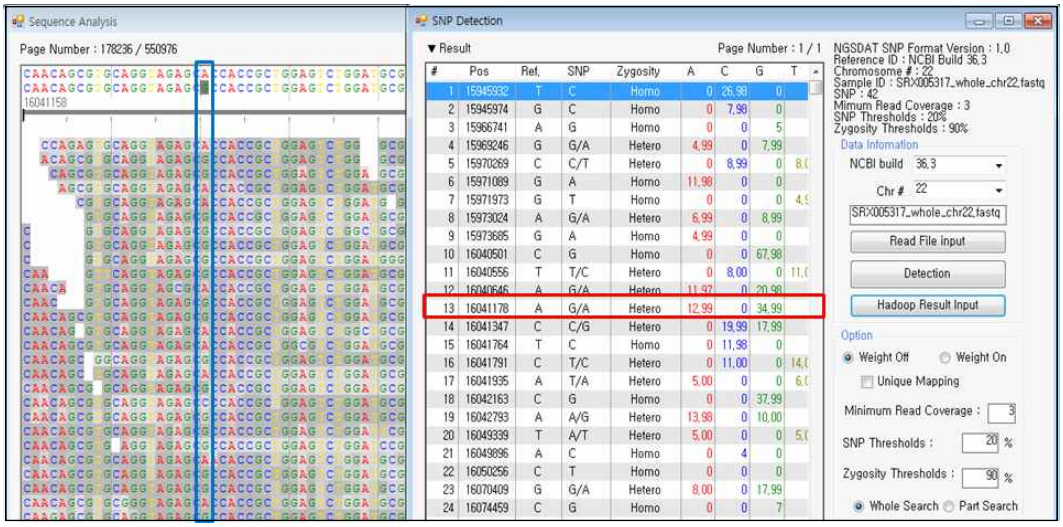
[그림 8] Seq-Analyzer의 화면 확대/축소 및 결과 분석 화면의 예

데이터베이스와 연동 가능하다.

Seq-Analyzer의 기능을 예를 통해 설명하면 다음과 같다. [그림 7]은 Seq-Analyzer의 리드 정렬 결과 검색 화면의 예를 보인다. [그림 7]에서 ①번 서열은 표준 서열의 정보를 나타내며, 그 아래 나열된 ②번 서열들은 표준 서열의 각 위치에 정렬된 모든 리드의 염기 중에서 정렬된 횟수가 제일 많은 리드 염기를 출력하여 나타낸 것이다. ③에 나열된 짧은 서열들은 정렬된 리드 서열을 나타낸다. 이와 같은 리드 정렬 결과를 브라우징하며 사용자는 각 표준 서열 염기 위치에 정렬된 염기 정보를 시각적으로 확인할 수 있다. 또한 그림에 보이는 바와 같이 서열상의 A, C, G, T 염기 문자를 사용자가 색상 지정을 통하여 각각 다른 색상으로 표현할 수 있으며, 리드의 방향 등을 함께 출력할 수 있도록 고려하여 시각적 분별 기능을 높였다. 그 외에도 [그림 8]의 하단 화면에 보이는 바와 같이 정렬된 각 리드 서열을 분석하여 표준 서열의 염기와 다른 종류의 염기가 발견되는 경우, 해당 염기를 음영 효과를 주어 표시함으로써 SNP 후보 영역을 간단히 구별할 수 있도록 지원하였다.

이와 같은 리드 정렬 결과에 대한 검색을 수행하는 경우, 검색 공간이 매우 넓어지므로 분석에 어려움이 발생할 수 있다. 따라서 [그림 8]의 상단 화면에 보이는 바와 같이 페이지 기능과 확대/축소 기능을 제공하여 정렬 결과를 개괄적으로 혹은 자세하게 분석할 수 있도록 지원한다. Seq-Analyzer는 페이지 단위의 데이터 처리를 기본적으로 지원하고 있지만, 리드들이 페이지와 페이지 경계면에 정렬되는 경우를 고려하여 중복 (overlap) 영역을 설정할 수 있도록 지원한다. 다음의 [그림 9]는 Seq-Analyzer에서 정렬된 각 리드를 선택하여 리드의 세부 정보

[그림 9] Seq-Analyzer의 리드 정보 출력 화면의 예



[그림 10] Seq-Analyzer의 SNP 결과 검증 화면의 예

를 검색하는 화면이다. 리드 정보에는 정렬된 각 리드들의 리드 ID, 리드 서열, 리드 품질 점수, 정렬 위치, 정렬 방향, 부정 적합 (mis-match) 등의 세부 정보가 포함되어 있어 리드 단위로 정렬 결과를 보다 정확히 검증할 수 있다.

다음의 [그림 10]은 Seq-Analyzer에 의한 SNP 결과 검증 화면의 예를 보인다. 그림에서 보이는 바와 같이 본 시스템에서는 SNP 분석을 위한 사용자 지정 매개변수로 가중치 적용 여부, 리드 정렬 결과 필터링 방식, 품질 임계값, 정렬 임계값, 접합자 구조 임계값 등을 입력 받을 수 있도록 지원하고 있어, 사용자 목적에 맞는 유연성 있는 SNP 결과 검증이 가능하다. [그림 10]의 오른쪽 화면은 CD4+ 샘플을 DNA 서열에 정렬하여 얻어진 SNP 추출 결과의 일부를 출력한 예를 보인다. 여기에서 각 SNP 추출 결과는 앞에서 설명한 리드 정렬 결과 검색 화면과 연동하여 표시 가능하다. 그림에 보이는 바와 같이 빨간색 영역으로 표시한 SNP 추출 결과는 리드 정렬 결과 분석 화면의 파란색 영역으로 표시된 부분과 연동하여 표시 가능하다. 따라서 이 들 결과를 상호 비교함으로써 SNP 추출 과정 및 근거를 시각적으로 확인/검증할 수 있다.

5. 결론 및 향후 연구

본 논문에서는 대규모 RNA 데이터를 처리하기 위한 클라우드 컴퓨팅 기술 기반의 병렬 SNP 추출 알고리즘 CloudSNP를 제안하였다. 제안하는 병렬 SNP 추출방법은 맵리듀스 방식을 이용하여 표준 서열에 정렬된 리드의 염기 분포, 리드 품질 점수 등의 정보를 활용하여 SNP 영역을 추출한다. 현재는 클라우드 컴퓨팅 환경에서 보다 최적화된 로드 밸런싱 효과를 구현하기 위한 알고리즘 개선 방식에 관한 연구를 진행하고 있으며, 향후 시각화 분석 도구인 Seq-Analyzer를 통한 SNP 검증 기능을 강화할 예정이며, 또한 상용 클라우드 컴퓨팅 서비스를 이용한 대규모 병렬 컴퓨팅환경에서의 성능 개선 효과에 대한 분석 연구를 수행할 예정이다.

7. 참고문헌

[1] M. L. Metzker, "Sequencing technologies — the next generation," *Nature Reviews Genetics*, vol.11, pp.31−46, 2010.

[2] R. Redon et al., "Global variation in copy number in the human genome," *Nature*, vol.444, no.7118, pp.444−454, 2006.

[3] http://www.illumina.com

[4] http://www.genome−sequencing.com

[5] http://www.appliedbiosystems.com

[6] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Sixth Symposium on Operating System Design and Implementation*, 2004

[7] T. White, "Hadoop : The Definitive Guide," *O'REILLY*, 2009.

[8] Langmead et al., "Searching for SNPs with cloud computing," *Genome Biology*, vol.10, no.11, 2009.

[9] M. C. Schatz, "CloudBurst: Highly Sensitive Read Mapping with MapReduce," *Bioinformatics*, vol.25, no.11, pp.1363−1369, 2009.

[10] A. Matsunaga, M. Tsugawa and J. Fortes, "CloudBLAST: Combining MapReduce and virtualization on distributed resources for bioinformatics applications," *4th IEEE International Conference on e−Science*, pp.222−229, 2008.

[11] B. Giardine et al., "Galaxy: a platform for interactive large−scale genome analysis," *Genome research*, vol.15, no.10, pp.1451−1455, 2005.

[12] B. Langmead, K. Hansen, J. Leek, "Cloud−scale RNA−sequencing differential expression analysis with Myrna," *Genome Biology*, vol.11, no.8, 2010.

[13] A.D. Smith et al., "Updates to the RMAP short−read mapping software," *Bioinformatics*, vol.25, no.21, pp.2841−2842, 2009.

[14] B. Langmead et al., "Ultrafast and memory−efficient alignment of short DNA sequences to the human genome," *Genome biology*, vol.10, no.3, R25, 2009.

[15] R. Li et al., "SNP detection for massively parallel whole−genome resequencing," *Genome Research*, vol.19, no.6, pp.1124−1132, 2009.

[16] J. Adams, "The Proteome: Discovering the Structure and Function of Proteins," *Nature Education*, vol.1, no.3, 2008.

[17] 공진화 외 4명, "mRNA 리드 시퀀스 데이터를 이용한 선택 스플라이싱 유형 분석," *정보과학회 논문지*, 제 38권, 제 6호, pp.351−358, 2011.

[18] A. Mortazavi et al., "Mapping and quantifying mammalian transcriptomes by RNA−Seq," *Nature Methods*, vol.5, no.7, pp.621−628, 2008.

[19] C. Trapnell, L. Pachter, S.L. Salzberg, "TopHat: discovering splice junctions with RNA−Seq," *Bioinformatics*, vol.25, no.9, pp.1105−1111, 2009.

[20] K. F. Au et al., "Detection of splice junctions from paired−end RNA−seq data by SpliceMap," *Nucleic Acids Research*, vol.38, no.14, pp.4570−4578, 2010.

[21] T. D. Wu and S. Nacu, "Fast and SNP−tolerant detection of complex variants and splicing in short reads," *Bioinformatics*, vol.26, no.7, pp.873−881, 2010.

[22] C. Trapnell, B. A. Williams, G. Pertea, "Transcript assembly and quantification by RNA−Seq reveals unannotated transcripts and isoform switching during cell differentiation," *Nature Biotechnology*, vol.28, no.5, pp.511−515, 2010.

[23] I. Chepelev et al., "Detection of single nucleotide variations in expressed exons of the human genome using RNA−Seq," *Nucleic Acids Research*, vol.37, no.16, pp.e106, 2009.

[24] R. Li et al., "SOAP: short oligonucleotide alignment program," *Bioinformatics*, vol.24, no.5, pp.713−714, 2008.

[25] H. Li and R. Durbin, "Fast and accurate short read alignment with Burrows−Wheeler

transform," *Bioinformatics*, vol.25, no.14, pp.1754−1760, 2009.

[26] H. Li, J. Ruan and R. Durbin, "Mapping short DNA sequencing reads and calling variants using mapping quality scores," *Genome Research*, vol.18, no.11, pp.1851−1858, 2008.

[27] P. J. A. Cock et al., "The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants," *Nucleic Acids Research*, vol.38, no.6, pp.1767−1771, 2010.

[28] T. Barrett and R. Edgar, "Reannotation of array probes at NCBI's GEO database," *Nature Methods*, vol.5, no.2, p. 117, 2008.

[29] S. T. Sherry et al., "dbSNP: the NCBI database of genetic variation," *Nucleic Acids Research*, vol.29, no.1, pp.308−311, 2001.

김 덕 근

2010년 2월 한림대학교 컴퓨터공학과 졸업(학사)

2012년 2월 한림대학교 컴퓨터공학과 졸업(석사)

2012년 3월 - 현재 마크로젠

관심분야 : 바이오인포매틱스, 데이터베이스

e−mail : dklovesky@hallym.ac.kr

윤 지 희

1982년 2월 한양대학교 전자공학과 졸업(학사)

1985년 3월 일본 구주대학교 정보공학과 졸업(석사)

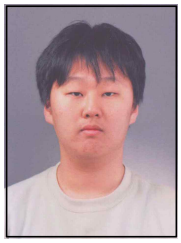
1988년 3월 일본 구주대학교 정보공학과 졸업(박사)

1998년 3월 - 1999년 2월 미국 UCLA 전산학과 방문교수

1988년 3월 - 현재 한림대학교 컴퓨터공학과 교수

관심분야 : 데이터 마이닝, XML, 바이오 정보처리, 시계열 데이터베이스

e−mail : jhyoon@hallym.ac.kr



홍 상 균

2005년 2월 한림대학교 정보통신공학부 졸업(학사)

2007년 2월 한림대학교 컴퓨터공학과 졸업(석사)

2007년 3월 - 현재 한림대학교 컴퓨터공학과 박사과정

관심분야 : 데이터베이스 시스템, 서열 정렬, 바이오 정보처리, 클라우드 컴퓨팅

e−mail : kyoons@hallym.ac.kr