올리고뉴클레오타이드 제작을 위해 효율적이고 차별적인 시드를 고르는 방법에 대한 고찰

(A Study of Choosing Efficient Discriminative Seeds for Oligonucleotide Design)

정 원 형 † 박 성 배 ††
(Won-Hyong Chung) (Seong-Bae Park)

요 약 생물정보분야에서 올리고뉴클레오타이드(oligonucleotide)를 제작하는 문제는 시간을 많이 소모하는 문제이다. 이 문제를 해결하기 위하여 해시를 이용한 가속계산이 주로 쓰이고 있고 BLAST란 프로그램이 대표적으로 생물정보분야에서 사용되고 있다. BLAST류의 프로그램들은 DNA서열의 특성에 따라 시드를 변형하여 해시를 개선하는 알고리즘을 적용하여 서열간의 유사도가 높은 부분을 찾는다. 그러나 이 프로그램들은 원래 올리고뉴클레오타이드 제작을 위해서가 아닌 지역정렬 문제를 해결하기 위한 방법들로써 발전하여 왔으므로 본 문제에 효율적인가에 대한 검증이 아직까지 이루어지지 않았다. 우리는 BLAST류의 프로그램에서 사용된 시드(seed)들이 올리고뉴클레오타이드 제작에 효과적인가를 판단할 수 있는 효율적이고 차별적인 잣대를 제시하고 이에 따라 다섯 종류의 대표적인 시드를 평가하였다. 평가에서 spaced seed라는 시드가 가장 좋은 결과를 보임을 정량적으로 계산할 수 있었다.

키워드 : 올리고뉴클레오타이드, 프로브, 해시, 시드, 생물정보, 블라스트

Abstract Oligonucleotide design is known as a time-consuming work in Bioinformatics. In order to accelerate the oligonucleotide design process, one of the most widely used approaches is the prescreening unreliable regions using hashing(or seeding) method represented by BLAST. Since the seeding is originally proposed to increase the sensitivity for local alignment, the specificity should be considered as well as the sensitivity for the oligonucleotide design problem. However, a measure of evaluating the seeds regarding how adequate and efficient they are in the oligo design is not yet proposed. we propose a novel measure of evaluating the seeding algorithms based on the discriminability and the efficiency. By the proposed measure, five well-known seeding algorithms are examined. The spaced seed is recorded as the best efficient discriminative seed for oligo design.

Key words : oligonucleotide, probe, hash, seed, bioinformatics, BLAST

· 본 논문은 지식경제부 및 정보통신진흥원의 정보통신선도기반기술개발사업(A1100-0601-0102)의 연구결과로 수행되었습니다. 또한 본 논문은 2008년도 2단계 두뇌한국(BK)21사업에 의하여 지원되었습니다.
· 이 논문은 2008 한국컴퓨터종합학술대회에서 '올리고뉴클레오타이드 제작을 위해 효율적이고 차별적인 시드를 고르는 방법에 대한 고찰'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 경북대학교 컴퓨터공학과
whchung@sejong.knu.ac.kr
†† 종신회원 : 경북대학교 컴퓨터공학과 교수
sbpark@sejong.knu.ac.kr

논문접수 : 2008년 8월 25일
심사완료 : 2008년 11월 9일

정보과학회논문지 : 시스템 및 이론 제36권 제1호(2009.2)

1. 서 론

휴먼게놈프로젝트를 계기로 생물분야에서 서열을 통한 분자생물학적 분석방법이 발전함에 따라 서열의 양이 폭발적으로 증가하고 있다. 이렇게 서열의 양이 증가함에 따라 대량의 서열을 이용하여 정보학적인 접근으로 보다 정확한 생물학 실험이 가능해졌다. 대표적인 정보학적 접근법으로 올리고뉴클레오타이드(올리고)의 제작이 있다. 올리고는 15에서 100bases 정도의 길이를 가지는 DNA(또는 RNA) 서열로써 뉴클레오타이드의 상보결합(=hybridization) 특성을 이용하여 목적하는 서열과 결합하도록 제작된다. 이는 원하는 서열을 생물학 실험상에서 찾아내는 탐침(probe)또는 서열을 증폭하는 프라이머(primer) 등으로 생물학에서 널리 쓰이고 있다.

또한 유전자 결정(Gene identification), PCR 증폭, DNA 마이크로어레이 등 생물학 실험의 기초단계에서 활용되고 있다.

올리고를 A, C, G, T(또는 U)의 네가지 알파벳을 가지는 문자열로 보면 올리고 제작문제는 컴퓨터분야의 스트링 매칭문제로 환원하여 풀이될 수 있다. 서열의 상보결합은 생화학적인 특성상 정확하게 상보적이지 않아도 결합할 수 있으므로 이는 부정확한 스트링 매칭(inexact string matching) 문제로 한정되고 이는 dynamic programming등의 알고리즘으로 풀이가능하나 계산량이 증가하게 된다. 따라서 결합이 일어나기 희박한 서열조각들을 cross-hybridization을 검사하기 전에 미리 제거하여 계산량을 줄이는 사전작업이 필요하고 이에 많은 휴리스틱 알고리즘이 적용되었다. 대표적인 알고리즘으로 다중 정렬(multiple alignment)[1], 서픽스 트리(suffix tree)[2], 해싱(hashing)[3,4]이 쓰였고 이중에서 해싱을 이용한 필터링이 가장 널리 쓰이고 있다. 생물정보분야에서 가장 잘 알려진 프로그램 중의 하나인 BLAST[3]는 서열에서부터 모든 k 길이의 단어를 해싱하여 중복된 단어들을 찾은 후 그 단어들을 정해진 임계값까지 좌우로 확장하며 서열내의 유사성이 있는 부위를 찾아낸다. 이 프로그램은 시드(seed)라고 불리는 k 길이의 단어가 정확히 일치하는 부분을 탐색한다. Ma는 그의 연구[4]에서 시드의 매칭패턴을 변화시킴으로써 검색 민감도(sensitivity)를 증가시킬 수 있음을 보였으며 이후 다양한 형태의 시드를 도입함으로써 유사서열의 검색 성능을 개선하고자 하는 알고리즘들이 많이 제시되었다[5].

현재 올리고 제작에 시드를 도입하여 속도 및 성능향상을 추구하는 방법이 점차 증가하고 있지만, 최근까지 개발된 시드들은 올리고 제작을 위해서가 아닌 지역정렬(local alignment)문제를 해결하기 위한 방법들로써 발전하여 왔으므로 올리고 제작 문제에 효율적인가에 대한 검증이 아직까지 이루어지지 않았다. 본 논문에서는 올리고 제작에서 효율적인 시드를 평가할 수 있는 잣대를 제시하고 이에 따라 기존에 알려진 다섯 종류의 시드를 평가하여 어떤 시드가 올리고 디자인에 가장 적합한가를 제시한다.

본 논문은 2장에서 지역정렬 문제와 올리고 제작 문제에서 시드가 문제 해결방법에 따라 다른 잣대로 적용되어야 함을 언급하고 올리고 제작에 효율적인 시드를 평가하는 방법을 효율성과 분리성의 두 잣대로 정의한다. 3장에서는 정의된 평가방법에 의하여 시드를 평가한 실험방법을 제시하고 실험에 사용된 다섯 종류의 시드에 대하여 설명한다. 4장에서는 본 논문에서 제시한 평가방법에 의하여 다섯 종류의 시드를 효율성과 차별성, 그리고 종합적인 측면에서 평가한 결과를 보인다. 마지막으로 본 논문을 정리하고 향후 연구되어야 할 과제에 대하여 논의하며 마무리 짓는다.

2. 문제 정의

생물정보학에서 지역정렬 문제는 DNA 또는 Amino acid 서열에서 정해진 임계값 이상의 유사도를 공유하는 영역을 찾는 문제이다. 반면 올리고 제작 문제는 DNA 또는 RNA 서열집합에서 개별 서열을 대표하는 짧은 길이의 서열조각을 찾는 문제이다. 두 문제는 공통적으로 답이 될 가능성이 희박한 서열부분을 제외시키는 사전작업을 통하여 계산속도를 향상시킬 수 있고, 가장 잘 알려진 방법으로 해싱이 있다. 이 방법은 시드길이에 따라 탐색범위가 달라지는 문제가 있다. 시드가 길어지면 시드에 의해서 필터링되는 영역이 늘어나면서 답이 될 가능성이 있는 부분까지 제외될 수 있고, 반면에 시드가 짧아지면 답이 될 가능성이 희박한 영역이 필터되지 않아 답이 아닌 영역을 더 많이 선택하게 된다.

이 문제를 해결하기 위하여 PatternHunter[4]에서는 새로운 시드 적용방법을 제안하였다. 이것은 k 길이의 시드가 정확하게 일치되는 부분을 해싱하는 BLAST와 달리 n+k 길이의 불연속 시드를 사용하는데 이는 일치되지 않아도 되는 n 개의 don't care 위치들을 시드 안에 부여하고 대신 나머지 k 개의 위치에서 정확하게 일치되는가를 탐색함으로써 BLAST의 시드와 같은 민감도를 보장한다. BLAST에서 사용된 시드를 continuous seed라고 부르고 PatternHunter에서 사용된 시드를 spaced seed 이라고 칭하고 정확하게 일치되어야 하는 k 개의 위치를 시드의 무게라고 한다[5]. 예를 들어 spaced seed의 don't care 위치를 0, 일치되어야 하는 위치를 1로 표기한다면 서열 "ATCCAG"와 "ATCAAG"는 "111011" 이란 무게 5의 spaced seed 패턴으로 검색했을 때 일치되는 영역으로 판단이 되지만, 같은 무게의 continuous seed "11111"에 의해서는 네 번째 위치의 문자가 다르기 때문에 일치되는 영역이 아니라고 판단된다(그림 1). 본 예는 시드의 적용방법을 변형함으로써 같은 무게의 시드에서도 민감도에 차이가 있을 수 있음을 보여준다. 이후 시드를 개선하기 위한 연구가 지속적으로 이루어져 transition-constrainted seed [6], vector seed[7], BLAT seed[8] 등의 시드들이 활용되고 있다.

Spaced seed 등의 시드들은 원래 지역정렬 문제를 효율적으로 개선시키기 위해 제안되었고 이때 동일한 무게에서 좀 더 많은 유사영역을 탐색하는 시드를 찾는 것만이 이슈이다. 그러나 올리고를 제작하는 문제에서 시드를 평가할 제약조건은 지역정렬 문제보다 더 많은

```
      123456
      ATCCAG
      |||  ||
      ATCGAG
S1:   111011
S2:   11111
S3:    11111
```

그림 1 spaced seed를 사용하여 유사영역을 찾은 경우: S1은 spaced seed에 속하고 S2와 S3는 continuous seed에 속한다.

경우가 고려되어야 한다. 왜냐하면 지역정렬의 경우 시드에 의하여 탐색된 영역이 유사영역인지에 대한 평가기준이 올리고 디자인에 비하여 상대적으로 간단하기 때문이다. 올리고 디자인의 경우 시드에 의하여 탐색된 영역에서 올리고의 후보를 선정한 다음 3.1절에서 언급하는 것처럼 시간이 많이 걸리는 후작업이 반드시 수반되어야 한다. 올리고인지를 평가하는 후작업의 시간이 지역정렬에 비하여 많이 소모되는 것과 동시에 고려해야 할 사항은 올리고가 가능하다고 평가된 모든 부분이 올리고로 사용이 되지 않을 수도 있다는 것이다. 예를 들어 특정 유전자를 탐색하는 올리고가 여러 개 탐색되더라도 실제 사용할 것은 그 중 사용자의 실험조건에 가장 부합하는 하나이다. 따라서 올리고 디자인은 시드에 의해 탐색된 영역의 평가에서 지역정렬보다 더 많은 시간이 투입되어야 하고 탐색 결과가 모두 최종적으로 활용되는 것이 아니므로 유사한 영역을 최대로 찾는 것보다 올리고의 제약조건에 맞는 후보 영역의 개수를 적절히 조절하면서 계산 속도를 높일 필요가 있다.

따라서 올리고 디자인에서 시드의 선택은 다음과 같이 지역정렬 문제와는 다른 이슈를 가진다. 첫째, 시드는 가능한 한 많은 올리고를 찾을 수 있어야 하고, 둘째, 시드는 올리고가 아닌 영역에서 잘못된 올리고 후보를 찾지 말아야 하며, 셋째, 올리고를 찾기 위해 사용되는 시드의 개수가 적을수록 좋다. 첫째와 둘째 제약조건은 상반되는 성격을 가진 조건으로써 두 조건간의 균형점을 찾는 것이 중요하므로 이를 "차별성(Discriminability)"이라는 잣대로 정의하고 셋째 조건은 알고리즘 상에서 중복되는 시드를 사용함으로써 발생하는 낭비를 최소화하도록 "효율성(Efficiency)"이란 잣대로 정의한다. 최종적으로 우리는 차별성과 효율성을 동시에 고려하는 "효율적으로 차별적인(Efficient Discriminability)" 제약조건으로 통합하여 올리고 제작에서 시드의 성능을 평가한다.

2.1 차별성

차별성을 정의하기 전에 먼저 올리고 제작 시 시드를

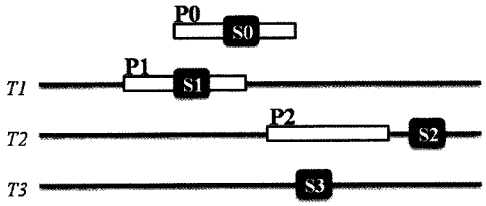

그림 2 올리고 제작에서 시드를 활용할 때 발생가능 한 경우들: T1~T3은 서열, P0는 올리고, P1과 P2는 P0와 매치되는 T1과 T2에서의 위치들이다. S0는 P0에 속하는 시드이고 S1~S3은 서열에서 시드와 매치되는 위치들이다.

이용하여 올리고가 매치하는 영역을 찾을 때 발생할 수 있는 경우들을 살펴본다. 시드의 탐색 위치를 통하여 올리고가 매치하는 위치를 찾는 경우를 테스트 조건으로 하고, 실제로 올리고가 매치하는 경우를 컨디션 조건으로 하자. 두 조건을 이진분류하면 다음의 네 가지 경우로 나누어진다.

- True positive(TP): 시드가 올리고 매칭위치를 찾음 (S_O). 다시 말해서, 올리고 매칭위치에서 시드가 발견됨(O_S)
- False negative(FN): 올리고 매칭위치에 시드가 발견되지 않음 ($O_{\bar{S}}$)
- False positive(FP): 시드가 발견된 위치가 올리고 매칭위치에 속하지 않음 ($S_{\bar{O}}$)
- True negative(TN): 시드가 발견되지 않은 위치에 올리고 또한 매치되지 않음

그림 2에서 목적하는 올리고를 P0라고 하고 P0가 매치하는 영역 P1, P2를 시드 S0의 탐색을 단서로 서열에서 찾고자 하는 상황을 추상적으로 표현하였다. S1이 P1에 겹친 경우는 목적하는 올리고를 시드를 단서로 찾은 경우(true positive)이고, S2와 S3는 시드가 찾은 영역이 올리고와 매치하지 않은 경우(false positive)이다. P2의 경우 올리고와 매치하지만 시드가 이를 찾지 못한 경우(false negative)이다.

차별성을 테스트할 이슈중의 하나인 시드가 올리고를 제대로 찾는 능력은 잘 알려진 통계적 방법인 precision으로 정의한다. Precision P는 올리고가 매치하는 영역에서 발견된 시드의 개수와 시드가 발견한 모든 위치의 개수의 비율로써 정의한다(식 (1)). 또 다른 이슈인 올리고가 아닌 영역에서 시드가 발견됨으로써 잘못된 올리고 후보를 찾는 경우를 최소화하는 능력은 recall로 정의한다. Recall R은 시드를 포함하는 올리고의 개수와 모든 올리고 매칭경우의 개수를 비율로써 정의한다(식 (2)).

$$P=\frac{TP}{FP+TP}=\frac{|S_O|}{|S_O|+|S_{\overline{O}}|} \quad (1)$$

$$R=\frac{TP}{FN+TP}=\frac{|O_S|}{|O_S|+|O_{\overline{S}}|} \quad (2)$$

위의 두 이슈는 태스트 정확도를 검사하는 대표적인 방법인 F-measure[9]로 통합하여 아래와 같은 수식으로 차별성을 정의한다(식 (3)) 실제 올리고 제작 에서 precision과 recall 중 어디에 더 무게를 두는가는 사용자의 의도에 따라 다르게 부여될 수 있다. 식 (3)에서는 F-measure의 인자 α를 조절하여 이를 조정할 수 있다. 인자 α의 변화에 따른 시드의 성능변화는 4.4절에서 다루기로 한다.

$$F_\alpha=\frac{(1+\alpha^2)PR}{\alpha^2P+R} \quad (3)$$

2.2 효율성

시드를 이용한 올리고 제작 시 두 가지 측면에서 시드 중복에 의한 효율성이 고려되어야 한다. 첫 번째는 해시를 구성하기 위하여 시드를 생성할 때 중복이 일어나는 경우이고, 두 번째는 올리고가 매치되는 영역에서 시드가 중복되는 경우이다. 현재까지 개발된 시드 중에서 BLAT, Vector Seed 등은 시드의 임의의 위치에서 발생하는 미스매치를 몇 개까지 허용함으로써 민감성은 증가하였으나 첫 번째의 경우처럼 동일한 위치에서 시드를 중복 선택하는 현상이 일어나고, 이는 해시크기를 증가시켜 메모리의 낭비를 유발한다. 따라서 올리고 제작의 효율성을 증가시키기 위해서는 중복 선택되는 시드의 개수를 줄여야 하고 이는 다음과 같이 정의된다(식 (4)).

$$D=\frac{number\ of\ generated\ seed\ hashes}{number\ of\ unique\ seed\ hashes} \quad (4)$$

또 다른 중복은 올리고 매치영역을 탐색할 때 발생한다. 예를 들어 "ATCCAG"라는 올리고와 동일한 영역 "ATCCAG"가 서열 내에 존재할 때 이를 무게 4의 continuous seed로 탐색한다면 올리고는 "ATCC", "TCCA", "CCAG"의 세 번의 시드탐색이 하나의 올리고를 찾는데 관여하지만, 실제로는 셋 중 하나의 시드만 있어도 올리고는 찾을 수 있다. 반면 "110101"의 패턴을 가지는 무게 4의 spaced seed로 탐색한다면 올리고는 한 번의 탐색만으로 올리고를 찾을 수 있다. 이러한 올리고 매치영역 내에서의 시드 중복을 줄임으로써 효율을 높일 수 있고 이는 다음과 같이 정의된다(식 (5))

$$A=\frac{number\ of\ seed\ hashes\ in\ oligos}{number\ of\ oligos} \quad (5)$$

식 (4)와 (5)에서 정의한 잣대 D와 A는 모두 최소값을 가질 때 가장 효율성이 좋도록 정의되었다. 그리고 이들 수식에서 분자는 분모의 경우를 모두 포함하고 있으므로 이들의 최소값은 1이다. 따라서 각각의 잣대는 모두 1보다 같거나 큰 특성을 가지고 있고, 이를 각각 *1/(1+ 인자*log잣대)*의 형태로 정규화하고 곱하여 효율성으로 정의한다(식 (6)). 정규화에 의하여 D와 A 모두 그 값이 적을수록 효율성은 1에 가까워지고 D 또는 A의 값이 커질수록 효율성은 0에 가까워진다.

효율성은 올리고를 디자인하는 성능에는 관계가 없이 속도에만 관계가 있는 잣대이므로 D와 A가 얼마나 영향을 미칠지 또한 사용자의 의도에 따라 조절이 가능하여야 한다. 이는 식 (6)에서 D의 조절인자로 β, A의 조절인자로 γ를 적용하여 각각의 성능에 대한 가중치를 설정한다. 인자가 1인 경우 잣대의 영향이 최대로 적용되고, 0인 경우 그에 해당하는 변수의 영향이 없어진다. 인자 β와 γ의 영향에 대해서는 4.4절에서 다루기로 한다.

$$E_{\beta,\gamma}=\frac{1}{(1+\beta\log D)}\frac{1}{(1+\gamma\log A)} \quad (6)$$

2.3 효율적인 차별성

앞서 정의한 식 (3)과 식 (6)은 각각 차별성과 효율성을 평가할 수 있는 잣대이다. 이들은 0과 1의 값으로 한정된 결과를 보이고 최고값인 1일 때 가장 좋은 결과를 보이도록 정규화 되어 있다. 올리고 제작시 사용되는 시드의 성능을 평가하는 잣대 "효율적인 차별성"은 차별성과 효율성을 곱하여 아래와 같이 정의한다(식 (7)). 차별성과 효율성이 정규화 된 잣대이므로 이를 곱셈으로 표현하여 효율적인 차별성 또한 정규화가 되도록 하였다. 따라서 이 성질은 시드가 올리고 제작에 가장 적합할 때 1의 값을 가지게 된다. 다시 말해서, 효율적인 차별성이 1일 때 시드는 모든 올리고 매칭위치에서 한 번만 나타나고 이외의 위치에서는 나타나지 않으며 시드를 생성할 때 중복 없이 한 번만 생성되어 시드의 성능이 최상인 동시에 낭비가 없다는 것을 나타낸다. 인자 α는 차별성의 조정을, 인자 β와 γ는 효율성의 조정을 가능하게 한다.

$$G_{\alpha,\beta,\gamma}=F_\alpha E_{\beta,\gamma} \quad (7)$$

3. 실 험

본 논문의 실험을 수행한 프로세스는 다음과 같다.

1) DNA 서열의 집합을 준비한다. 본 실험에는 50길이의 DNA서열 1000개를 서열간의 유사도에 따라 고르게 준비하였다.
2) 평가를 위하여 각 서열에서 모든 가능한 올리고를 제작하고 매칭위치를 기록한다. (올리고 제작 조건은 3.1절에서 다룬다.)
3) 효율적인 차별성에 필요한 인자 α, β, γ를 지정한다. (본 실험에서는 기본적으로 1의 값을 주었다. 각 인자를 변형한 실험결과는 4.4절에서 다루었다.)

4) 실험에 사용할 시드를 선택한다. 본 실험에서는 다섯 종류의 시드들(3.2절 참조)을 7에서 25까지의 무게에 따라 제작하여 실험에 사용하였다.
5) 서열집합의 가능한 모든 위치에서 4)에서 정한 시드를 이용하여 해시를 구성한다.
6) 앞에서 구한 결과에 따라 4)에서 정한 시드에 대한 효율적인 차별성을 계산한다.
7) 시드를 교환하면서 4)에서 6)까지의 프로세스를 반복한다.

본 논문의 주된 실험은 50길이의 DNA서열을 대상으로 하였다. 70길이의 DNA서열에 대한 실험을 위와 동일한 작업으로 수행하고 그 결과를 4.5절에서 다루었다.

3.1 올리고 제작

실험에서 시드의 평가를 위해서 우선 준비된 서열 집합에서 정확한 올리고의 매칭위치를 파악했다. 올리고를 제작하는 순서는 모든 가능한 올리고를 선정하고, 느리지만 가장 민감한 지역정렬 알고리즘인 FASTA[10]을 통하여 각각의 올리고에 대한 매칭위치가 될 후보를 선정하였다. 올리고와 매칭 후보서열은 전역정렬 알고리즘인 CLUSTALW[1]를 통하여 정렬하여 정확한 서열 유사도를 계산하고 OligoArrayAux[11]를 이용하여 자유에너지를 계산하였다. 앞에서 구한 결과들을 종합하여 올리고와 매칭이 될 후보서열이 결합(hybridization)이 일어날지 계산하여 매칭위치를 선정하였다.

올리고와 매칭위치의 서열은 결합(hybridization)이라는 생화학적 반응에 의하여 실제 결합여부가 정해지지만, 최근 생물화학분야의 연구를 통하여 실험없이 서열간의 비교를 통하여 결합여부를 결정하는 방법이 제시되어 왔다. Kane 등의 연구[12]는 50base의 올리고가 결합하는 조건으로 75%의 유사도와 15base 이상의 연속적인 매칭을 제시하였고, 최근 He 등의 연구[13]에서는 85% 유사도에 15base 이상, 그리고 -30Kcal/mol 이하의 자유에너지를 평가 조건으로 제시하였다. 본 논문에서는 [13]의 연구결과를 올리고의 매칭위치와의 결합여부를 판단하는 조건으로 사용하였다.

3.2 평가에 사용된 시드

본 논문에서는 생물정보 분야에서 잘 알려진 시드 다섯 가지를 선정하여 평가하였다.

- continuous seed: 시드와 정확하게 매치하는 영역을 해싱함. BLAST에서 사용됨.
- spaced seed: 시드에 don't care 위치를 삽입하여 비연속적인 매치 패턴을 해싱함. PatternHunter에서 사용됨.
- transition-constrained seed: spaced seed와 유사하나 생물학적으로 transition에 해당하는 경우(A와 G, 또는 C와 T가 불일치)매치와 미스매치 사이의 중간값을 부여하는 시드. YASS[6]에서 사용됨.
- BLAT seed: continuous seed의 임의의 위치에서 미스매치를 허용하는 시드.
- vector seed: spaced seed의 비연속적인 매치 패턴과 BLAT seed의 임의의 위치에서의 미스매치를 모두 허용하는 시드.

4. 결 과

본 장에서는 다섯 종류의 시드(Continuous, Spaced, Transition-constrained, BLAT, Vector)에 대한 올리고 제작에서의 성능을 차별성, 효율성, 효율적인 차별성으로 나누어 비교하고, 인자 α, β, γ가 시드의 성능측정에 미치는 영향에 대하여 논의한다. 그리고 본 논문에서 제시된 평가방법으로 실제 올리고 디자인된 결과를 평가하여 적절한 올리고를 제시한다.

4.1 차별성 측정 결과

그림 3은 다섯 종류의 시드를 무게 7부터 무게 25까지 증가시키며 차별성을 실험한 결과를 보여준다. 이 결과에서 무게 12의 spaced seed는 0.96의 값으로 가장 좋은 성능을 보였다. 무게 11의 continuous seed와 무게 12의 transition-constrained seed는 근소한 차이로 낮은 성능을 보였다. 전체적으로 transition-constrained seed와 spaced seed는 비슷한 패턴을 보였다. 이는 transition-constrained seed가 spaced seed에 transition에 해당하는 미스매치가 더 자주 일어나면 스코어를 보상하도록 개선된 시드지만 본 실험에서 사용한 서열은 랜덤하게 생성되었으므로 transition이 일어나는 확률이 같아서 거의 같은 결과를 보인 것이다. continuous seed는 무게 11까지 높은 성능을 보이다 이후 빠르게 차별성이 감소함을 보였다. 이는 시드의 무게가 11이하일 때 continuous seed는 유용함을 보여준다. 반면 BLAT seed와 vector seed는 0.3 부근의 낮은 성능을 거의 일정하게 보인다. 이것은 두 시드들이 민감도가 극

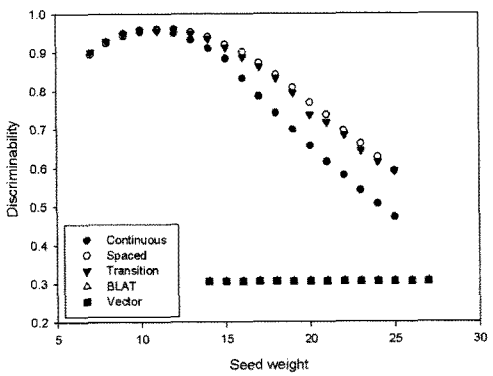

그림 3 다섯 종류의 시드에 대한 차별성

대화 되도록 변형된 형태의 시드이기 때문에 민감도와 관계가 있는 recall은 모두 1.0의 좋은 결과를 보이나 precision에서 약 0.18의 낮은 성능을 보였기 때문이다.

4.2 효율성 측정 결과

그림 4는 다섯 종류의 시드를 무게 7부터 무게 25까지 증가시키며 효율성을 실험한 결과를 보여준다. 이 결과에서 모든 시드들의 성능은 시드의 무게가 늘어남에 따라 증가함을 보였다. 효율성이란 시드의 낭비가 줄어들면 성능이 좋아지는데 시드 무게가 늘어나면 시드의 precision이 늘어나는 현상이 반영되기 때문이다. 전체적으로 transition-constrained seed와 spaced seed는 비슷한 패턴을 보이며 가장 좋은 성능을 보였다. continuous seed는 무게 15부분에서 효율성이 상대적으로 느리게 증가함을 보였다. BLAT seed는 가장 낮은 효율성을 보여주었고 vector seed는 BLAT seed와 같이 낮은 성능을 보이지만 무게가 증가함에 따라 효율성이 빠르게 개선됨을 보여주었다. BLAT seed와 vector seed가 전체적으로 낮은 성능을 보이는 이유는 2.2절에서 밝혔듯이 해시를 구성할 때 중복된 시드의 선택이 자주 일어나기 때문이다.

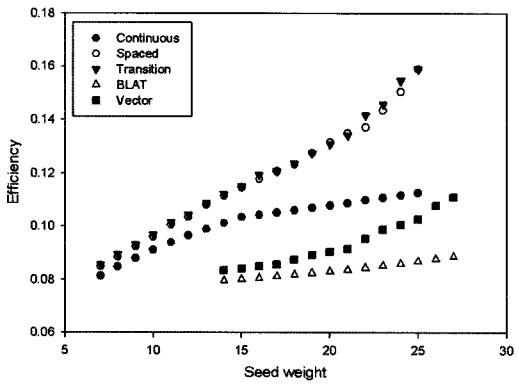

그림 4 다섯 종류의 시드에 대한 효율성

4.3 효율적인 차별성 측정 결과

그림 5는 다섯 종류의 시드를 무게 7부터 무게 25까지 증가시키며 효율적인 차별성을 실험한 결과를 보여준다. 이 결과에서 무게 16의 spaced seed는 0.106의 값으로 가장 좋은 성능을 보였다. 전체적으로 transition-constrained seed와 spaced seed는 비슷한 패턴을 보였지만 무게 15이상의 시드에서 상대적으로 약간 낮은 성능을 보여주었다. continuous seed는 효율성이 고려되면서 spaced seed, transition-constrained seed보다 확연히 낮은 성능을 보였다. 그리고 무게 15까지 성능이 증가하다 이후 빠르게 감소함을 보였다. 이는 올리고의 매치를 결정할 때 길이 15이상 연속으로 매치가

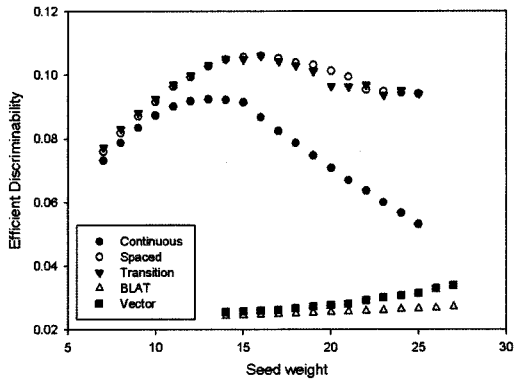

그림 5 다섯 종류의 시드에 대한 효율적인 차별성

되는 영역을 매치된다고 판단한 제약조건의 영향으로 무게 15까지는 전체적인 성능 향상이 있지만 이후 성능에 보탬이 되는 기술적 요인이 없기 때문이다. BLAT seed와 vector seed는 0.02 부근의 낮은 성능으로 시작하여 점차 성능이 증가함을 보였다. 이것은 두 시드들이 차별성과 효율성에서 낮은 성능으로 시작하기 때문이다. vector seed의 경우 무게가 증가함에 따라 성능이 상대적으로 많이 증가함을 보였다.

4.4 차별성과 효율성에서 인자의 영향 측정 결과

차별성과 효율성의 성능에 영향을 미치는 인자 α, β, γ는 올리고 디자인에 사용되는 서열의 종류와 사용자의 의도에 따라 조절되어야 하는 것들로 일반적인 최적의 값을 정의하기 곤란하다. 그렇지만 각 인자들의 성격과 성능에 미치는 영향의 정도를 확인함으로써 각 인자의 조절에 참고할 수 있다.

차별성의 성능에 영향을 미치는 인자 α는 차별성의 두 성질, precision과 recall의 영향을 조절하는 가중치의 성격을 가진다. precision은 시드의 무게가 증가함에 따라 증가하고, recall은 시드의 무게가 증가함에 따라

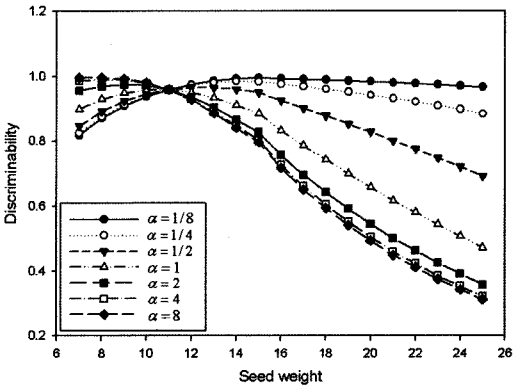

그림 6 인자 α의 변화가 차별성에 미치는 영향

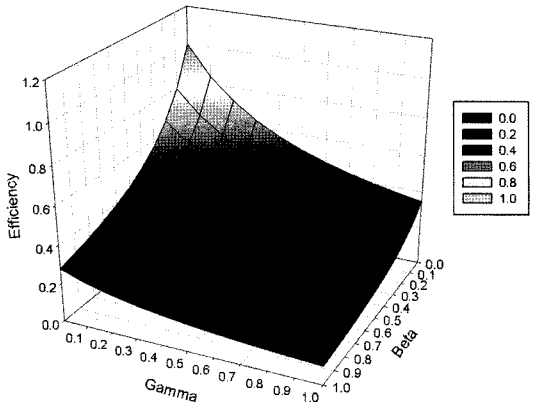

그림 7 인자 β, γ의 변화가 효율성에 미치는 영향

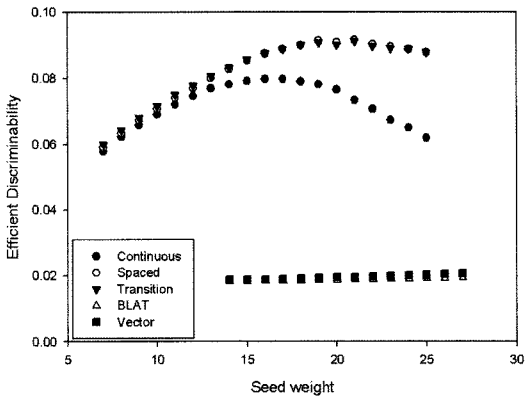

그림 8 길이 70의 올리고에서의 효율적인 차별성

감소하는 상반관계를 가진다. 따라서 α가 1일 때 precision과 recall에 같은 가중치가 주어지고, α가 증가하면 precision에 가중치를, α가 감소하면 recall에 가중치를 증가시키는 결과를 가진다. 그림 6은 α의 변화에 따른 차별성의 변화를 시드의 무게에 따라 측정한 결과이다. 올리고 디자인 시 가능한 한 많은 올리고를 찾고자 한다면 α를 감소시켜서 recall의 가중치를 증가시키고, 탐색시간을 줄이고자 한다면 α를 증가시켜 precision의 가중치를 증가시키면 된다.

효율성의 성능에 영향을 미치는 인자 β와 γ는 올리고 디자인 시 해시의 생성 및 활용에서 낭비되는 요소를 평가에 얼마나 반영할 지를 결정하는 인자들이다. 차별성과 달리 효율성의 두 성질 D와 A는 상반관계가 아니기 때문에 두 성질을 각각 조절하는 인자가 따로 필요하다. 그리고 효율성은 올리고를 디자인하는 결과에는 관계없이 속도에만 관계있는 잣대이므로 효율적인 차별성에서 효율성은 차별성에 대하여 가중치의 성격을 가진다. 그림 7은 D=0.28, A=0.35 인 경우 인자 β, γ가 효율성에 미치는 영향을 그래프로 표현한 것이다. 두 인자는 1의 값이 주어졌을 때 각각의 성질에 대한 패널티를 전부 반영하므로 효율성은 0.097의 값을 가진다. β=1, γ=0 인 경우 D의 패널티만 반영되므로 0.28의 값을 가지고, 그 반대의 경우 A의 패널티 0.35의 값을 가진다. 두 인자가 모두 0일 때 효율성에 각각의 성질이 주는 영향이 없어져서 효율성은 1의 값을 가지게 된다. 따라서 효율적인 차별성의 측정에서 효율성을 고려하지 않으려 한다면 β와 γ를 모두 0으로 주면 된다.

4.5 올리고 길이변화에 따른 효율적인 차별성

올리고 디자인에 적합한 시드를 선택하는데 있어서 올리고의 길이가 변함에 따라 그에 적합한 시드의 무게도 변함을 확인하였다. 그림 8은 길이 70의 올리고를 대상으로 한 다섯 시드들의 효율적인 차별성을 측정한 결과이다. 길이 70의 경우 최적의 시드는 무게 21의 spaced seed였다. 그림 5의 길이 50인 올리고를 대상으로 한 결과와 그림 8의 길이 70의 결과를 비교했을 때, 올리고 길이의 변함이 시드의 종류의 우열을 바꾸는 결과를 보여주지는 않았다.

5. 결론 및 향후과제

본 논문에서 우리는 올리고 제작에 사용되는 시드의 성능을 측정할 수 있는 새로운 잣대를 제시하였다. 기존의 시드는 올리고 제작이 아닌 지역정렬에 특화되어 개발되었으므로 BLAT seed와 같이 지역정렬에서 좋은 성능을 보이는 시드가 의외로 올리고 제작에서는 효과적이지 않을 수 있다는 것을 보였다. 우리가 제시한 "효율적인 차별성"은 0과 1의 값으로 제한되고 값이 증가할수록 올리고 제작에 더 적합한 시드임을 보였다.

이 잣대에 따라 시드는 올리고가 매치하는 위치를 더 많이 찾고 올리고가 아닌 영역을 탐색하는 오류를 줄이면서 이때 시드가 중복 적용되는 낭비가 줄어드는 방향으로 개선됨을 하나의 값으로 평가될 수 있다. 우리는 생물정보 분야에서 잘 알려진 시드 다섯 가지(continuous, spaced, transition-constrained, BLAT, vector)를 선정하여 차별성, 효율성, 그리고 효율적인 차별성에 대하여 측정하였고, 그 결과 효율적인 차별성에서 가장 좋은 결과를 보인 시드는 올리고의 길이가 50일 때 무게 16의 spaced seed, 올리고의 길이가 70일 때 무게 21의 spaced seed임을 실험적으로 결과를 얻었다.

본 논문에서는 기존에 알려진 시드에 대해서 평가하는 작업을 수행하였지만 차후에는 효율적인 차별성을 바탕으로 올리고 제작에 가장 적합한 시드를 새롭게 제작하여 제시하는 작업이 필요하다. 그리고 시드의 개선으로 실제 생물학 실험에서 얼마나 더 좋은 결과를 얻었는가를 측정하는 작업이 필요하다. 또한 입력 서열에

따라 시드를 변형하여 기존의 올리고 제작 프로그램의 성능을 높이는 툴의 제작 또한 가능하다.

참 고 문 헌

[1] Thompson,J.D., Higgins,D.G., and Gibson,T.J. "CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," Nucleic Acids Res., 22, pp. 4673-4680, 1994.

[2] Gusfield,D. "Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology," Cambridge University Press, New York, NY, USA, 1997.

[3] Altschul,S.F., Gish,W., Miller,W., Meyers,E., and Lipman.D. "Basic local alignment search tool," J. Mol. Biol., 215, pp. 403-410, 1990.

[4] Ma,B., Tromp,J., and Li,M. "PatternHunter: faster and more sensitive homology search," Bioinformatics, 18, 3, pp. 440-445, 2002.

[5] Brown,D.G., Li,M. and Ma,B. "A TUTORIAL OF RECENT DEVELOPMENTS IN THE SEEDING OF LOCAL ALIGNMENT," J. BioInfo. Comp. Biol. 2, 4, pp. 819-842, 2004.

[6] No´e,L. and Kucherov,G. "YASS: enhancing the sensitivity of DNA similarity search," Nucleic Acids Res., 33, 2, pp. W540-W543, 2005.

[7] Brejova,B., Brown,D., and Vinar,T. "Vector seeds: an extension to spaced seeds allows substantial improvements in sensitivity and specificity," In Proceedings of the 3rd International Workshop in Algorithms in Bioinformatics, pp. 39-54, 2003.

[8] Kent,W.J. "BLAT - the BLAST-like alignment tool," Genome Res., 12, pp. 656-664, 2002.

[9] Rijsbergen, C. J. van. "Information Retrieval, second edition," Butter-worths. 1979. (http://www.dcs.gla.ac.uk/Keith/Preface.html)

[10] Pearson,W. "Searching protein sequence libraries: Comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms," Genomics, 11, pp. 635-650, 1991.

[11] Markham,N.R. and Zuker,M. "DINAMelt web server for nucleic acid melting prediction," Nucleic Acids Res., 33, pp. W577-W581, 2005.

[12] Kane,M., Jakoe,T., Stumpf,C., Lu,J. Thomas,J., and Madore,S. "Assessment of the sensitivity and specificity of oligonucleotide (50mer) microarrays," Nucleic Acids Res., 28, pp. 4552-4557, 2000.

[13] He,Z., Wu,L., Li,X., Fields,M. and Zhou,J. "Empirical establishment of oligonucleotide probe design criteria," Appl. Environ. Microbiol., 71, pp. 3753-3760, 2005.

정 원 형

1997년 경북대학교 컴퓨터공학과 졸업(학사). 1999년 경북대학교 대학원 컴퓨터공학과 졸업(석사). 2001년~2003년 (주)프로바이오닉 연구원. 2003년~2008년 한국생명공학연구원 연구원. 1999년~현재 경북대학교 대학원 컴퓨터공학과 박사과정. 관심분야는 생명정보학, 기계학습, 마이크로어레이, 정보추출

박 성 배

1994년 한국과학기술원 컴퓨터과학과 졸업(학사). 1996년 서울대학교 대학원 컴퓨터공학과 졸업(석사). 2002년 서울대학교 대학원 컴퓨터공학과 졸업(박사). 2004년~현재 경북대학교 컴퓨터공학과 교수. 관심분야는 기계학습, 자연어처리, 텍스트마이닝, 정보추출, 생명정보학