DOI: 10.3745/KIPSTD.2009.16D.5.661

정렬된 리드의 통계적 분석을 기반으로 하는 CNV 검색 알고리즘

홍 상 균[†] · 홍 동 완[††] · 윤 지 희[†††] · 김 백 섭[††††] · 박 상 현[†††††]

요 약

인간의 유전체 서열에는 유전체 단위반복변위(copy number variation, CNV)를 포함하는 다양한 유전적 구조 변이(genetic structural variation)가 존재하며, 이는 기능적으로 질병에 대한 감수성, 치료에 대한 반응, 유전적 특성 등과 밀접한 관련이 있다. 본 논문에서는 기가 시퀀싱(giga sequencing)의 결과 산출되는 대량의 짧은 길이의 DNA 서열 데이터를 이용한 새로운 CNV 검색 방식을 제안한다. 제안하는 알고리즘에서는 레퍼런스 시퀀스에 DNA 서열 데이터를 서열 정렬시켜 각 레퍼런스 시퀀스의 위치에 대한 서열 데이터의 출현 빈도 정보를 얻은 후, 출현 빈도 정보의 패턴을 분석하여 통계적 유의성을 갖는 1kbp 이상의 연속 영역을 CNV 후보 영역으로 추출한다. 또한 제안된 알고리즘을 효율적으로 지원하기 위한 서열 정렬 방식에 대한 비교 및 분석을 수행한다. 제안된 기법의 유용성을 규명하기 위하여 다양한 실험을 수행하였다. 실험 결과에 의하면, 제안된 기법은 비교적 낮은 커버리지의 기가 시퀀싱 데이터를 이용하여 반복되거나 결실되는 다양한 형태의 CNV 영역을 효율적으로 검출하며, 또한 작은 사이즈의 CNV 영역에서부터 큰 사이즈의 CNV 영역까지 다양한 크기의 CNV 영역을 효율적으로 검출할 수 있는 것으로 나타났다.

키워드 : 유전체 단위반복변위(CNV), 기가 시퀀싱, 서열 정렬, 통계적 유의성

A CNV detection algorithm based on statistical analysis of the aligned reads

Sang-Kyoon Hong[†] · Dong-Wan Hong[††] · Jee-Hee Yoon[†††] · Baek-Sop Kim[††††] · Sang-Hyun Park[†††††]

ABSTRACT

Recently it was found that various genetic structural variations such as CNV(copy number variation) exist in the human genome, and these variations are closely related with disease susceptibility, reaction to treatment, and genetic characteristics. In this paper we propose a new CNV detection algorithm using millions of short DNA sequences generated by giga-sequencing technology. Our method maps the DNA sequences onto the reference sequence, and obtains the occurrence frequency of each read in the reference sequence. And then it detects the statistically significant regions which are longer than 1Kbp as the candidate CNV regions by analyzing the distribution of the occurrence frequency. To select a proper read alignment method, several methods are employed in our algorithm, and the performances are compared. To verify the superiority of our approach, we performed extensive experiments. The result of simulation experiments (using a reference sequence, build 35 of NCBI) revealed that our approach successfully finds all the CNV regions that have various shapes and arbitrary length (small, intermediate, or large size).

Keywords : Copy Number Variation(CNV), Giga-Sequencing, Sequence Alignment, Statistical Significancy

1. 서 론

2002년에 초안이 발표된 휴먼 게놈 프로젝트(human genome project: http://www.genome.gov/10001772)는 인간의 서열 정보 해석을 기반으로 하는 질병의 예측 및 치료 연구를 위한 초석이 되었다. 이 들 유전체(genome) 분석을 위한 비용은 2000년 초에는 약 30억 달러 이상이 소요되었으나, 최근의 보고서에서는 2012년 이후에 1인당 유전체 분석 비

※ 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No.2009-0065503).
† 준 회 원 : 한림대학교 컴퓨터공학과 박사과정
†† 정 회 원 : 한림대학교 바이오메디컬학과 겸임조교수
††† 정 회 원 : 한림대학교 컴퓨터공학과 교수(교신저자)
†††† 정 회 원 : 한림대학교 컴퓨터공학과 교수
††††† 종신회원 : 연세대학교 컴퓨터과학과 부교수
논문접수 : 2009년 2월 9일
수 정 일 : 1차 2009년 5월 27일
심사완료 : 2009년 5월 27일

용이 1,000 달러 이하로 하락하여 개인 유전체 시퀀싱(personalized sequencing) 시대가 도래할 것을 예상하고 있다[1]. 이와 같은 인간의 유전체 정보 분석을 위한 DNA 시퀀싱(sequencing) 기술은 제 1세대의 Sanger 시퀀싱 시대를 거쳐 현재는 제 2세대인 기가 시퀀싱(giga-sequencing) 시대로 분류되고 있으며, 전 세계적으로 학계 및 사업체를 중심으로 유전체 분석 기술의 개발을 위한 연구가 활발히 진행되고 있다.

인간의 유전체는 약 30억bp(base pair)의 긴 서열 정보로 이루어져 있다. 그러나 각 개인의 서열 정보 사이에는 부분적 차이가 존재하며, 이러한 서열 정보의 부분적 차이가 유전적 특성을 나타내기도 하고 유전병의 발병 원인이 되기도 하는 것으로 알려져 있다[2]. 따라서 각 개인의 서열 정보가 완성되면 그 부분적 차이를 판별하여 개인의 특정 질병의 발병 가능성이나 유전적 결함을 예측하는 것이 가능하게 된다.

이러한 서열 정보간의 차이를 밝혀내기 위한 연구는 이미 10여년 전부터 시작되어 단일 염기 다형성(Single Nucleotide Polymorphism, SNP로 약칭) 프로젝트[3] 등이 수행되었다. SNP는 인간의 유전체 염기서열 중에서 단일 염기의 차이를 보이는 유전적 변화 또는 변이를 말하는 것으로 SNP 위치 검색을 위한 많은 연구가 진행되어 왔다. 그러나 최근 인간의 유전체 염기 서열 상에는 SNP와 같이 길이가 짧은 서열 정보의 차이뿐 아니라 수십kbp에서 수백kbp 이상의 긴 서열 영역에 걸쳐 구조적 변이가 발생할 수 있다는 연구결과가 보고되고 있다[4]. 일반적으로 유전적 변이를 크게 SNP와 같은 작은 영역의 시퀀스 미스매치(small sequence mismatch), 삽입(insertion), 삭제(deletion), 전이(inversion), 유전체 단위반복변이(copy number variation, CNV로 약칭) 등으로 구분한다[5]. 삽입, 삭제, 전이는 서로 다른 두 서열 정보를 비교하여 임의의 서브 시퀀스가 한 쪽의 서열에서 추가적으로 발견되거나 삭제되어 있거나 역방향으로 발견되어지는 경우를 나타낸다. 또한 CNV는 임의의 서브 시퀀스가 양쪽 서열에서 발견되는데 한쪽 서열에서 추가적인 카피(copy) 서열을 발견할 수 있는 경우를 나타내며, 일반적으로 발견이 어려운 유전적 구조 변이로 알려져 있다.

인간 유전체 서열에 존재하는 CNV 영역을 검출하기 위한 대표적인 방법으로서 마이크로어레이 기술을 이용한 방법[4,6]과 서열 비교 방법[7,8,9]을 들 수 있다. 마이크로어레이 기술을 이용한 방식은 oligonucleotide array 혹은 BAC array 등을 이용한 실험적 방식으로서 주로 수백kbp 이상의 큰 사이즈(large size)의 CNV 발견에 유용한 것으로 알려져 있다. 한편 서열 비교 방식은 기존에 밝혀진 서열 데이터를 상호 비교하는 계산적 접근법(computational approach)을 사용하는 방식이다. 이 방식은 마이크로어레이 기술을 이용하는 방식에 비하여 CNV 영역을 보다 정확하게 밝힐 수 있는 장점이 있어, 작거나 중간 정도 사이즈(small, intermediate size)의 CNV 발견 방법으로도 적용 가능한 것으로 알려져 있다. 그러나 서열 비교 방식을 적용할 수 있는 기존에 밝혀진 서열 데이터는 매우 제한적이며, 또한 새로운 서열 데이터를 얻기 위한 시퀀싱 작업은 아직 초고가의 비용을 필요로 한다.

본 연구에서는 기가 시퀀싱의 결과 산출되는 대규모 서열 데이터를 이용한 새로운 CNV 검색 방식을 제안한다. 기가 시퀀싱의 결과 생성되는 서열 데이터를 리드(read)라고 부르며, 리드는 그 길이가 수십bp에서 수백bp 정도로서 짧고, 그 수가 매우 많다는 특징을 갖는다. 본 논문에서 제안하는 방식은 레퍼런스 시퀀스에 리드를 서열 정렬 시킨 후, 정렬된 리드의 분포를 통계적으로 분석하여 CNV 영역을 검출하고자 하는 방식으로서 이 전에 시도된 바 없는 독창적인 방식이다.

제안하는 CNV 영역 검색 알고리즘은 후보 영역 추출 단계와 정제 단계로 이루어진다. 후보 영역 추출 단계에서는 레퍼런스 시퀀스에 리드를 서열 정렬시켜 각 레퍼런스 시퀀스의 위치에 대한 리드의 출현 빈도 정보를 얻는 후, 이동 평균 계수 k를 갖는 이동 평균 변환을 수행하여 잡음이 제거된 전반적인 출현 정보 패턴을 얻는다. 이 결과 얻어진 리드의 출현 빈도 정보는 Gaussian 분포를 가지며, 이로부터 통계적 유의성을 갖는 1kb 이상의 연속 영역을 CNV 후보 영역으로 추출한다. 정제 단계에서는 후보 영역에 대한 후처리 작업을 수행하여 정확한 CNV 영역을 추출하며, 반복 혹은 결실된 형태, 영역 크기 등 CNV의 특성 분석 결과를 함께 반환한다.

우리의 알고리즘은 간단하고 직관적이다. 그러나 알고리즘이 좋은 성능을 갖기 위하여 다음 두 가지 사항이 고려되어야 한다. 우선 첫 번째로 레퍼런스 시퀀스와 리드에 존재할 수 있는 시퀀싱 에러를 효율적으로 처리할 수 있는 서열 정렬 방식이 사용되어야 한다. 다음 두 번째로 유전체 서열의 반복(repeat) 영역 문제를 효율적으로 처리할 수 있는 서열 정렬 방식을 사용하여야 한다. 유전체 서열에는 수많은 반복 영역이 존재하며[10], 이 반복 영역에 의하여 발생하는 반복 출현 리드들은 CNV 영역 추출을 방해하는 요소로 작용한다. 특히, 리드의 길이가 짧은 경우, CNV 영역을 제외한 다른 반복 영역으로부터 추출된 동일 리드의 발생 확률이 높아지며, 이 들은 CNV 영역을 탐색하는데 가장 큰 장애로 작용한다. 본 연구에서는 서열 정렬 방식을 상호 비교, 분석하여 제안된 CNV 검색 방식을 효과적으로 지원하기 위한 최적의 서열 정렬 방식을 보인다.

제안된 방식의 유효성을 보이기 위하여 NCBI(National Center for Biotechnology Information)의 레퍼런스 서열 build 35를 사용한 시뮬레이션 실험을 수행하였다. 실험 결과에 의하면, 제안된 기법은 비교적 낮은 커버리지의 기가 시퀀싱 데이터를 이용하여 반복되거나 결실되는 다양한 형태의 CNV 영역을 효율적으로 검출하며, 또한 다양한 크기의 CNV 영역을 효율적으로 검출할 수 있 수 있는 것으로 나타났다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련 연구로서 기가 시퀀싱 기법의 현황 및 서열 정렬 방식에 대하여 설명하고, 기존의 서열 비교에 의한 CNV 검색 방식을 기술

한다. 제 3장에서는 CNV 영역 검색의 문제를 정의하고, 본 연구에서 제안하는 CNV 영역 검색 방식을 제시하며, 이를 위한 구체적 알고리즘을 보인다. 제 4장에서는 실험 결과에 의하여 제 3장에서 제안한 CNV 영역 검색 방법의 타당성을 보이고, 성능 평가 결과를 보인다. 마지막으로 제 5장에서는 본 논문을 요약하고, 결론을 내린다.

2. 관련 연구

본 장에서는 관련 연구로서 기가 시퀀싱 기법과 리드의 서열 정렬 방식에 대하여 기술하고, 기존의 CNV 발견 방식을 간단히 살펴본다.

2.1 기가 시퀀싱 기법

대표적인 시퀀싱 기술 보유 회사(<표 1> 참조)에 속하는 Solexa, 454 Life Science, NimbleGen 등의 머신은 비교적 적은 비용에 30~100bp 정도의 짧은 서열을 대규모로 생성해 낸다. 이와 같은 짧은 길이의 리드를 이용한 디노버(de novo) 시퀀스 어셈블리 방법으로서 SSAKE[11], VCAKE[12], SHARCGS[13] 등의 알고리즘이 알려져 있다. SSAKE는 서로 다른 두 리드 사이의 가장 긴 오버랩 영역을 접두어 트리를 사용하여 탐색하여, 이 들을 점진적으로 연결하여 콘티그(contig)들을 생성하여 가는 방식이며, VCAKE와 SHARCGS는 SSAKE와 유사한 방식으로서 어셈블리 과정 중 리드 상의 오류를 허용/보정하는 과정을 추가적으로 제공하고 있다. 그러나 이들 알고리즘을 사용하여 디노버 시퀀스 어셈블리를 수행하기 위하여는 비교적 짧은 시퀀스의 경우에도 수십에서 수백 커버리지(coverage)의 리드를 필요로 한다[11-13]. 즉, 리드의 길이가 짧은 이유로 정확성을 높이기 위하여 반복적인 실험의 수행이 필수적이며, 따라서 아직까지 인간 유전체의 시퀀싱 작업은 초고가의 실험으로 분류되고 있다.

〈표 1〉 기가 시퀀싱 플랫폼의 예[1]

Company	Format	Read Length (bases)	Expected Throughput MB(million bases)/day
454 Life Sciences	Parallel bead array	100	96
Agencourt Bioscience	Sequencing by ligation	50	200
Applied Biosystems	Capillary electrophoresis	1,000	3-4
Microchip Biotechnologies	Parallel bead array	850-1,000	7
NimbleGen Systems	Map and survey microarray	30	100
Solexa	Parallel microchip	35	500
LI-COR	Electronic microchip	20,000	14,000
Network Biosystems	Biochip	800+	5
VisiGen Biotechnologies	Single molecule array	NA	1,000

2.2 리드의 서열 정렬 방식

기가 시퀀싱을 지원하는 대부분의 시스템은 유전체 분석 소프트웨어를 지원한다. 예를 들어 대표적인 기가 시퀀싱 머신 중 하나인 Solexa는 서열 정렬 프로그램으로서 Eland를 제공한다[14]. Eland는 리드와 레퍼런스 시퀀스와의 서열 정렬을 수행하여 완전매치(exact match), k=1 미스매치(mismatch), k=2 미스매치에 성공한 리드들의 서열 정렬 위치를 결과로서 반환한다. 서열 정렬 프로그램은 3Gbp 정도의 대규모 레퍼런스 시퀀스와 수백만에서 수천만개의 리드 집합을 대상으로 하므로 일반적으로 큰 주기억장치 공간과 많은 처리 시간을 필요로 하며, 시퀀싱 에러의 영향을 최소화할 수 있는 정렬 방안을 마련하여야 한다. 서열 정렬 알고리즘에 대한 다양한 연구가 진행되고 있으며, SOAP[15], MAQ[16], RMAP[17], AQUESA[18]등의 프로그램이 알려져 있다. 이 들 프로그램은 서픽스 트리[19], 서픽스 어레이[20], 해쉬, q-gram 등의 다양한 인덱싱 기법을 적용하여 레퍼런스 시퀀스와 리드 집합의 처리 속도 향상을 꾀하고 있으며, 또한 리드를 구성하는 각 염기의 품질 스코어(quality score) 정보를 활용하여 서열 정렬의 정확도를 개선하고 있다.

2.3 기존의 CNV 발견 기법

CNV 영역을 검출하기 위한 방법을 크게 마이크로어레이 기술을 이용한 방법[4, 6]과 서열 비교 방법[7, 8, 9]으로 분류할 수 있다. 마이크로어레이 기술을 이용한 방식은 oligonucleotide array 혹은 BAC array 등을 이용한 실험적 방식이다. 이 방식은 비교적 저가의 실험 비용을 필요로 하여 일반적으로 많이 사용되는 방법이다. 단 noise에 약한 특성으로 인하여 주로 수백kbp 이상의 큰 사이즈(large size)의 CNV 발견에 유용한 것으로 알려져 있다.

서열 비교법은 기존에 완성된 어셈블리 시퀀스들을 상호 비교하여 인간 유전체에 존재하는 CNV 등의 구조적 변이를 찾아내는 방식이다[21]. 이 방식에서는 BLAST[22], BLAT[23] 등을 사용하여 두 서열 사이의 정렬 연산을 수행한 후, 이를 기반으로 서로 다른 두 서열 사이에 존재하는 차이점을 발견한다. 최근에 발표된 다음 3 가지 논문에서는 각각 서로 다른 서열들을 비교, 해석하여 인간 유전체 서열에 존재하는 구조적 변이를 밝히고 있다. 참고 문헌 [7]에서는 human fosmid paired-end sequence와 인간 유전체 어셈블리 시퀀스 (build 35)를 비교하여 8 Kbp 이상의 크기를 갖는 297개의 구조적 변이 후보 영역을 보고하고 있다. 참고 문헌 [8]에서는 HapMap 프로젝트에서 개발된 DNA re-sequencing trace sequence(WGS, TSC, WCS)와 인간 유전체 어셈블리 시퀀스(build 35)를 비교하여 1-9989 bp의 크기를 갖는 415,436개의 구조적 변이 후보 영역을 추출, 이를 보고하고 있다. 또한 참고 문헌 [9]에서는 서로 다른 2개의 인간 유전체 어셈블리 시퀀스를 비교하여 구조적 변이

영역을 추출하고 있다. 대표적인 2개의 어셈블리 시퀀스로서 간주되는 Celera Genomics의 어셈블리 시퀀스(R27c)와 IHGSC(International Human Genome Sequencing Consortium)의 레퍼런스 시퀀스(build 35)를 이용하였으며, 서열 비교를 수행하여 419개의 CNV 영역을 포함하는 총 13,534개의 구조 변이 후보 영역을 추출하여 보고하고 있다. 이 방식은 마이크로어레이 기술을 이용하는 방식에 비하여 CNV 영역을 보다 정확하게 밝힐 수 있는 장점이 있어, 작거나 중간 정도 사이즈(small, intermediate size)의 CNV 발견 방법으로도 적용 가능한 것으로 알려져 있다. 그러나 이들 방식은 어셈블리가 완성된 시퀀스를 비교 대상으로 하는 CNV 검색 방식으로서, 기가 시퀀싱의 결과 생성되는 짧은 길이의 리드를 그대로 적용 대상으로 할 수 없다.

3. CNV 영역 검색

본 장에서는 제안하는 CNV 영역 탐지 방법에 대하여 기술한다. 제 3.1절에서는 본 연구에서 해결하고자 하는 문제를 정의하고 CNV 영역 검색 방식을 간단히 설명한다. 다음 제 3.2절에서는 구체적 알고리즘을 보인다.

3.1 기본 전략

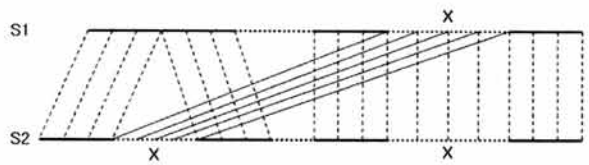
<표 2>는 본 논문에서 사용하는 기호를 보인다. 서열 비교법에 의하여 검색하고자 하는 CNV 영역은 다음과 같이 정의된다.

[정의 1: CNV 영역]

서로 다른 두 시퀀스 S1과 S2를 비교하여 임의의 서브 시퀀스 X가 양쪽 시퀀스에서 발견되는데 한쪽 시퀀스에서 추가적인 복사(copy)를 발견할 수 있으며, 서브 시퀀스 X의

〈표 2〉 기호 정의

기호	정의
$S=(s[i])$	데이터 시퀀스 $(0 \le i < Len(S))$, $Len(S)$는 S에 포함되는 요소 값의 수
$X=(x[i])$	S에 포함되는 임의의 서브 시퀀스 $(0 \le i < Len(X) \le Len(S))$
$R_j=(r_j[i])$	리드 $(0 \le i < Len(R_j))$, $(0 \le j < n)$, n은 리드의 개수
$F\_S=(f\_s[i])$	S의 각 위치에 대한 리드의 정렬 빈도 수를 표현하는 시퀀스 $(0 \le i < Len(S))$
$MV_m(F\_S)$	이동 평균 계수 m으로 이동 평균 변환한 F_S
C	테스트 시퀀스로부터 생성된 리드의 커버리지 수
m	이동 평균 변환 계수 $(1 \le m < Len(S))$
$Len(X)$	시퀀스 X에 포함되는 요소 값의 수.



(그림 1) CNV 영역의 예

크기가 1kbp(kilo base pair) 이상의 경우, 그 영역을 CNV 영역이라고 부른다[3]. □

정의 1에 보인 CNV 영역의 예를 그림 1에 보인다. 즉, 두 어셈블리 시퀀스 S1과 S2에 유사한 서브 시퀀스 X가 존재하고, 동시에 서열 S2에는 같은 영역이 두 번 나타나는 예를 나타내며, 반복 출현한 서브 시퀀스 X를 CNV 영역이라고 부른다.

본 연구에서는 비교 대상의 두 시퀀스로서 이미 시퀀싱이 완성되어 특성이 잘 알려진 표준의 레퍼런스 시퀀스와 임의의 테스트 시퀀스를 사용한다. 그러나 테스트 시퀀스는 시퀀싱이 완성되지 않은 형태로서, 기가 시퀀싱의 결과 얻어진 리드 집합을 그대로 사용하는 것을 가정한다. 이와 같은 가정 하에 테스트 시퀀스 상에 존재하는 CNV 영역을 검색하는 방법을 개발하고자 한다.

제안하는 방식에서는 레퍼런스 시퀀스에 리드를 서열 정렬 시킨 후, 서열 정렬된 리드의 분포를 통계적으로 분석하여 CNV 영역을 추출하고자 한다. 이를 위한 단계별 과정은 다음과 같다.

[정의 2: 리드의 서열 정렬 연산]

다음 식과 같이 레퍼런스 시퀀스 S 상에서 각 리드 R_j와 e 이하의 유사도를 갖는 모든 유사 서브 시퀀스 $S[p \dots p+Len(R)-1]$를 검색한다. 그 결과로서 R_j, S 상의 정렬위치 $[p \dots p+Len(R)-1]$, 해당 유사도를 반환한다. 단, 모든 리드의 길이는 $Len(R)$로 동일하다고 가정한다.

$S[p \dots p+Len(R)-1] \simeq R_j[0 \dots Len(R)-1] \quad 0 \le p < Len(S)$, $0 \le j < n$, n은 리드의 개수.

여기에서 A≃B는 동일한 길이의 두 서브 시퀀스 A와 B가 e 이하의 유사도를 갖는 유사 서브 시퀀스임을 나타낸다. □

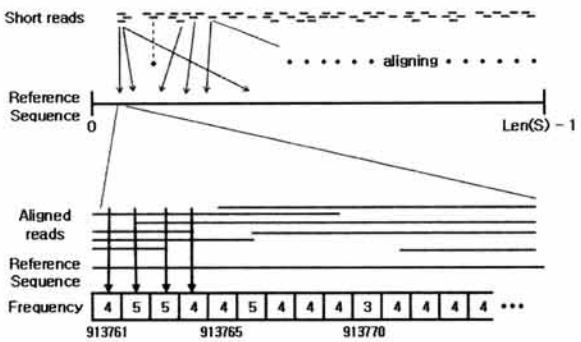
정의 2와 같이 레퍼런스 시퀀스에 대한 리드의 서열 정렬 연산을 수행한다. 여기에 사용되는 유사도 e의 값으로서 완전매치 혹은 k-미스매치 등을 지정할 수 있다.

[정의 3: 리드의 서열 정렬 빈도]

레퍼런스 시퀀스 S에 대한 각 리드의 서열 정렬 빈도 $F_S=(f_s[i], 0 \le i < Len(S))$는 다음과 같이 정의된다.

$$f\_s[i]=\sum_{j=0}^{n-1} Sc_j \begin{cases} Sc_j=1, & R_j\text{가 } s[i]\text{에 서열 정렬된 경우} \\ Sc_j=0, & R_j\text{가 } s[i]\text{에 서열 정렬되지않은 경우} \end{cases}$$ □

정의 2에 보인 리드의 서열 정렬 연산 결과를 이용하여 정의 3의 리드의 서열 정렬 빈도를 계산한다. 여기에서 리드의 서열 정렬 빈도 $f_s[i]$는 S의 i번째 위치에 정렬된 리드의 총 수를 의미한다. 이 과정을 도식적으로 표현하면 (그



(그림 2) 리드의 서열 정렬 빈도 계산 예

림 2)와 같이 나타낼 수 있다. 첫 번째 과정은 리드를 레퍼런스 시퀀스에 정렬시키는 과정을 나타내며, 다음은 레퍼런스 시퀀스의 각 위치에 정렬된 리드의 수를 계산하는 과정을 나타내고 있다.

그러나 이와 같이 얻어진 레퍼런스 시퀀스에 대한 리드의 출현 빈도로부터 직접 리드의 출현 변화 패턴을 파악하기 어렵다. 그 이유는 시퀀싱 에러 및 커버리지 수의 부족으로 인하여 리드의 출현 빈도에 그 값이 0이거나 매우 큰 값이 자주 발생하기 때문이다. 따라서 이와 같은 잡음(noise)의 영향 없이 전체적인 변화 패턴을 파악하기 위하여 다음과 같은 이동 평균 변환을 수행한다.

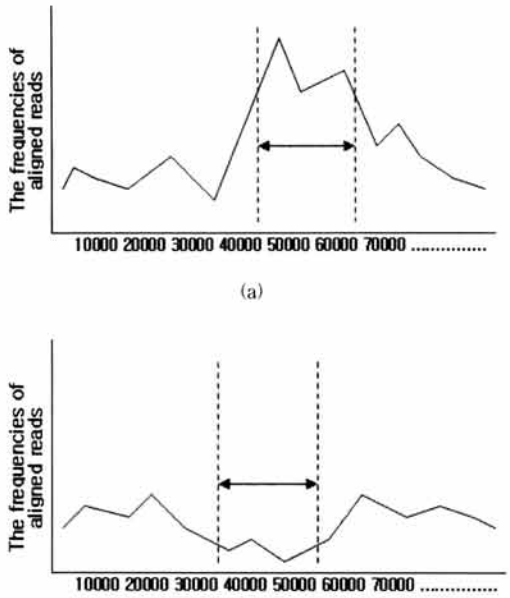
[정의 4: 이동 평균 변환]

서열 정렬 빈도 $F_S=(f_s[i])(0 \le i < Len(s))$를 이동 평균 계수 $m(1 \le m < Len(S))$으로 이동 평균 변환한 $MV_m(F_S)$를 구한다. 단, 여기에서는 이동 평균 계수를 $(m=2n+1)$의 홀수로 가정하며, $MV_m(F_S)=(f_s_m[i])\ (n < i \le Len(s)-n)$은 다음과 같이 정의된다[24].

$$f\_s_m[i]=\frac{1}{m}\times(f\_s[i-n]+\ldots+f\_s[i]+\ldots f\_s[i+n]) \quad \square$$
$$=\frac{1}{m}\times\sum_{i=i-n}^{i+n} f\_s[i]$$

정의 4에 나타난 바와 같이 이동 평균 변환은 연속되는 m개의 요소 값들의 평균값들을 순차적으로 나열하는 변환이다. 이동 평균 변환을 통하여 시퀀스 내에서 나타나는 잡음의 영향을 제거할 수 있으며, 이동 평균 변환 계수 m은 해당 응용에서 잡음의 영향을 줄이고자 하는 정도에 따라 선택된다.

이동 평균 변환된 리드의 출현 빈도 $MV_m(F_S)$는 그 값의 분포가 Gaussian 분포[24]를 따른다. 따라서 리드의 출현 변화 패턴을 나타내는 $MV_m(F_S)$로부터 CNV 영역을 나타내는 특이 패턴을 검색하는 문제는 Gaussian 분포를 갖는 $MV_m(F_S)$로부터 통계적으로 유의성이 있는 영역을 추출하는 문제로 바꾸어 생각할 수 있다. 즉, 이동 평균 변환된 출현 빈도를 분석하여 통계적으로 유의성이 있는 연속 영역을 추출하며, 그 연속 영역의 크기가 1kbp이상인 경우 CNV 후보 영역으로 한다.



(그림 3) CNV 후보 영역의 예

직관적 이해를 위하여 레퍼런스 시퀀스의 각 위치에 대한 리드 출현 회수를 그래프로 나타낸 가상의 예를 그림 3에 보인다. (그림 3)-(a)는 특정 영역의 빈도가 상대적으로 높은 경우를 나타내며, 이는 레퍼런스 시퀀스에 비하여 테스트 시퀀스에 임의의 영역이 여러 번 출현하여 많은 리드가 생성되었을 가능성을 나타낸다. 다음 (그림 3)-(b)는 이와 반대의 경우를 가정한 빈도 그래프를 나타낸다. 즉 우리는 이와 같은 특정 영역을 자동 검색하여 CNV 후보 영역으로 추정한다.

레퍼런스 시퀀스 S와 리드 집합 $\{R_j\}$을 이용한 CNV 후보 영역 추출 과정을 단계별로 보이면 다음과 같다.

step 1: 레퍼런스 시퀀스 S에 리드 집합 $\{R_j\}$을 서열 정렬하여 리드의 서열 정렬 빈도 정보 $F_S=(fs[i], 0 \le i < Len(S))$를 구한다.

step 2: 이동 평균 변환을 수행하여 $MV_m(F\_S)=(f\_s_m[i])$ $(n < i \le Len(S)-n, m=2n+1)$를 구한다.

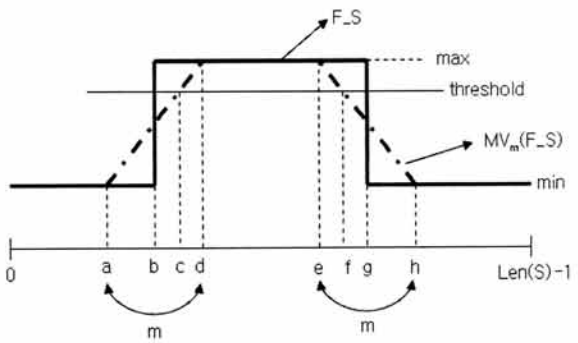
step 3: $MV_m(F_S)$로부터 통계적으로 유의성을 갖는 영역을 추출하고 그 연속 영역의 크기가 1kbp이상의 경우, CNV 후보 영역으로 추출한다.

앞에서 기술한 바와 같이 시퀀싱 에러 및 커버리지 수의 부족으로 인하여 계산 방식만을 이용하여 정확한 CNV 영역을 추출하기는 매우 어렵다. 그러나 제안된 CNV 후보 영역을 기반으로 보다 정확한 영역을 추정하기 위하여 다음과 같은 후처리 과정을 생각할 수 있다. 다음의 (그림 4)는 잡음이 없는 가상의 리드의 출현 빈도를 가정하여 CNV 후보 영역을 추출하는 과정을 나타낸다. 여기에서 실선으로 보이는 F_S는 리드의 출현 빈도를 나타내며, 파선으로 보이는 $MV_m(F_S)$는 이동 평균 변환된 값을 나타낸다. 또한

$MV_m(F_S)$의 값의 분포에 대하여 $p-value = 0.05$를 유의값으로 설정하여 임계치를 (threshold) 정한 예를 보인다. 이 예에 의하면 정확한 CNV 영역은 $S[b \ldots g]$에 위치하고 있으나, 제안된 방식은 CNV 후보 영역으로 $S[c \ldots f]$를 추출하게 된다. 즉 추출된 CNV 후보 영역은 CNV 영역에 모두 포함되나, $S[b \ldots c]$와 $S[f \ldots g]$의 영역을 찾지 못할 수 있다. 이를 보정하기 위하여 max 값과 min 값을 이용하여 다음 (식 1), (식 2)와 같이 b와 g의 값을 추정한다. 또한 해당 영역의 모니터링 및 분석을 지원하는 툴의 사용이 추정 값을 보다 정확히 보완할 수 있다.

$$b = c - \frac{threshold - \frac{\max + \min}{2}}{\frac{\max - \min}{m}} \quad \ldots\ldots \quad (식\ 1)$$

$$g = f + \frac{threshold - \frac{\max + \min}{2}}{\frac{\max - \min}{m}} \quad \ldots\ldots \quad (식\ 2)$$



$S[b \ldots g]$: CNV 영역
$S[c \ldots f]$: 추출된 CNV 후보영역

(그림 4) CNV 후보 영역을 위한 후처리 과정

3.2 CNV 영역 검색 알고리즘

CNV 영역을 검색하기 위한 알고리즘 FIND_CNV를 Algorithm 1에 보인다. Algorithm 1은 리드 집합 R과 레퍼런스 시퀀스 S를 입력으로 받아 CNV 영역을 찾아낸다. Algorithm 1의 동작 과정을 단계별로 설명하면 다음과 같다.

우선, 레퍼런스 시퀀스 상의 각 위치에 대한 리드의 출현 회수를 기록하기 위한 배열 FreqArray를 설정하고, 이를 초기화한다(line 1). 이때 FreqArray 배열의 크기는 레퍼런스 시퀀스의 길이와 같다. 다음, 리드 검색을 효율적으로 수행하기 위하여 함수 Sort&Count()를 호출하여 중복이 제거된 리드의 집합 RC를 구한다(line 2). 입력으로 주어진 리드의 집합 R에는 수많은 리드가 들어 있으며, 그 중에는 중복된 리드가 존재할 수 있다. 따라서 중복되어 나타나는 리드에 대한 과정을 최적화하기 위하여 함수 Sort&Count()에서는 리드에 대하여 정렬(sorting)을 수행하여 중복 출현하는 리드를 소거하고, 중복 리드들의 출현 회수를 카운팅하여 저장한다. 여기에서 RC[i].SR은 중복이 제거된 리드를 나타내며, RC[i].Num은 RC[i].SR의 중복 출현 회수를 나타낸다.

Algorithm 1: FIND_CNV : 영역 검색 알고리즘

Input : set of reads R, reference sequence S, similarity e, align type T1, repeat type T2, p-value P_V, moving average coefficient M
Output : set of CNV regions CNV

```
1.  Initialize frequency array FreqArray ;
2.  RC := Sort&Count(R) ;
3.  for each short read RC[i].SR of the RC do
4.      aligned position set P := AlignRead(S, RC[i].SR, e, T1);
5.      for each aligned position P[j] of the P do
6.          CalculateFreq(FreqArray, RC[i].Num, P[j], ReadLen,
            T2);
7.  MV_FreqArray := Transform(FreqArray, M);
8.  Signal := Find_Signal(MV_FreqArray, P_V);
9.  CNV := FindCNV_region(Signal);
10. return CNV;
```

다음 line 3-6은 각 리드에 대하여 서열 정렬 연산을 수행하여 레퍼런스 시퀀스와의 정렬 위치 P[j]를 구하고, 이를 기반으로 배열 FreqArray의 값을 얻는 과정을 나타낸다. 우선 함수 AlignRead()를 호출하여 리드의 레퍼런스 시퀀스 상의 정렬 위치 P[j]를 얻는다(line 4). 여기에서 사용된 함수 AlignRead()는 리드의 서열 정렬 연산을 수행하는 함수를 나타내며, 유사도 e, 서열 정렬 방식 T1에 의하여 서로 다른 서열 정렬 결과를 얻을 수 있다. 예를 들어 유사도 e의 값으로 완전 매치, k-mismatch 등을 지정할 수 있으며, T1의 값으로 all-match 혹은 best-match를 지정할 수 있다. 여기에서 all-match은 리드와 e 이하의 유사도를 갖는 모든 유사 서브 시퀀스를 검색하여 이를 정렬 위치에 포함 시키는 것을 의미하며, best-match는 리드와 가장 유사한 서브 시퀀스만을 검색하여 이를 정렬 위치에 포함 시키는 것을 의미한다. 예를 들어 e의 값으로 k=1 미스매치가 주어진 경우, all의 경우에는 완전 매치, k=1 미스매치를 만족하는 모든 유사 서브 시퀀스를 검색하여 해당 위치를 반환한다. 한편, best의 경우에는 만약 완전 매치를 만족하는 유사 서브 시퀀스가 검색되면 해당 위치만을 반환하고 그 이상의 유사 서브 시퀀스 검색을 수행하지 않는다. 레퍼런스 시퀀스와의 서열 정렬 결과, 각 리드는 레퍼런스 시퀀스에 전혀 출현하지 않거나, 유일하게 1번 출현 하거나 2번 이상 출현할 수 있다. 이와 같은 리드의 반복 출현 회수는 P[i].Count에 저장된다.

다음, 리드의 정렬위치 정보 P를 이용하여 각 리드의 정렬 위치에 대한 빈도수를 산출한다(line 5-6). 이 단계에서는 해당 리드가 정렬되어 나타난 구간(P[i]~P[i]+ReadLen-1)의 FreqArray 배열 값으로 현재의 값에 RC[i].Num와 P[i].Count의 값을 이용하여 빈도수를 계산한다. 이 때 반복 출현하는 리드에 대하여 빈도수를 산출하는 방법을 변수 T2에 의하여 지정한다. T2의 값으로 'read eliminate', 'random select', 'weighted method' 등을 지정할 수 있다. 'read eliminate'는 해당 리드가 여러 영역에 정렬되었을 경우 이를 반복 영역

에 의한 오류로 보고 빈도수 산정에서 제외시키는 방법이다. 'random select'는 정렬된 여러 영역 중 임의의 한 영역만 선택하여 빈도수 산출에 이용하는 방법이다. 'weighted method'는 한 리드가 여러 개의 영역에 정렬되었을 때 모든 영역을 빈도수 산출에 적용하며, 이 때 정렬된 영역의 수와 유사도 값에 따라 가중치를 부가하여 빈도수를 산출하는 방법이다.

다음의 line 7 - line 10은 빈도 정보 FreqArray를 이용한 CNV 영역을 추출하는 과정을 나타낸다. Line 7의 함수 transform()는 이동 평균 변환 계수 M의 이동 평균변환을 수행하여 그 결과를 새로운 배열 MV_FreqArray에 저장하는 함수를 나타낸다. 다음 line 8의 함수 Find_Signal()은 Gaussian 분포를 갖는 MV_FreqArray의 출현 빈도 정보로부터 통계적 유의성을 갖는 연속 영역을 검색하여 이들을 CNV 후보 영역으로 반환한다. 이때 통계적 유의한 영역을 찾아내는 기준은 p-value에 의존하며, 입력 값 PV로 주어진다. 다음, 함수 Find_CNV_region()은 후처리 작업을 수행하는 함수를 나타내며, 정확한 CNV 영역을 추출하고, 해당 CNV의 특성 분석 결과를 함께 반환한다.

4. 성능 평가

본 장에서는 제안하는 CNV 영역 검출 방법의 효용성을 시뮬레이션 실험을 통하여 검증한다. 제 4.1절에서는 실험 환경 및 방법에 대하여 기술하고, 제 4.2절에서는 다양한 실험 결과를 분석하여 제안하는 방법의 성능을 평가한다.

4.1 실험 방법

본 실험에서는 NCBI Build 35[25]의 시퀀스 일부를 레퍼런스 시퀀스로서 사용하고, 동일 시퀀스 영역에 보고된 CNV 영역을 추가로 삽입/삭제하여 테스트 시퀀스를 생성한다. 실험에 사용된 CNV 영역에 대한 정보는 Database of Genomic Varaint[26]의 CNV 데이터베이스에 수록된 정보를 사용한다. 레퍼런스 시퀀스로 인간 크로모좀 20번의 NT_028392.5의 콘티그를 사용하였으며, 이 콘티그에는 총 28개의 CNV 영역이 보고되어 있다. 보고된 CNV 영역의 위치 및 크기를 표 3에 보인다. CNV는 영역의 크기에 따라 수kbp 정도의 작은 사이즈의 CNV, 수십kbp 정도의 중간 사이즈의 CNV, 수백kbp를 넘는 큰 사이즈의 CNV로 분류된다[8].

시뮬레이션 실험에 사용된 테스트 시퀀스로부터 리드를 추출하는 방법은 다음과 같다. 테스트 시퀀스로부터 랜덤하게 선택된 위치에서 Solexa machine[14]의 성능과 유사하게 36bp의 리드를 추출한다. 이때 오류율 3%의 리드를 생성하였으며, 이는 일반적인 기가 시퀀싱 머신에서 발생할 수 있는 오류율과 유사한 정도이다. 이때 생성하는 리드의 개수는 1.6 커버리지의 리드와 3.2 커버리지의 리드를 사용하였다.

리드의 서열 정렬을 위한 프로그램으로서 SOAP(Short Oligonucleotide Alignment Program)[15]과 접미어 트리 인덱스[27]를 기반으로 하는 서열 정렬 프로그램을 사용하였다. 제 3.2절의 Algorithm 1에서 보인 바와 같이 서열 정렬 방법으로서 best-match와 all-match의 방식을 생각할 수 있다. SOAP은 best match의 결과만을 제공하므로 자체 개발된 서열 정렬 프로그램은 all match의 결과를 제공하도록 구현하였다. 또한 본 실험의 모든 정렬과정에서는 리드의 오류를 감안하여 유사도 e의 값으로 k=2 미스매치를 설정하였다.

반복 출현하는 리드에 대하여 빈도수를 산출하는 방법으로 다음 5가지의 방법을 사용하였다. 이 들 방식은 제 3.2절에 보인 Algorithm 1에서 함수 AlignRead()와 CalculateFreq()의 입력 변수 T1, T2의 값에 의하여 선택되는 방식이다.

(1) 최적리드 소거법(BE) : T1, T2의 값으로 각각 'best-match'와 'read eliminate'을 지정한 경우이다. 최적의 유사도를 갖는 영역을 검색하여, 그 결과 리드가 여러 영역에 정렬되었을 경우 이를 반복 영역에 의한 오류로 보고 빈도수 산정에서 제외시키는 방법이다. Solexa의 Eland 등 초기의 서열 정렬 프로그램에서는 반복 영역의 문제를 해결하기 위하여 반복 출현하는 리드를 서열 정렬 결과에서 제외시키는 방식을 주로 사용하였다. 본 논문에서는 이 방식을 BE 방식으로 부른다.

(2) 최적리드 임의 선택법 (BR): T1, T2의 값으로 각각 'best-match'와 'random select'를 지정한 경우이다. 최적의 유사도를 갖는 영역을 검색하여, 그 결과 리드가 여러 영역에 정렬되었을 경우 정렬된 여러 영역 중 임의의 한 영역만 선택하여 빈도수 산출에 이용하는 방법이다. 본 논문에서는 이 방식을 BR 방식으로 부른다.

(3) 최적리드 가중치 기법(BW) : T1, T2의 값으로 각각 'best-match'와 'weighted method'를 지정한 경우이다. 최적의 유사도를 갖는 영역을 검색하여, 그 결과 리드가 n개 영역에 정렬되었을 경우, 영역의 각 위치에 대한 가중치를 1/n로 계산하는 방법이다. 본 논문에서는 이 방식을 BW 방식으로 부른다.

(4) 최대리드 가중치 기법(AW) : T1, T2의 값으로 각각 'all-match'와 'weighted method'를 지정한 경우이다. 해당 리드와 k=2 미스매치 이하의 유사도를 갖는 영역을 모두 검색하여, 그 결과 리드가 총 n개 영역에 정렬되었을 경우, 영역의 각 위치에 대한 가중치를 1/n로 계산하는 방법이다. 본 논문에서는 이 방식을 AW 방식으로 부른다.

(5) 최대리드 차등가중치 기법(ADW) : 최대리드 가중치 기법과 같이 T1, T2의 값으로 각각 'all-match'와 'weighted method'를 지정한 경우이다. 그러나 해당 리드와 k=2 미스매치 이하의 유사도를 갖는 영역을 모두 검색하여 그 결과 리드가 총 n개 영역에 정렬되었을 경우, 각 영역의 유사도에 따라 가중치를 차등적으로 부가하여 빈도수를 산출한다. 본 실험에서는 k=2 미스매치 영역에 비하여 k=1 미스매치 영역에 2배 더 큰 가중치를 적용하고 완전 매치의 영역에 4배 더 큰 가중치를 적용하여 빈도수를 산출한다. 본 논문에서는 이 방식을 ADW 방식으로 부른다.

〈표 3〉 NT_028392.5의 콘티그에 보고된 CNV 영역[26]

No.	position		length	No.	position		length
	start	end			start	end	
1	3187	6588	3402	15	2104729	2111517	6789
2	29540	74297	44758	16	2104733	2111517	6785
3	238577	246878	8302	17	2104735	2111517	6783
4	413991	417801	3811	18	2518924	2522870	3947
5	674368	677790	3423	19	3011306	3015101	4796
6	835522	848469	12948	20	3011625	3015430	3806
7	907275	1084684	177410	21	3012030	3015327	3298
8	1362036	1370413	8378	22	3437952	3440376	2425
9	1539215	1691890	152676	23	3711584	3897145	185562
10	1619279	1628354	9076	24	4171957	4179628	7672
11	1683320	1689657	6338	25	4188353	4194352	6000
12	1691899	1944258	252360	26	4355911	4364450	8540
13	1712694	1734610	21917	27	4721986	4875518	153533
14	1963517	2032555	69039	28	4941453	4976444	34992

본 실험에서는 이동 평균 변환 계수 m의 값으로 1000을 사용하였다. 실험을 위한 플랫폼으로는 64비트 Linux(Kernel Version 2.6.26)를 운영체제로 사용하고, 8GB의 주기억 장치, 640GB 디스크를 갖는 Core2Quad 2.83GHz의 PC를 사용한다. 통계처리를 위한 소프트웨어 로서 R 패키지 2.8을 사용한다.

4.2 실험 결과 및 분석

4.2.1 실험 1

실험 1에서는 제안하는 방법이 다양한 패턴의 유전적 구조 변이 영역과 구별하여 CNV 영역을 효율적으로 추출 가능한지를 검증한다. 검증을 위하여 레퍼런스 혹은 테스트 시퀀스에 삽입/삭제하는 서브 시퀀스 X의 크기는 20kbp이다. 또한 테스트 시퀀스로부터 생성되는 리드의 커버리지 수 C는 1.6을 사용한다. CNV를 포함하는 유전적 구조 변이 영역의 예로서 다음 5가지 경우에 대하여 실험을 수행하였다.

(a) case 1 : 레퍼런스 시퀀스와 테스트 시퀀스에 유사한 서브 시퀀스 X가 존재하고, 테스트 시퀀스에 시퀀스 X의 카피가 1개 추가적으로 존재하는 경우.

(b) case 2 : 레퍼런스 시퀀스와 테스트 시퀀스에 유사한 서브 시퀀스 X가 존재하고, 레퍼런스 시퀀스에 시퀀스 X의 카피가 1개 추가적으로 존재하는 경우.

(c) case 3 : 레퍼런스 시퀀스와 테스트 시퀀스에 유사한 서브 시퀀스 X가 존재하고, 레퍼런스 시퀀스와 테스트 시퀀스에 동시에 시퀀스 X의 카피가 1개 추가적으로 존재하는 경우.

(d) case 4 : 레퍼런스 시퀀스에 존재하는 서브 시퀀스 X가 테스트 시퀀스에 존재하지 않는 경우.

(e) case 5 : 테스트 시퀀스에 존재하는 서브 시퀀스 X가 레퍼런스 시퀀스에 존재하지 않는 경우.

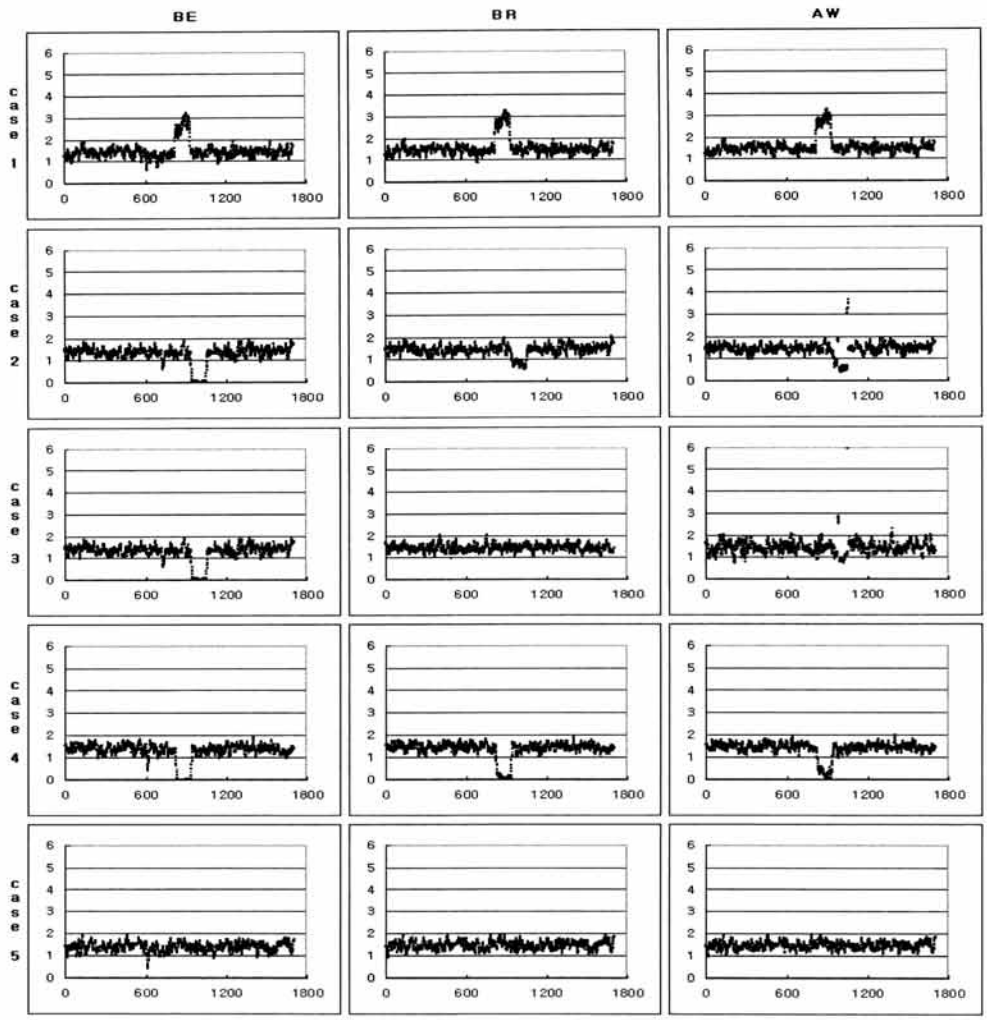
여기에서 case 1과 case 2는 테스트 시퀀스에 반복되거나 결실된 CNV 영역이 존재하는 경우를 나타낸다. 그러나 case 3은 동일한 서브 시퀀스 영역이 두 곳에서 발견되는 경우로 일종의 반복 영역의 예를 나타내며, case 4는 테스트 시퀀스에 삭제 영역이 존재하는 경우를 나타내며, case 5는 이와 반대로 테스트 시퀀스에 삽입 영역이 존재하는 경우를 나타낸다.

다음의 (그림 5)는 각 실험에 의하여 얻어진 리드의 서열 정렬 빈도를 그래프로 나타낸 것이다 제 3.2절에서 보인 Algorithm 1의 $MV_{1000}(F_S)$의 결과를 그래프로 나타낸 것으로서 전체 레퍼런스 시퀀스 영역 중 리드의 서열 정렬 빈도가 다른 영역과 차이를 나타내는 부분만을 확대하여 1/200로 서브 샘플링(sub-sampling)하여 표현한 것이다. 서열 정렬 방식의 비교를 위하여 BE, BR, AW의 3 가지 서열 정렬 방법으로 실험을 수행하였다.

우선, case 1과 case 2의 실험 결과는 CNV 영역이 반복되거나 결실된 경우의 빈도 그래프를 나타낸다. case 1의 빈도 그래프에서는 CNV 영역에 대한 빈도가 주변 영역에 비하여 높아 확실한 차이를 보이고 있으며 이로부터 CNV 영역의 자동 추출이 가능함을 알 수 있다. 서열 정렬 방식을 비교하면, BE, BR, AW이 거의 유사한 결과를 보이고 있으나, BE에 비하여 BR, AW가 안정된 패턴을 보이고 있다. 다음 case 2의 빈도 그래프의 경우에도 CNV 영역에 대한 빈도가 주변 영역에 비하여 낮아 확실한 차이를 보이고 있으며, 이로부터 CNV 영역의 자동 추출이 가능함을 알 수 있다. 서열 정렬 방식을 비교하면, BR, AW는 비교적 안정된 패턴을 보이고 있다. 그러나 BE의 경우에는 반복 출현한 리드를 서열 정렬 결과에서 제외시키므로 CNV 영역에서 빈도가 0을 나타내고 있으며, 따라서 CNV 영역과 단순 반복 영역을 구별할 수 없다.

다음의 case 3, case 4, case 5의 실험 결과는 CNV 영역이 아닌 경우의 빈도 그래프를 나타낸다. 우선, case 3의 결과로부터 BR, AW는 반복 영역의 빈도가 주변 영역과 거의 차이를 보이지 않아 CNV 영역과 구별 가능함을 알 수 있다. 그러나 BE의 경우에는 case 2의 경우에서 설명한 바와 같이 반복 영역과 CNV 영역을 구별할 수 없다. 다음 case 4의 경우, BR, AW는 삭제 영역에서 빈도가 거의 0을 나타내므로 case 2의 경우와 구별되어 삭제 영역임을 판별할 수 알 수 있다. BE의 경우에도 삭제 영역의 빈도가 0을 나타내지만 앞에서 설명한 바와 같이 반복 영역, CNV 영역과의 구별이 불가능하다. 마지막으로 case 5의 경우에는 당연한 결과로서 BE, BR, AW는 모든 영역의 빈도가 거의 차이를 보이지 않으므로 CNV 영역 검출에 장애가 되지 않는다.

이 들 결과로부터 제안된 방식을 사용하여 DNA 시퀀스에 존재하는 반복되거나 결실되는 CNV 영역을 적절히 검출할 수 있음을 알 수 있다. 특히 서열 정렬 방식으로 BR 혹은 AW를 사용하는 경우, 비 CNV 영역으로 볼 수 있는 반복, 삭제, 삽입 영역과 구별하여 CNV 영역을 효율적으로 추출할 수 있음을 알 수 있다. 일반적으로 CNV 영역을 검색한다는 것은 앞에서 보인 case 1의 경우를 의미한다. 따라



(그림 5) CNV, 반복, 삽입, 삭제 영역에 대한 리드의 빈도 그래프 (C=1.6의 경우)

서 이 후의 실험에서는 레퍼런스 시퀀스에 비하여 테스트 시퀀스에 카피 수가 많은 CNV의 경우에 대하여 실험을 수행한다.

4.2.2 실험 2

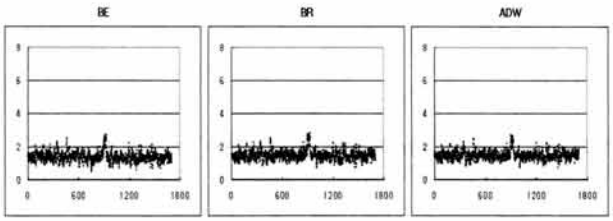
일반적으로 레퍼런스 시퀀스와 테스트 시퀀스를 비교하여 테스트 시퀀스에 CNV로 추정되는 영역의 카피 수가 2 카피 이상으로 증가하면 CNV 영역의 추출이 쉬워진다. 예를 들어 테스트 시퀀스에 CNV 영역이 3 카피 존재하게 되면 리드의 서열 정렬 빈도에서 CNV 영역의 빈도가 확실하게 차이를 보이기 때문이다. CNV 영역 검출에서 가장 발견하기 어려운 경우는 1.5 카피 영역으로 알려져 있다. 여기서 1.5 카피 영역은 부모로부터 한쪽에서 1 카피를 물려받고 다른 한쪽에서 2 카피를 물려받아 자손에게서는 1.5 카피 영역이 나타나는 경우이다. 본 실험에서는 1.5 카피를 갖는 CNV 영역에 대한 실험을 수행하여, 제안하는 방법이 이러한 1.5 카피 영역을 효율적으로 추출함을 보인다.

실험에 사용된 CNV 영역의 크기는 8,302bp로 <표 3>의 3번 CNV 영역이다. <표 3>의 CNV 영역은 약 40%가 6 - 9kbp의 크기를 가진다. 따라서 본 실험에서는 자주 출현 하는 CNV 영역의 크기에 해당하는 3번 영역을 사용하였다.

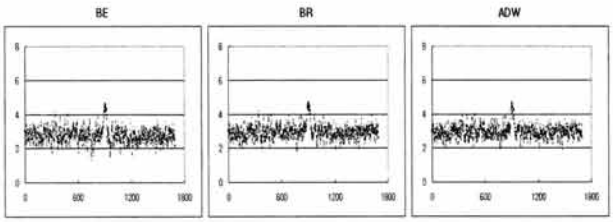
서열 정렬 방식의 비교를 위하여 BE, BR, ADW의 3 가지 서열 정렬 방법으로 실험을 수행하였다. 다음의 (그림 6)에 각 실험에 의한 리드의 서열 정렬 빈도를 보인다. 실험 1의 경우와 같이 전체 레퍼런스 시퀀스 영역 중 리드의 서열 정렬 빈도가 다른 영역과 차이를 나타내는 부분만을 확대하여 샘플링 사이즈 200으로 표현한 것이다. (그림 6)-(a)와 (그림 6)-(b)는 각각 C=1.6 과 C=3.2의 리드를 사용한 실험 결과를 나타낸다.

이 경우에는 CNV 영역의 크기가 비교적 작고, 또한 주변 영역에 대한 빈도 차가 크지 않기 때문에 CNV 영역의 추출이 어렵다. 특히 (그림 6)-(a)에 보인 바와 같이 커버리지의 수가 1.6으로 비교적 리드의 수가 적은 경우에는 주변 잡음의 영향으로 인하여 영역의 변별력이 떨어진다. 그러나 (그림 6)-(b)에 보인 바와 같이 커버리지의 수를 높여 C=3.2 정도로 리드의 수가 많아지면, CNV 영역이 비교적 확실히 나타난다.

다음 <표 4>에 C=3.2의 경우에 대한 BE, BR, ADW의



(a) C=1.6의 경우



(b) C=3.2의 경우

(그림 6) 1.5 카피 영역을 갖는 CNV 영역에 대한 리드의 빈도 그래프

〈표 4〉 CNV 영역 추출을 위한 알고리즘의 성능 비교
(1.5 카피의 CNV 영역, C = 3.2, p-value = 0.01의 경우)

	BE	BR	ADW
Total CNV regions	8302		
Detected CNV regions	7218	7470	5754
False Negatives	1083	831	2547
False Positives	26705	2128	3827
False Negative rate(%)	13.04505	10.00964	30.67935
False Positive rate(%)	321.66948	25.63238	46.09733

〈표 5〉 CNV 영역 추출을 위한 알고리즘의 성능 비교
(다양한 크기의 CNV 영역, C = 3.2, p-value = 0.05의 경우)

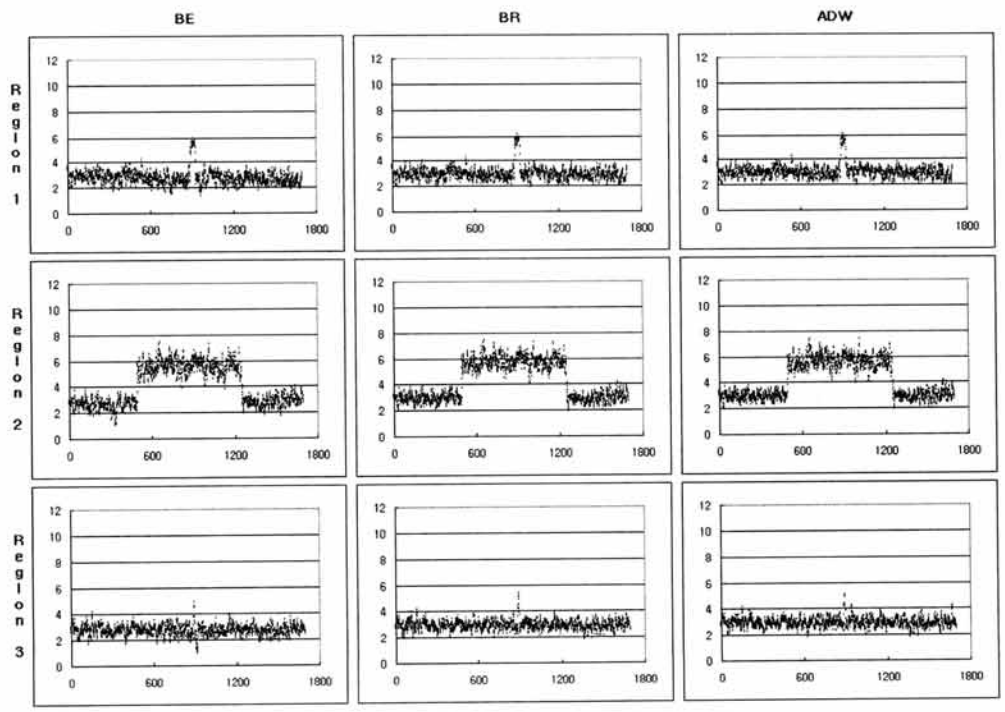
	BE	BR	BW	ADW	ADW
Total regions	163403				
Detected CNV regions	155622	158691	158971	156881	157761
False Negatives	7781	4712	4432	6522	5642
False Positives	7830	0	0	0	0
False Negative rate(%)	4.76185	2.88367	2.71231	3.99136	3.5763
False Positive rate(%)	5.03142	0.0	0.0	0.0	0.0

CNV 영역 추출 성능을 보인다. 이 성능은 제 3.2절에서 보인 Algorithm 1의 함수 Find_Signal()의 결과를 분석한 것으로 후처리 과정을 수행하지 않은 CNV 후보 영역의 추출 결과에 대한 성능을 나타낸다. 이 결과는 CNV 영역에 대한 통계적 유의도로서 p-value = 0.01을 설정한 경우이다. BE의 경우, 실험 1에서 설명한 이유로 인하여 false positive 비율이 높게 나타나고 있다. 이 3 가지 방법 중에서 BR이 가장 우수한 성능을 보이고 있으며, BR의 false negative 비율은 10.0%, false positive 비율은 25.6%를 나타낸다.

이 결과로부터 가장 검색이 어려운 것으로 알려져 있는 1.5 카피의 작은 CNV 영역의 경우에도 제안된 알고리즘은 C=3.2의 작은 커버리지 수의 리드를 이용하여 그 영역을 효율적으로 검색하고 있음을 알 수 있다.

4.2.3 실험 3

실험 3에서는 제안된 방식에 의하여 다양한 크기의 CNV



(그림 7) 다양한 크기의 CNV 영역에 대한 리드의 빈도 그래프(C=3.2의 경우)

영역을 동시에 효율적으로 추출 가능한지를 검증한다. 표 3에 보인 CNV 영역 중에서 가장 작은 사이즈를 갖는 2,425bp의 22번 CNV 영역, 중간 사이즈의 8,302bp의 3번 CNV 영역, 그리고 큰 사이즈의 152,676bp의 9번 CNV 영역을 이용하였다. 즉, 서로 크기가 다른 이 3개의 CNV 영역은 레퍼런스 시퀀스에 1 카피씩 존재하며 테스트 시퀀스에는 2 카피씩 존재한다고 가정한다.

서열 정렬 방식의 비교를 위하여 BE, BR, BW, AW, ADW의 5 가지 서열 정렬 방법으로 실험을 수행하였다. (그림 7)에 BE, BR, ADW에 의한 서열 정렬 빈도를 보인다. 여기에서 Region 1, 2, 3는 각각 전체 레퍼런스 시퀀스 영역 중 리드의 서열 정렬 빈도가 다른 영역과 차이를 나타내는 부분만을 확대하여 샘플링 사이즈 200으로 표현한 것이다. 결과 산출에는 C=3.2의 리드를 사용하였다.

(그림 7)의 그래프로부터 검색 대상의 CNV 영역의 크기가 달라도 해당 영역들이 비교적 명확히 판별 가능함을 알 수 있다. 각 서열 정렬 방식의 자세한 비교를 위하여 표 5에 BE, BR, BW, AW, ADW의 CNV 영역 추출 성능 비교 결과를 보인다. 이 결과는 실험 2의 경우와 같이 후처리 과정을 거치지 않은 CNV 후보 영역의 추출 성능을 나타낸 것이다. 또한 CNV 영역에 대한 통계적 유의도로서 p-value = 0.05를 설정한 경우이다. 실험 결과로부터 5 가지의 서열 정렬 방법이 매우 우수한 성능을 보이고 있음을 알 수 있다. 특히 BR, BW, AW, ADW는 false negative 비율이 2-3%로 매우 낮고, false positive 비율이 0%를 나타내고 있다.

실험 3의 결과로부터 제안된 알고리즘은 C=3.2의 작은 커버리지 수의 리드를 이용하여매우 작거나 중간 사이즈 혹은 큰 사이즈의 CNV 영역을 동시에 매우 효율적으로 검출할 수 있음을 알 수 있다.

5. 결론 및 향후 연구

본 논문에서는 기가 시퀀싱의 결과 산출되는 대량의 짧은 리드를 레퍼런스 시퀀스에 서열 정렬 시킨 후, 리드의 서열 정렬 빈도수 정보를 통계적으로 분석하여 CNV 영역을 찾아내는 새로운 방법을 제안하였다.

본 연구의 공헌은 다음과 같다. (1) 기가 시퀀싱의 결과 산출되는 짧은 길이의 DNA 서열 데이터를 이용한 새로운 CNV 영역 검색 알고리즘을 제안하였다. (2) CNV 영역 검색을 위한 다양한 서열 정렬 방식을 제안하였다. (3) CNV 영역 검색에 장애가 되는 반복 출현 리드의 문제를 해결할 수 있는 방안을 제시하였다. (4) 다양한 실험의 수행을 통하여 제안된 방식에 의한 CNV 검색 성능을 제시함으로써, 3.2 정도의 낮은 커버리지의 기가 시퀀싱 데이터를 이용한 계산적, 통계적 분석 방식에 의하여 CNV 영역 검색이 가능함을 입증하였다. (5) 실험을 통하여 제안된 방식은 작은 사이즈의 CNV 영역에서부터 큰 사이즈의 CNV 영역까지 다양한 크기의 영역을 효율적으로 검출할 수 있음을 입증하였다.

현재 추출된 CNV 후보 영역의 후처리 과정에 관한 연구와 모니터링 툴의 개발을 수행 중에 있다. 이 연구 성과에 따라 본 논문의 실험에 제시된 알고리즘의 성능은 보다 개선될 수 있다. 금후 실제의 유전체 데이터를 대상으로 하는 실험을 수행하여 제안된 방식의 성능 분석 및 개선에 대한 연구를 수행할 예정이다. 또한 실유전체 데이터의 염색체별 특성을 고려하여 제안된 통계적 방법론의 검증 및 보완에 관한 연구를 수행할 예정이다.

참 고 문 헌

[1] F. S. Robert, "The Race for the $1000 Genome," SCIENCE, Vol.311, pp.1544-1546, 2006.

[2] R. Redon, et al, "Global variation in copy number in the human genome," Nature, Vol.444, pp.444-454, 2006.

[3] J. Sebat, B. Lakshmi, J. Troge, J. Alexander, J. Young, P. Lundin, S. Månér, H. Massa, M. Walker, M. Chi, N. Navin, R. Lucito, J. Healy, J. Hicks, K. Ye, A. Reiner, T. C. Gillian, B. Trask, N. Patterson, A. Zetterberg, and M. Wigler, "Large-Scale Copy Number Polymorphism in the Human Genome," Science, Vol.305, pp.525-528, 2004.

[4] A. J. Iafrate, L. Feuk, M. N. Rivera, M. L. Listewnik, P. K. Donahoe, Y. Qi, S. W. Scherer, and C. Lee, "Detection of large-scale variation in the human genome," Nat. Genet., Vol.36, pp.949-951, 2004.

[5] E. Tuzun, A. J. Sharp, J. A. Bailey, R. Kaul, V. A. Morrison, L. M. Pertz, E. Haugen, H. Hayden, D. Albertson, D. Pinkel, M. V. Olson, and E. E. Eichler, "Fine-scale structural variation of the human genome," Nat. Genet., Vol.37, No.7, pp.727-732, 2005.

[6] R. E. Mills, C. T. Luttig, C. E. Larkins, A. Beauchamp, C. Tsui, W. S. Pittard, and S. E. Devine, "An initial map of insertion and deletion (INDEL) variation in the human genome," Genome Res., Vol.16, pp.1182 - 1190, 2006.

[7] R. Khaja, J. Zhang, J. R. MacDonal, H. Yongshu, M. J. Joseph-George, J. Wei, M. A. Rafiq, C. Qian, Shago M., L. Pantano, H. Aburatani, K. Jones, R. Redon, M. Hurles, L. Armengol, X. Estivill, R. J. Mural, C. Lee, S. W. Scherer, and L. Feuk, "Genome assembly comparison identifies structural variants in the human genome," Nat. Genet., Vol.38, No.12, pp.1413-1418, 2006.

[8] S. W. Schrer, C. Lee, E. Birney, D. M. Altshuler, E. E. Eichler, N. P. Carter, M. E. Hurles, and L. Feuk, "Challenges and standards in integrating surveys of structural variation," Nat. Genet., Vol.39, No.7, S7-S15, 2007.

[9] 홍상균, 홍동완, 윤지희, 김종일. "Short read 서열정렬에 의한 CNV 영역 추출," In proceedings of KDBC 2008, pp.297-305, 2008.

[10] http://www.cbcb.umd.edu/software/RepeatFinder

[11] R. L. Warren, G. G. Sutton, S. J. Jones, and R. A. Holt, "Assembling millions of short DNA sequences using SSAKE," Bioinformatics Vol.23, No.4, pp.500-501, 2007.

[12] W. R. Jeck, J. A. Reinhardt, D. A. Baltrus, M. T. Hickenbotham, V. Magrini, E. R. Mardis, J. L. Dangl, and C. D. Jones, "Extending assembly of short DNA sequences to handle error," Bioinformatics Vol.23, No.21, pp.2942-2944, 2007.
[13] J. C. Dohm, C. Lottaz, T. Borodina, and H. Himmelbauer, "SHARCGS, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing," Genome Res. Vol.17, No.11, pp.1697-1706, 2007.
[14] http://www.illumina.com
[15] R. Li, et al, "SOAP: short oligonucleotide alignment program.," Bioinformatics Vol.24, No.5, pp.713-714, 2008.
[16] http://maq.sourceforge.net
[17] http://rulai.cshl.edu/rmap
[18] http://brainarray.mbni.med.umich.edu/Brainarray/SequenceAlignment/AQUESA
[19] P. Weiner, "Linear Pattern Matching Algorithms," Proc. 14th IEEE Annual Symp. on Switching and Automata Theory, pp.1-11, 1973.
[20] U. Manber and G.e Myers, "Suffix arrays: a new method for on-line string searches," SIAM Journal on Computing, Vol.22, Issue 5, pp.935-948, 1993.
[21] S. W. Schrer, C. Lee, E. Birney, D. M. Altshuler, E. E. Eichler, N. P. Carter, M. E. Hurles, and L. Feuk, "Challenges and standards in integrating surveys of structural variation," Nat. Genet., Vol.39, No.7, S7-S15, 2007.
[22] S. Altschul, T. Madden, A. Schaffer, J. Zhang, W. Miller, and D. Lipman, "Gapped BLAST and PSI-BLAST: A New Generation of Protein Data-base Search Programs," Nucleic Acids Research, Vol.25 No.17 pp.3389-3402, 1997.
[23] W. J. Kent, "BLAT - The Blast - Like Alignment Tool," Genome Research, Vol.12, No.4, pp.656-664, 2002.
[24] W. W. Daniel, "Biostatistics (8th ed.)," Wiley, 2005.
[25] D. L. Wheeler, C. Chappey, A. E. Lash, D. D. Leipe, T. L. Madden, G. D. Schuler, T. A. Tatusova and B. A. Rapp, "Database resources of the National Center for Biotechnology Information," Nucleic Acids Research, Vol.28 No.1 pp.10-14, 2000.
[26] http://projects.tcag.ca/variation
[27] S. Tada, R. Hankins, and J. Patel, "Practical Suffix Tree Construction," In Proceedings of the 30th VLDB Conference, pp.36-47, 2004.

홍 상 균

e-mail : kyoons@hallym.ac.kr
2005년 한림대학교 정보통신공학부(학사)
2007년 한림대학교 컴퓨터공학과(공학석사)
2008년~현 재 한림대학교 컴퓨터공학과 박사과정
관심분야 : 데이터베이스 시스템, XML, 저장 시스템, 서열 정렬

홍 동 완

e-mail : dwhong@hallym.ac.kr
1996년 한림대학교 전자계산학과(학사)
1098년 한림대학교 컴퓨터공학과(공학석사)
2008년 한림대학교 컴퓨터공학과(공학박사)
2003년~2005년 송곡대학 미디어컨텐츠학과 전임강사
2005년~2007년 송곡대학 미디어컨텐츠학과 조교수
2008년~현 재 한림대학교 바이오메디컬학과 겸임조교수
관심분야 : 기가 시퀀싱, 유전체 변이, 서열 정렬, 마이크로어레이, 유전자 기능 분석

윤 지 희

e-mail : jhyoon@hallym.ac.kr
1982년 한양대학교 전자공학과업(학사)
1985년 일본 구주대학교 정보공학과(공학석사)
1988년 일본 구주대학교 정보공학과(공학박사)
1998년~1999년 미국 UCLA대학교 전산학과 방문교수
1988년~현 재 한림대학교 컴퓨터공학과 교수
관심분야 : 시계열 데이터베이스, 데이터 마이닝, XML, 공간 데이터베이스, GIS

김 백 섭

e-mail : bskim@hallym.ac.kr
1978년 한양대학교 전자공학과(학사)
1980년 한국과학기술원 전기및 전자공학과(석사)
1985년 한국과학기술원 전기및 전자공학과(박사)
1997년~1998년 미국 Suny Buffalo 대학 방문교수
1987년~현 재 한림대학교 컴퓨터공학과 교수
관심분야 : 패턴인식, 영상이해, 데이터 마이닝

박 상 현

e-mail : sanghyun@cs.yonsei.ac.kr
1989년 서울대학교 컴퓨터공학과(학사)
1991년 서울대학교 컴퓨터공학과(석사)
2001년 UCLA대학교 전산학과(박사)
1991년~1996년 대우통신 연구원
2001년~2002년 IBM T. J. Watson Research Center Post-Doctoral Fellow
2002년~2003년 포항공과대학교 컴퓨터공학과 조교수
2003년~2006년 연세대학교 컴퓨터과학과 조교수
2006년~현 재 연세대학교 컴퓨터과학과 부교수
관심분야 : 데이터베이스, 데이터 마이닝, 바이오인포매틱스, 적응적 저장장치 시스템