Haplotype 블록 분할을 위한 LD 기반 알고리즘

나경락[0], 김상준, 여상수, 김성권
중앙대학교 컴퓨터공학부 알고리즘 및 정보보호 연구실
{bike[0], jjuns, ssyeo}@alg.cse.cau.ac.kr, skkim@cau.ac.kr

LD-based Algorithm for Haplotype Block Partitioning

Kyung-Rak Na[0]. Sang-Jun Kim. Sang-Soo Yeo. Sung-Kwon Kim
School of Computer Science & Engineering, Chung-Ang University, Seoul, Korea

요 약

본 연구는 Haplotype 데이터에서 나타나는 별개의 Haplotype의 수를 최소화하는 블록으로 분할하는 방법을 제안한다. Multi-population case인 Haplotype 데이터를 분석하기 위해 패턴의 개수를 최소한으로 줄이는 블록 분할 방법은 전산학적인 최적해의 의미를 가지게 되며, 이와 더불어 생물학적인 의미를 가지는 블록 경계를 찾기 위해 $|D'|$을 계산하고 LD를 분석하였다. 분석된 LD는 블록 분할 알고리즘에서 블록 결정 함수로 사용하였으며, 이에 대한 검정은 χ^2-test를 통해 이루어졌다. 많은 Sample로 구성된 Haplotype 데이터로부터 평균 패턴의 개수를 최소화하고 긴 블록 길이를 가지는 블록 분할의 결과를 얻었다.

1. 서 론

인간의 Haplotype 지도를 만들기 위한 기반 기술의 연구는 유전체학 이후의 SNP 연구에서 가장 활발한 연구 영역중의 하나이다. Haplotype 데이터로부터 블록 구조를 식별하는 것은 Haplotype 지도 연구에 있어서 핵심 과정이다. 여러 인종에서 관찰된 블록 안 별개의(distinct) Haplotype의 수는 제한적인 것으로 알려졌고, 이 발견은 Haplotype과 표현형 사이의 훨씬 더 강력한 연합을 얻을 수 있을 것이라 기대되어 질병 연관 연구에 매우 중요하다. 따라서 블록과 흔한 Haplotype이 식별된 이후, 이러한 중요성으로 인해 블록을 분할하기 위한 여러 알고리즘이 제안되고 있다. 본 논문에서는 Haplotype 블록을 최소수인 별개의 Haplotype으로 분할하는 전산학적 방법과 LD에 기반한 블록 분할로써 SNP의 매 쌍 간의 LD 신뢰도 값을 χ^2을 통하여 검정한 블록 분할 알고리즘을 제안한다.

2. 알고리즘

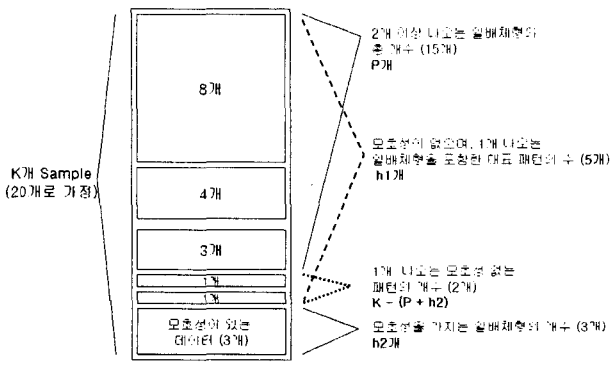
2.1 Haplotype 블록 분할 (Block Partitioning) 관련 연구

질병에 관련된 유전자를 찾기 위해 가계도를 살피는 전통적인 연관 (linkage) 연구에 비해, LD에 기반한 군집 (Association) 연구는 혈연이 아닌 개개인의 유전자형 (Genotype) 데이터를 이용할 수 있으므로 많은 수의 개인을 표본으로 사용할 수 있는 장점을 가진다. 또한 LD가 역사적인 재조합 사건 (Historical recombination event)의 많은 수를 반영하여 가계 연구보다 넓은 범위의 유전자 사상 (Mapping)이 가능해 진다. Mapping을 위해 개개인의 매 SNP마다 유전자형 분석을 하는 것은 그 수가 너무 많아지며 현재의 기술로는 비용 또한 많이 들게 된다. 유전체 전체에 걸친 군집 (Association)연구를 위해 필요한 SNP의 수는 LD 형 (Pattern)에 따라 결정되는 것으로 알려졌다. [2]. 연구를 통해 Haplotype이 관찰된 인종에서 각 블록 안의 별개의 Haplotype의 수는 매우 제한적이다. [5]

보통 블록내의 80-90%가 3-5개의 흔한 Haplotype에 속한다. 이러한 발견은 질병 군집 연구에서 매우 중요하다. 모든 개인의 SNP을 분석하는 대신, 블록 분할을 하고 각 블록으로부터 보다 적은 수의 대표 SNP을 선택하면 대부분의 Haplotype 블록을 대표 (Tag) SNP으로 대부분 획득할 수 있기 때문이다.[1, 3, 4] 이를 통해 군집 (Association) 연구에서 필요한 SNP의 수를 많은 능력 (Power)의 손실 없이 상당히 줄일 수 있을 것으로 기대하고 있다. 이러한 방법은 연구자들이 많은 수의 SNP으로 인해 실험에 들이는 비용을 실질적으로 줄일 수 있게 된다.

2.2 Haplotype 블록 분할 시스템

본 논문에서는 별개의 Haplotype의 수를 최소화 하는 최적 블록의 분할을 찾는 알고리즘을 제안한다. [그림 1]과 같이 K개의 Sample이 입력으로 들어오면 알고리즘은 목적함수 값인 2개 이상의 흔한 Haplotype들의 패턴 수 $f(\cdot)$를 계산한다.



[그림 1] 분할된 한 블록 내의 데이터 분석 모식도 예

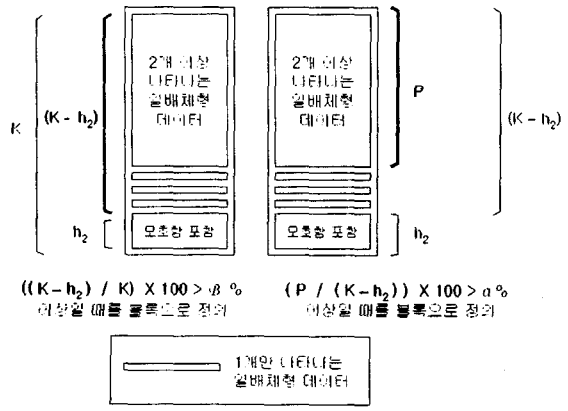
$f(\cdot)$는 모호성이 없는 Sample의 모든 패턴 수(h_1)에서 모호성을 가지지 않으며 1번 나오는(Rare) Haplotype의 패턴 개수

본 연구는 한국 과학재단 목적기초연구 (R01-2003-000-11573-0)지원으로 수행되었음.

를 제외하여 얻은 2개 이상 나타나는 흔한 (Common) Haplotype의 대표 패턴의 수가 된다. 목적함수 $f(\cdot)$를 최소화하는 최적 블록을 찾아내기 위해 Dynamic Programming Algorithm을 이용하였다. 패턴 수 분석에 포함되는 Haplotype은 2번 이상 나타난 흔한 (Common) Haplotype으로 한정하므로, 블록 내 1번만 나타나는 (Rare) Haplotype과 모호함을 가지는 (Ambigous) Haplotype은 Noise의 의미를 가지게 된다.

분할된 블록에서 목적함수인 패턴의 수가 작게 나와도, 전체에서 차지하는 Noise의 비율이 많은 결과는 의미가 없다. 별개인 패턴의 수를 최소한으로 줄이는 블록 분할을 찾고자 할 경우, 블록이 길어질수록 흔한 Haplotype의 패턴수는 작지만 Noise의 비율이 많아진다. 반대로 블록이 길이가 너무 짧으면 블록 분할의 의미가 없어진다. 따라서 목적 함수와는 별도로 블록으로 인정하는 블록 결정 함수를 설정하고, 이 조건에 맞는 경우만을 블록으로 인정하여 이러한 문제를 해결하였다.

블록 결정 함수는 [그림 2]의 수식에 의해 결정된다.



[그림 2] 블록 결정 함수의 설정

α%는 모호함을 제외한 부분(K - h_2)을 전체로 보았을 때, 2번 이상 나타나는 Common Haplotype으로 구성된 P의 Coverage를 설정한다. β%는 모호함을 제거한 데이터의 Coverage를 제한한다. 구현한 시스템은 동적 프로그래밍 알고리즘을 사용하여 각 블록 구간 $i \ldots j$ 에 대해 블록 분할을 수행하게 된다.

2.3 LD를 이용한 블록분할

별개의 패턴수를 최소로 가지는 블록 분할은 전산학적인 의미만을 가지는 블록 경계를 생성하게 된다. 이 방법을 이용하고, 일반화된 불균형 계수 (Normalized disequilibrium coefficient) D'을 계산하여 Recombination이 없는 구간을 블록으로 결정하여 블록 경계의 생성에 생물학적인 의미를 고려한 블록 분할 방법을 구현하였다. 입력된 데이터로부터 D'을 계산하여 블록을 판정하기 위해 [그림 3]과 같은 분할표를 이용한다.

	B_1	B_2	
A_1	n_{11}	n_{12}	
A_2	n_{21}	n_{22}	
			N

[그림 3] 입력된 데이터로부터 구성된 분할표

A는 $pA\cdot$ 를, B는 $p\cdot B$를 의미한다. 또한, A_1, B_1은 major alllele, A_2, B_2는 minor allele로 구성된 데이터임을 의미한다.

n_{ij}는 $pA\cdot_i\ p\cdot B_j$로 정의된다. 예를 들어 n_{12}의 경우는 $pA\cdot_1 p\cdot B_2$로 구성된 데이터를 의미한다. N은 해당 블록의 총 샘플수의 합이다. 불균형 계수 D' (Disequilibrium coefficient)는 다음과 같은 식을 통해 계산된다.

$$D = h_{11} - pA_1pB_1$$

불균형 계수 D는 n_{11}, n_{12}, n_{21}, n_{22}중 어느 것을 기준으로 하여도 같은 값이 나오며 본 논문에서는 n_{11}을 기준으로 하여 계산하였다. 계산된 D 값은 $D>0$ 또는 $D<0$인 각각의 경우에 대해 다음 식을 통해 0부터 1사이의 값을 가지는 $|D|_{max}$를 계산하게 된다.

$$|D|_{\max} = \begin{cases} min(p_{A_1}p_{B_2}, p_{A_2}p_{B_1}) & \text{if } D>0 \\ min(p_{A_1}p_{B_1}, p_{A_2}p_{B_2}) & \text{if } D<0 \end{cases}$$

최종적으로 일반화된 불균형 계수 D'은 다음과 같은 식을 통해 계산된다.

$$D' = \frac{D}{|D|_{\max}}$$

D'은 1과 -1 사이의 값을 가지며, 판단기준으로 사용할 경우 절대값을 취해 0과 1 사이의 값인 $|D'|$을 블록 판정에 이용한다. 본 논문에서는 Gabriel [2]과 같은 기준으로 LD를 판정하였다. 계산된 $|D'|$ 값은 marker allele 빈도에 종속되므로, 분할표 상에서 계산된 $|D'|$ 값이 1인 경우에도 Recombination이 일어난 데이터가 될 수 있는 통계적 오류가 발생하게 된다. 이러한 경우를 블록 분할 알고리즘에서 검정하여 블록 경계를 결정하기 위해 Pearson의 χ^2 통계량을 이용하였다. χ^2은 다음의 식을 통해 적합도 검정을 수행하였다.

$$\sum \frac{(obs-exp)^2}{exp} > \chi^2(1\,;1-\alpha)$$

관찰값 (obs)은 분할표의 n_{11}, n_{12}, n_{21}, n_{22}에 대해 각각 계산되며, 기대값 (exp)은 분할표로부터 각 n_{ij}에 대해 계산된다. 자유도 df (degree of freedom)는 (행 범주 개수 - 1)(열범주 개수 - 1)을 계산하여, $\chi^2(1df)$로써 유의수준 판단을 하였다. χ^2 분포표에서 df = 1이며, 1 - α = 0.95일 때 5% 유의 수준에서 χ^2이 P = 3.84 이상일 경우 LD 블록으로 인정한다.

3. 데이터 집합 및 성능 평가

사용된 데이터는 R. Hudson [6]의 Neutral 모델을 통해 샘플을 생성하는 프로그램인 ms를 이용하여 만든 80개 샘플과 234개의 SNP으로 이루어진 인공 데이터이다. 비교 대상으로 같은 Dynamic Programming Algorithm을 이용하여 구현된 HapBlock [7, 8]을 이용하였으며, 블록의 결정에 염색체 적용 범위(Patil et al. [1]) 80, 90% 이상을 각각 설정하여 블록 분할을 수행하였다.

3.1 인공 데이터를 사용하여 구현한 Coverage 설정 방법과 HapBlock의 결과 비교

[표 1]에 나타낸 결과에서 HapBlock [3, 4]의 적용범위를 α = 80%로 주었을 때, 평균 블록의 길이는 5.20개, 패턴의 개수는 12.82개이다. 구현한 방법은 α= 80%, β= 90%로 설정하였을 때 평균 블록 길이 8.67개와 패턴의 개수 8.37개이다.

사용된 프로그램	Coverage α %	Coverage β %	합계 및 평균	블록의 길이	2개 이상인 Common Haplotype 개수	분석에 포함되는 총 SNP	패턴 개수	모호한 부분
제안하는 방법	70	70	합계	234	821	13127	169	88
			평균	16.71	58.64		12.07	6.29
		80	합계	234	821	13127	169	88
			평균	16.71	58.64		12.07	6.27
		90	합계	234	828	13067	167	71
			평균	16.71	59.14		11.93	5.07
	80	70						
		80						
		90						
Zhang HapBlock	80							

[표 1] Coverage α, β를 변경시킨 실험결과

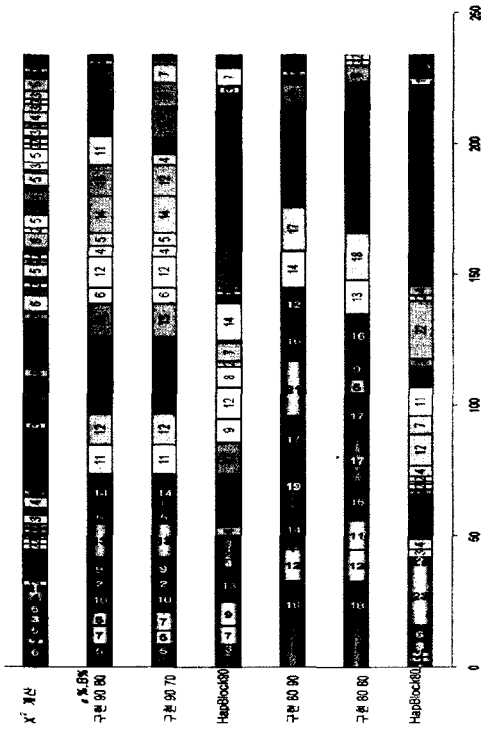
분석에 포함된 총 SNP의 수는 HapBlock [3, 4]이 14269개이며, 구현한 방법은 14735개를 보여, 구현한 방법이 보다 많은 SNP의 분석이 가능하다는 것을 보여준다. 분석에 포함되는 총 SNP 수란 Missing을 가져 모호함을 포함한 부분(Ambigous한 Data로 구분된 것)과 1번 밖에 나오지 않은 Rare Haplotype을 제외한 흔한 Haplotype을 구성하는 SNP의 총 수로써, 많을수록 전체 Sample을 Loss없이 분석함을 나타낸다. 평균 블록의 길이는 α=80일 때 기존의 Hapblock [3, 4]의 경우 5.20개이며, 구현한 방법은 각각 12.31개와 12.32개를 보인다. 블록의 길이가 길고 패턴의 개수는 더 작은 값을 보이는 것을 볼 수 있다.

제안하는 방법으로부터 생성된 블록이 적은 수의 패턴과 블록 수를 가진다면, 이는 적은 수의 Tag SNP을 가진다는 의미가 되므로, HapBlock에서 나타나는 패턴과 블록의 길이를 비교하였다. 표본의 수가 작을 경우에는, HapBlock 방법이 제안하는 방법에 비해 더 적은수의 Tag SNP을 가지나, 표본의 수가 많아지면 기존의 HapBlock은 각 블록의 Tag SNP을 계산하기 힘들어지며 수행 시간도 많이 늘어나게 된다. 이러한 경우는 표본에서 나타나는 패턴의 수를 최소로 하는 블록으로 분할하고 분할된 블록에서 Tag SNP을 구하는 것이 보다 효율적이라 판단하여 시스템을 구성하였으며, 표본의 수가 많은 Haplotype 데이터의 블록 분할에서 성능 향상을 확인할 수 있다.

3.2 Coverage 설정 및 LD에 기반한 블록 분할, HapBlock이 각각 수행된 결과의 블록 경계 비교

[그림 4]는 Coverage 설정 및 LD에 기반한 블록 분할 결과와 HapBlock [3, 4]과의 비교를 위해 블록 분할 결과를 제시하였다. LD를 계산하고 χ^2 검정을 수행하여 분할된 총 블록의 개수는 91개로, 많은 블록 수를 보여 개선이 필요하다.

Coverage를 설정하여 구현한 방법은 분할된 블록내의 패턴수를 줄이면서 블록의 길이가 큰 블록 분할을 수행하여 좋은 결과를 보인다. 구현한 방법의 Coverage는 순서대로 각각 α%, β%인 경우의 블록 경계를 나타낸다.



[그림 4] 블록 분할 방법들의 블록 경계 비교

4. 결론 및 향후 연구과제

본 논문에서는 많은 샘플로 구성된 데이터를 분석하기 위해 별개의(distinct) Haplotype의 수를 줄이는 블록 분할 방법을 구현하여, 좋은 성능을 확인하였다. 이와 더불어 블록 경계의 생성에 생물학적인 의미를 고려하기 위하여 $|D'|$을 계산하여 블록을 분석하였다. Missing 데이터로 인해 나타나는 Noise 부분의 처리와 다양한 LD 측정 방법 및 모델을 적용하여 성능을 향상시키는 방법을 구현중이다.

[참 고 문 헌]

[1] N. Patil, et al, "Blocks of Limited Haplotype Diversity Revealed by High-Resolution Scanning of Human Chromosome 21", *SCIENCE*, vol. 294, pp. 1719-1722, November 2001.

[2] S. B. Gabriel, et al, "The Structure of Haplotype Blocks in the Human Genome", *SICENCE*, vol. 296, pp. 2225-2229, June 2002.

[3] K. Zhang, "A Dynamic Programming Algorithm for Haplotype Blocks partitioning", *PNAS*, vol. 99. no. 11, pp 7335-7339, 2002.

[4] K. Zhang, "Dynamic Programming Algorithms for Haplotype Block Partitioning : Applications to Human Chromosome 21 Haplotype Data", *ACM RECOMB*, pp. 332-340, 2003.

[5] Mark. J. Daly, John D. Rioux, Stephen F. Schaffner, Tomas J. *Hudson, and Eric S. Lander. High-resolution haplotype structure* in the human genome, *Nature Genetics*, vol. 29, pp. 151-158, 2001.

[6] Hudson. R. R. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics* 18: 337-338, 2002.