CNV 영역 검색 알고리즘

홍상균, 홍동완, 윤지희
한림대학교 컴퓨터공학과
e-mail:kyoons@hallym.ac.kr

A CNV Detection Algorithm

Sang-Kyoon Hong, Dong-Wan Hong, Jee-Hee Yoon
Dept of Computer Engineering, Hallym University

요 약

최근 생물정보학 분야에서 인간 유전체에 존재하는 CNV(copy number variation)에 관한 연구가 주목받고 있다. CNV 영역은 1kbp~3Mbp 사이의 서열이 반복되거나 결실되는 변이 영역으로 정의된다. 우리는 선행 연구에서 기가 시퀀싱(giga sequencing)의 결과 산출되는 DNA 서열 조각인 리드(read)를 레퍼런스 시퀀스에 서열 정렬하여 CNV 영역을 찾아내는 새로운 CNV 검색 방식을 제안하였다. 후속 연구로서 본 논문에서는 DNA 서열에 존재하는 repeat 영역 문제를 해결하기 위한 새로운 방안을 제안하고, 리드의 출현 빈도 정보를 분석하여 CNV 영역을 찾아내는 CNV 영역 검색 알고리즘을 보인다. 제안된 알고리즘은 Gaussian 분포를 갖는 출현 빈도 정보로부터 통계적 유의성을 갖는 영역을 추출하여 CNV 영역 후보로 하고, 다음 정제 과정을 거쳐 최종의 CNV 영역을 추출한다. 성능 평가를 위하여 프로토타입 시스템을 개발하였으며, 시뮬레이션 실험을 수행하였다. 실험 결과에 의하여 제안된 방식은 반복되거나 결실되는 형태의 CNV 영역을 효율적으로 검출하며, 또한 다양한 크기의 CNV 영역을 효율적으로 검출할 수 있음을 입증한다.

1. 서론

2002년에 초안이 발표된 휴먼 게놈 프로젝트(human genome project: http://www.genome.gov/12011238)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 비용이 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 유전체(genome)는 약 30억bp의 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]. 최근, 이러한 서열 정보 간의 차이를 밝혀내기 위한 연구 중 하나로서 copy number variation(CNV)에 관한 연구가 주목 받고 있다. CNV는 임의의 서브 시퀀스가 양쪽 서열에서 발견되는데 한쪽 서열에서 추가적인 copy를 발견할 수 있는 경우를 나타낸다[2]. CNV 영역 검출을 위한 기존 방식은 마이크로어레이 기술을 이용한 방법[3,4]과 서열 비교 방법[5,6,7]으로 나눌 수 있으며, 참고 논문 [8]에 이 들 방식에 관한 설명과 비교가 잘 정리되어 있다.

우리는 선행 연구 [9]에서 새로운 CNV 검색 방식을 제안한 바 있다. 우리가 제안한 CNV 검색 방식은 기가 시퀀싱의 결과 산출되는 수 많은 짧은 길이의 리드를 레퍼런스 시퀀스에 서열 정렬 시키고, 리드의 출현 빈도를 이용하여 CNV 영역을 검색하는 방식으로서 이 전에 시도된 바 없는 전혀 새로운 방식이다. 본 연구의 목적은 이 선행 연구 결과를 확장하여 보다 정확히 CNV 영역을 검출하는 알고리즘을 제시하고, 이를 더욱 효율적으로 처리하는 방법을 제시하는 것이다.

우리가 제안한 방식은 CNV 영역에 대한 리드의 출현 빈도의 변화를 예측하여 CNV 영역을 검출하고자 하는 방식이다. 그러나 유전체 서열에는 수많은 짧은 길이의 repeat 영역이 존재하며, 이 repeat 영역에 의하여 발생하는 반복 출현 리드들은 CNV 영역 추출을 방해하는 주요 요소로 작용한다. 특히, 리드의 길이가 짧은 경우, CNV 영역을 제외한 다른 repeat 영역으로부터 추출된 동일 리드의 발생 확률이 높아지며, 이 들은 CNV 영역을 탐색하는데 가장 큰 장애로 작용한다. 이를 위한 간단한 방안으로서 리드를 레퍼런스 시퀀스에 서열 정렬시켜 두 군데 이상의 정렬 위치가 존재하는 경우, 해당 리드를 버리는 방법을 사용할 수 있다[9]. 그러나 이 경우, CNV 영역으로부터 추출된 많은 리드가 삭제될 수 있어 정보 손실이 크다.

본 연구에서는 그 해결 방안으로서 리드에 가중치 정보를 부여하여 빈도 정보를 관리하고, 통계적 오류 정보를 소거하는 새로운 방안을 제안한다. 제안하는 방식에서는 리드를 레퍼런스 시퀀스에 서열 정렬시켜 n개의 정렬 위치가 존재하는 경우, 각 위치의 출현 빈도를 1/n만큼 증가시킨다. 또한 측정된 빈도 정보에서 통계적 오류 정보를 소거하여 빈도 정보로 활용한다.

이와 같이 추출된 레퍼런스 상의 출현 빈도를 이용한 CNV 영역 검색 알고리즘은 후보 영역 추출 단계와 정제 단계로 이루어진다. 후보 영역 추출 단계에서는 Gaussian 분포를 갖는 출현 빈도 정보로부터 통계적 유의성을 갖는 연속 영역을 검색하여 이들을 CNV 후보 영역으로 반환한다. 다음 정제 단계에서는 후처리 작업을 수행하여 정확한 CNV 영역을 추출하며, 반복 혹은 결실된 형태의 종류, 영

역 크기 등 CNV의 특성 분석 결과를 반환한다.

제안된 방식의 유효성을 보이기 위하여 NCBI(National Center for Biotechnology Information의 레퍼런스 서열(build 35)을 사용한 시뮬레이션 실험을 수행하였으며, 실험 및 분석 결과에 의하여 다양한 형태의 CNV 영역이 효율적으로 검출됨을 입증한다.

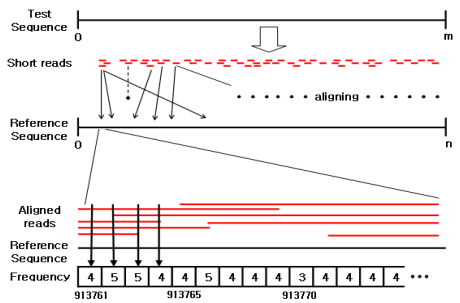
2. CNV 영역 검색 알고리즘

2.1 문제 정의

서로 다른 두 서열을 비교하여 검색하는 유전적 구조 변이 중의 하나인 CNV는 다음과 같이 정의된다. 임의의 서브 시퀀스가 양쪽 서열에서 발견되는데 한쪽 서열에서 추가적인 copy를 발견할 수 있는 경우로서 그 영역의 크기가 1kbp(kilo base pair) 이상의 경우 이를 CNV라고 부르며, 그 영역을 특히 CNV 영역이라고 부른다.

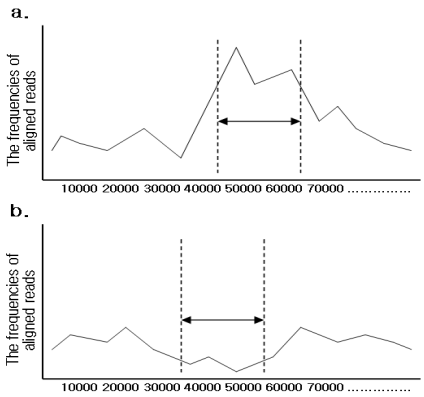
본 연구에서는 비교 대상이 되는 두 시퀀스로서 이미 시퀀싱이 완성된 표준의 레퍼런스 시퀀스와 임의의 테스트 시퀀스를 가정하며, 테스트 시퀀스 상에 존재하는 CNV 영역을 검색하는 방법을 개발한다. 단, 여기에서 테스트 시퀀스는 수 많은 짧은 DNA 시퀀스인 리드로 이루어져 있는 경우에 해당한다.

선행 연구[9]에서 제안된 CNV 검색 방식을 간단히 요약하면 다음과 같다. 제안하는 방법에서는 테스트 시퀀스로부터 생성된 수 많은 리드를 레퍼런스 시퀀스에 서열 정렬시킨 후, 그 정렬된 위치와 출현 빈도 정보를 이용하여 CNV 영역을 추정한다. 그 출현 빈도 정보를 분석하여, 만약 레퍼런스 시퀀스의 임의의 영역에서 리드의 출현 빈도가 주위에 비하여 상대적으로 높게 나타났거나 낮게 나타났다면 해당 영역의 서브 시퀀스가 테스트 시퀀스에 여러 번 반복되어 나타난 부분일 가능성이 높다. 다시 말해, 레퍼런스 시퀀스 상에 이렇게 빈도수의 차이를 보이는 영역은 CNV를 포함하는 유전적 구조 변이를 나타내는 영역을 나타낼 가능성이 높다고 판단하여 이들을 CNV 영역으로 추정하는 방식이다.



(그림 1) 리드의 서열 정렬 과정

제안된 방식을 도식적으로 표현하면 그림 1과 같이 나타낼 수 있다. 첫 번째 과정은 테스트 시퀀스로부터 수많은 리드가 생성되는 과정을 나타내며, 다음 두 번째 과정은 이들 리드를 레퍼런스 시퀀스에 정렬시키는 과정을 나타낸다. 다음 세 번째 과정은 레퍼런스 시퀀스에 정렬된 리드의 위치를 확대하여 보여주며, 각 위치에 정렬된 리드의 출현 회수를 표현하고 있다. 다음의 그림 2는 이와 같이 생성된 빈도 정보를 이용하여 레퍼런스 시퀀스의 각 위치에 대한 리드 출현 회수를 그래프로 나타낸 예를 나타낸다. 그림 2.a는 특정 영역의 빈도가 상대적으로 높은 경우를 나타내며, 이는 레퍼런스 시퀀스에 비하여 테스트 시퀀스에 임의의 영역이 여러번 출현하여 많은 리드가 생성되었을 가능성을 나타낸다. 다음 그림 2.b는 이와 반대의 경우를 가정한 빈도 그래프를 나타낸다. 즉 우리는 이와 같은 특정 영역을 자동 검색하여 CNV 후보 영역으로 추정한다.



(그림 2) CNV 추정 영역의 예

2.2 CNV 영역 검색 알고리즘

CNV 영역을 검색하기 위한 알고리즘 **FIND_CNV**를 Algorithm 1에 보인다. Algorithm 1은 리드 집합 Q와 레퍼런스 시퀀스 R을 입력으로 받아 CNV 영역을 찾아낸다. Algorithm 1의 동작 과정을 단계별로 간단히 설명하면 다음과 같다.

Algorithm 1: FIND_CNV : 영역 검색 알고리즘

```
Input : set of reads Q,  reference sequence R,
        window size |W|, skip_width S
Output : set of CNV regions CNV
1.  Initialize frequency array FreqArray ;
2.  Index T := BuildIndex(R) ;
3.  QC := Sort&Count(Q) ;
4.  for each short read QC[i].SR of the QC do
5.      aligned position set P:=SearchIndex(T, QC[i].SR);
6.      for each aligned position P[j] of the P do
7.          for k := 0 to ReadLen do
8.              FreqArray[k+P[j]] :=
                    FreqArray[k+P[j]] + (QC[i].Num/P[j].Count);
9.  W_FreqArray := Convert(FreqArray, |W|, S);
10. Signal := Find_Signal(W_FreqArray);
11. CNV := FindCNV_region(Signal);
12. return CNV;
```

(1) 리드의 서열 정렬 및 출현 빈도 정보 추출 과정

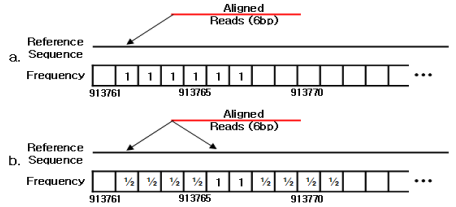
우선, 레퍼런스 시퀀스 상의 각 위치에 대한 리드의 출현 회수를 기록하기 위한 배열 FreqArray를 설정하고, 이를 초기화한다(line 1). 이때 FreqArray 배열의 크기는 레퍼런스 시퀀스의 길이와 같다. 다음, 리드의 고속 검색을 위하여 함수 BuildIndex()를 호출하여 레퍼런스 시퀀스 R에 대한 인덱스 T를 구성하고(line 2), 함수 Sort&Count()를 호출하여 중복이 제거된 리드의 집합 QC를 구한다(line 3). 입력으로 주어진 리드의 집합 Q에는 테스트 시퀀스에서 추출한 수백만~수천만개의 리드가 들어 있으며, 그 중에는 중복된 리드가 존재할 수 있다. 따라서 중복되어 나타나는 리드에 대한 인덱스 검색을 최적화하기 위하여 함수 Sort&Count()에서는 리드에 대하여 사전에 정렬(sorting)을 수행하여 중복 출현하는 리드를 소거하고, 중복 리드들의 출현 회수를 카운팅하여 저장한다. 여기에서 QC[i].SR

은 중복이 제거된 리드를 나타내며, QC[i].Num은 QC[i].SR의 중복 출현 회수를 나타낸다.

다음 line 4-8은 각 리드에 대하여 인덱스 검색을 수행하여 레퍼런스 시퀀스와의 정렬 위치 P[j]를 구하고, 이를 기반으로 배열 FreqArray의 값을 얻는 과정을 나타낸다. 우선 함수 SearchIndex()를 호출하여 각 리드에 대한 인덱스 탐색을 수행하며, 그 결과로서 리드의 레퍼런스 시퀀스 상의 정렬 위치 P[j]를 얻는다(line 5). 여기에서는 각 리드에 대하여 exact match, k=1 mismatch, k=2 mismatch 연산을 수행하여 해당하는 정렬 위치를 모두 검색하는 것을 가정한다.

레퍼런스 시퀀스와의 서열 정렬 결과, 각 리드는 레퍼런스 시퀀스에 전혀 출현하지 않거나, 유일하게 1번 출현 하거나 2번 이상 출현할 수 있다. 이와 같이 얻어진 리드의 정렬 위치 개수는 P[i].Count에 저장된다. 다음, 리드의 정렬 위치 정보 P를 이용하여 각 리드의 정렬 위치에 대한 빈도수를 산출한다(line 7-8). 이 단계에서는 해당 리드가 정렬되어 나타난 구간(P[i]~P[i]+readLen)의 FreqArray 배열 값으로 현재의 값에 QC[i].Num/P[i].Count의 값을 더한 값을 할당한다. 여기서 QC[i].Num는 해당 리드의 중복 출현 회수를 나타내며, ReadLen은 리드의 길이를 나타낸다.

다음 그림 3에 배열 FreqArray의 값을 설정하는 과정을 예를 들어 보인다. 이 때 배열 FreqArray는 초기화 되어 있으며 해당 리드는 6bp의 크기를 가지고 QC[i].Num = 1의 경우를 가정한다. 그림 3.a는 해당 리드가 레퍼런스 시퀀스의 유일한 영역에 정렬된 경우를 나타내며, 이 경우 해당 영역에는 빈도수 1이 할당된다. 다음 그림 3.b는 하나의 리드가 2곳에 정렬된 경우(P[i].Count=2)를 나타내며, 이 경우에는 해당 영역에 각각 1/2의 빈도 값이 더해지게 된다.



(그림 3) 가중치 기반 정렬 빈도 계산

(2) CNV 영역 추출 과정

다음 line 9 - line 12는 빈도 정보 FreqArray를 이용한 CNV 영역 추출 과정을 나타낸다. 그러나 빈도 정보 FreqArray에는 0의 값이 자주 출현하며, 또한 repeat 영역에 의하여 생성된 특이 값이 많이 포함되므로 이로부터 직접 CNV 영역을 추출하기 어렵다. Line 9의 함수 convert()는 윈도우 사이즈 |W|, 쉬프트 사이즈 S를 입력으로 받아 각 빈도 정보를 윈도우 사이즈 단위로 평균을 내서 S 간격으로 저장한 새로운 배열 W_FreqArray를 생성한다. 다음, line 10의 함수 Find_Signal()은 Gaussian 분포를 갖는 W_FreqArray의 출현 빈도 정보로부터 통계적 유의성을 갖는 연속 영역을 검색하여 이들을 CNV 후보 영역으로 반환한다. 다음, 함수 Find_CNV_region()은 후처리 작업을 수행하는 함수를 나타내며, 정확한 CNV 영역을 추출하고, 해당 CNV의 특성 분석 결과를 반환한다.

3. 실험

3.1 실험 방법

레퍼런스 시퀀스로서 NT_028392.5의 contig를 사용하였으며, 2,525,983-2,548,917의 영역에 약 20kbp 크기의 CNV 영역이 존재한다. 추가 copy 영역은 CNV 영역으로 보고된 20kbp의 서열 정보를 추가로 삽입하여 생성하였다. 실험은 CNV 후보의 경우라 판단할 수 있는 2가지의 경우와 CNV 후보가 아닌 3가지 경우로 총 5가지의 경우로 분류하여 실험하였다.

테스트 시퀀스로부터 랜덤하게 선택된 위치에서 Solexa machine의 성능과 유사하게 36bp(base pair)의 리드를 추출한다. 이 때 생성하는 리드의 개수는 서열 데이터의 80%를 커버하는 수준으로 2번 생성한 것으로 1.6 coverage를 가지게 된다. 레퍼런스 시퀀스에 대해서는 접미어 트리 인덱스를 생성한다. 사용되는 접미어 트리는 참고 문헌[10]의 Top-down disk-based 전략을 사용하여 전체 유전체 서열 정보와 같은 대용량의 시퀀스를 효과적으로 인덱싱할 수 있는 방법으로 알려져 있다. 본 실험에서는 참고 문헌[10]의 웹사이트(http://www.eecs.umich.edu/tdd/index.html)로부터 접미어 프로그램을 다운로드하여 인덱스를 작성하였다.

실험 결과 그래프는 윈도우 사이즈는 2kbp, 쉬프트 사이즈는 200bp로 하여 작성하였다. 즉 그래프의 한점은 실제 빈도 정보에서 2kbp의 데이터의 평균이 된다.

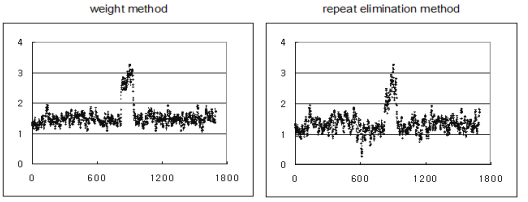
실험을 위한 플랫폼으로는 Linux (Kernel Version 2.6.26)를 운영체제로 사용하고, 8GB의 주기억 장치, 640GB 디스크를 갖는 Core2Quad 2.83GHz의 PC를 사용한다.

4.2 실험 결과

Repeat 영역 보정 방법으로 2 가지 방법을 사용하여, 이 결과를 비교하였다. 첫 번째 방법은 리드의 정렬 위치가 유일하지 않을 경우 해당 리드의 정렬 정보를 버리는 방법으로 일반적으로 리드의 정렬 방법에 널리 사용되는 방법이다. 본 실험에서는 이 방법을 repeat elimination method라고 명명한다. 또한 본 연구에서 제안하는 가중치를 이용한 방법을 weight method라고 명명한다.

(1) 실험 1

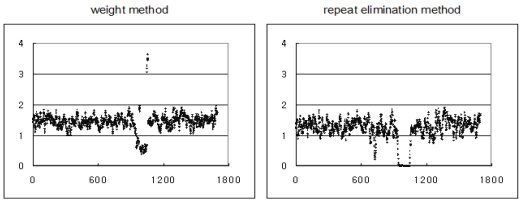
CNV 후보 영역으로 레퍼런스 시퀀스에 1 copy가 있고 테스트 시퀀스에 2 copy가 존재하는 경우로 테스트 시퀀스에 더 많은 copy 영역이 존재하는 경우이다. 그림 4는 정렬빈도 중 빈도수가 다른 영역과 차이를 나타내는 영역을 확대하여 그려놓은 것이다. 두 방법 모두 그림과 같이 특정 영역의 빈도가 높게 나타나는데 이 영역은 보고된 CNV영역과 일치한다.



(그림 4) 테스트 시퀀스에 더 많은 copy가 존재하는 경우

(2) 실험 2

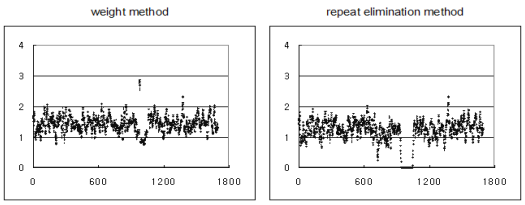
실험 1과는 반대로 레퍼런스 시퀀스에 2 copy가 존재하고 테스트 시퀀스에 1 copy가 존재하는 경우이다. 그림 5는 빈도 영역이 특별한 차이를 나타내는 영역을 나타낸 것이다. 두 방법 모두 그림과 같이 특정 영역의 빈도가 낮게 나타남을 볼 수 있다. 이 역시 CNV 영역으로 보고된 위치와 일치한다. 단, 제안하는 weight method의 경우 빈도가 낮은 경우에도 일부 영역이 높게 나타나는데 이는 repeat 영역으로 볼 수 있는 영역으로 일반적인 노이즈 보정법으로 보정이 가능하다.



(그림 5) 레퍼런스 시퀀스에 더 많은 copy가 존재하는 경우

(3) 실험 3

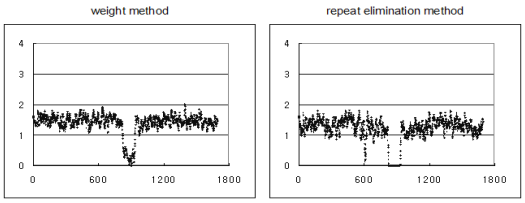
동일한 시퀀스 영역이 두 곳에서 발견되는 레퍼런스 시퀀스와 테스트 시퀀스를 사용한 실험으로, copy 수가 다르지 않기 때문에 CNV라 볼 수 없는 경우이다. Repeat elimination method의 경우에는 정렬 위치가 여러 곳인 경우에 해당 리드의 정보를 버리기 때문에 해당 영역의 빈도수가 0에 가깝게 나타나고 있다. 이는 CNV 후보영역의 두 번째의 경우와 유사한 결과로 해당 영역을 CNV 후보 영역으로 판단할 수 있는 오류가 있다. 반면 weight method의 경우 상대적으로 빈도수가 일정하게 나타남을 볼 수 있다.



(그림 6) 두 시퀀스에 길이가 긴 repeat 영역이 존재하는 경우

(4) 실험 4

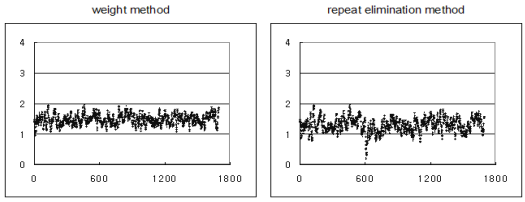
레퍼런스 시퀀스에는 존재하는 영역이 테스트 시퀀스에는 존재하지 않는 경우로 deletion으로 정의되는 경우이다. 그림 7을 보면 두 경우 모두 빈도수가 낮은 영역이 나타나고 있다. 이는 CNV 후보 영역으로 판단될 오류가 있는 부분이다.



(그림 7) 테스트 시퀀스에 deletion이 존재하는 경우

(5) 실험 5

레퍼런스 시퀀스에는 존재하지 않는 영역이 테스트 시퀀스에 존재하는 경우로 일종의 insertion 경우를 나타낸다. 그림 8을 보면 두 방법 모두 빈도수가 일정하게 나타남으로 insertion을 CNV 영역으로 판단하지 않는다. 하지만 repeat elimination method의 경우 아주 작은 부분들의 빈도수가 낮게 나타남을 알 수 있다. 이는 repeat 영역에 의해 빈도수가 낮게 나타난 경우일 수 있지만, 그 영역이 1kbp이상이기 때문에 CNV 후보 영역으로 볼 수 도 있다는 문제점이 있다.



(그림 8) 테스트 시퀀스에 insertion이 존재하는 경우

4. 결론 및 향후 연구

본 논문에서는 레퍼런스 시퀀스와 리드를 서열 정렬을 통해 계산된 빈도수 정보를 통해 CNV 영역을 찾아내는 새로운 방법을 제안하였다. 본 연구에서는 CNV 후보 영역과 비 후보 영역의 다양한 경우를 실험을 통해 효과적으로 검색함을 보였다. 금후 실제 규모의 유전체 데이터를 대상으로 하는 실험을 통해 제안한 방법의 성능 및 검색 능력 개선 방안에 대한 연구를 수행할 예정이다.

참고문헌

[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] S. Tada, R. Hankins, and J. Patel, "Practical Suffix Tree Construction," In Proceedings of the 30th VLDB Conference, pp. 36-47, 2004.