DNA칩의 Probe 선정을 위한 빠른 전처리 알고리즘

강승호° 최문호 정인선 임형석
전남대학교 전산학과
kinston@natural.chonnam.ac.kr°, howork@paran.com, isjung0@hotmail.com,
hslim@chonnam.ac.kr

Fast Preprocessing Algorithms to Select Probes of DNA chip

Seung-Ho Kang°, Mun-Ho Choi, In-Seon Jeong, Hyeong-Seok Lim
Dept. of Computer Science, Chonnam National University

요 약

DNA칩의 성능은 칩을 구성하는 probe에 의해 결정된다. 좋은 probe는 homogeneity, sensitivity, specificity와 같은 속성을 갖추어야 한다. 이중 specificity는 probe의 특정 유전자에 대한 선별적 결합 능력을 나타내는 것으로 이를 계산하는데 가장 많은 시간이 요구된다. 본 논문은 유전자의 개별 후보 probe들에 대한 선정 작업을 실행하기 전에 q-gram을 이용하여 비교가 필요한 유전자들만을 전체 유전자의 길이가 n인 경우 $O(1/4^q n^2)$의 시간에 선별하는 전처리 알고리즘을 제안한다. 그리고 제안한 알고리즘을 사용함으로써 기존 방법들보다 빠른 probe 선정이 가능함을 실제 데이터를 사용하여 보인다

1. 서 론

DNA칩은 유전자 발현에 관한 실험에 동시성을 제공함으로써 대규모의 유전자 발현 연구를 빠른 시간 내에 가능하게 하였다. 이를 이용하여 유전자 발현 과정 및 유전자 간의 관계 규명과 같은 기초 유전자 연구뿐 아니라 유전질환 관련 유전자의 존재 여부나 세균, 바이러스 등의 감염 여부를 신속히 알아낼 수 있고 개인의 유전적 특성에 따른 최적 약재의 사용도 가능하게 하였다[1].

probe들은 개별 유전자들의 고유한 내부 문자열로서 특정 유전자들의 지문역할을 하게 되며 유전자 칩의 성능은 선택된 probe들에 크게 의존한다.

칩의 성능을 향상시키기 위한 좋은 probe란 목적 유전자만을 대상으로 결합반응을 일으키고(specificity) 다른 probe들과 같은 실험 온도 내에서 반응이 일어나야 하며(homogeneity) 자신의 상보적 서열로 인한 이차 구조가 생성되지 않아야 한다(sensitivity).

이 중 homogeneity와 sensitivity는 개별 probe를 대상으로 선형시간 내에 계산이 가능하다. 그러나 전체 크기가 n인 유전자들을 대상으로 크기 m인 모든 후보 probe들의 specificity를 단순기법(Brute Force Technique)으로 계산하는 데는 $O(mn^2)$의 시간이 요구된다.

기존에 제안된 specificity를 만족하는 probe 선정 방법들은 유전자의 개별 후보 probe들 각각에 대해 해밍거리(Hamming Distance) 혹은 편집거리(Edit Distance)를 판별치(measure)로 계산하여 일정한 임계치 이상인 probe만을 선택한다. 하지만 이러한 방법은 대규모 유전자 집합에 대해서는 실용적이지 않다.

확률론적인 예측과 실제 데이터를 분석해 보면 특정 유전자의 후보 probe들을 임계치 미만으로 만드는 문자열을 가진 유전자들은 매우 적다는 사실을 알 수 있다. 본 논문은 이에 근거하여 각 유전자의 후보 probe들을 임계치 미만으로 만드는 유전자들만을 q-gram을 이용해 빠르게 선별해내는 전처리 방법을 제시한다.

2. 관련연구

좋은 probe가 갖추어야 할 성질을 Lockhart[5] 등이 제안한 이래로 이들을 만족하는 probe 집합을 빠른 시간 내에 찾기 위한 여러 연구들이 진행되어왔다. Li and Stormo[4]는 접미사배열과 Landscape을 이용하여 probe 집합을 선정하는 휴리스틱 알고리즘을 제시하였다.

하지만 이 방법은 대규모의 유전자 집합에는 사용하기 어려운 수행시간이 요구된다. Kaderali and Schliep[3]는 열역학적 성질을 고려하고 동적 프로그래밍 기법을 제안하였다. 이 방법은 열역학적 성질을 직접 고려하였다는 점에서 probe 선정 방법으로는 이상적이라 할 수 있다. 하지만 이 방법이 근거하고 있는 최인접이웃모델(Nearest Neighbor Model)[9]은 일반 수용액에서의 반응을 대상으로 성립된 모델로 DNA칩에 그대로 적용하기에는 문제가 있다. 또한 규모가 큰 유전자 집합에 적용하기에는 너무 많은 계산 시간을 요구한다. Rahmann[8,9]은 최장공통부분문자열을 사용하여 개별 probe들의 specificity를 판별함으로써 큰 규모의 유전자 집합에도 적용 가능할 정도의 시간상의 개선을 가져왔다. 하지만 이 방법은 크기가 30 이내인 작은 probe에만 사용 가능한 한계를 가지고 있다. Sung and Lee[11] 는 비둘기집 원리를 이용하여 실제 비교가 필요한 probe만을 대상으로 specificity를 판별함으로써 손실 없이 빠른 probe 집합 선정을 가능하게 하였다. 하지만 비교가 필요한 probe를 찾는 과정에 소요되는 시간에 대한 언급이 없다. 최근에 Gasieniec[2]등은 무작위서열을 이용하는 방법을 제시하였다. 하지만 이 방법은 개별 유전자들의 probe 전체를 찾지 않고 단지 하나 또는 몇 개만을 찾는다.

3. Probe 선택을 위한 유전자 선별 전처리 알고리즘

본 논문에서는 두 서열간의 불일치 문자 수를 의미하는 해밍거리를 판별치로 사용한다. 유전자들의 specificity를 만족하는 모든 probe들을 선정하는 문제는 다음과 같다.

정의 1: specificity를 만족하는 probe 선정 문제

한 유전체를 구성하는 유전자의 집합을 $G = \{g_1, g_2, g_3, \cdots, g_n\}$ 이라 할 때 다음 조건을 만족하는 개별 유전자 $g_i (1 \le i \le n)$의 크기 m인 모든 probe, p의 집합을 찾는다.

$HD(p, q) >= k$

q: $G - \{g_i\}$의 크기 m인 모든 부분문자열

k: 불일치 문자 수의 임계치

$HD(p, q)$는 크기 m의 부분문자열 p와 q의 해밍거리를 계산하는 함수이다. 이때 p가 자신이 속한 유전자 이외의 모든 유전자들에 대하여 위 조건을 만족하면 " 좋은" probe라하고 조건을 위반하게 하는 문자열이 하나라도 존재하면 " 나쁜" probe라 한다.

관찰 1. 크기 m인 probe의 허용 불일치 문자 수를 k라하고 유전자를 구성하는 네 가지 염기가 동일한 확률로 분포한다고 하면 임의의 probe p가 나쁜 probe일 확률은 다음과 같다.

$$\frac{1}{4^m}\sum_{i=0}^{k-1}\binom{m}{i}3^i$$

관찰 1로부터 특정 probe를 나쁜 probe로 만들 염색체 내의 부분 문자열은 아주 적음을 알 수 있다.

정의 2: 좋은 관계, 나쁜 관계

상대방 유전자에 대하여 나쁜 probe를 하나도 갖지 않는 두 유전자의 관계를 " 좋은 관계" 라 하고 좋은 관계에 있지 않은 두 유전자의 관계를 " 나쁜 관계" 라 한다.

소정리 1. 비교되는 두 유전자 g_i, g_j의 크기 m인 부분문자열 p, q중 가장 작은 $HD(p, q)$를 $minHD$이라 할 때, $minHD$가 임계치 k보다 크거나 같으면 두 유전자는 좋은 관계에 있다.

증명) 두 유전자의 크기 m인 부분문자열중 가장 작은 HD를 갖는 두 부분문자열의 HD가 좋은 probe가 되기 위한 조건인 임계치 k보다 크거나 같다면 다른 부분문자열들의 HD는 당연히 k보다 크거나 같다. 따라서 두 유전자의 크기 m인 모든 부분문자열은 좋은 probe의 조건을 만족하게 되며 두 유전자간에는 상대방에 대한 나쁜 probe가 존재하지 않는다.

따름정리 1. 두 유전자의 크기 m인 부분문자열중 일치되는 문자 수가 가장 큰 부분문자열의 일치되는 문자 수가 $m - k$보다 크면 두 유전자는 나쁜 관계에 있다.

probe 선정문제는 소정리 1과 따름정리 1에 의해 다음과 같이 해결 될 수 있다.

1) 개별 유전자들에 대해 나쁜 관계에 있는 유전자들을 선별한다.

2) 선별된 유전자들에 대해서만 개별 유전자들의 나쁜 probe를 제거한다.

특정 유전자에 대해 좋은 관계에 있는 유전자는 결코 나쁜 probe를 만들어 낼 수 없으므로 전처리 절차에 의해 선별된 나쁜 관계에 있는 유전자들에 대해서만 개별 probe들의 *HD*를 계산하여 specificity를 결정한다면 probe 선정문제는 해결된다.

그러나 이러한 방법이 빠른 시간과 정확성을 담보하려면 1) 나쁜 관계에 있는 유전자의 수가 적어야 하고 2) 전처리 절차가 정확하고 빠른 시간 내에 가능해야 한다. 관찰 1로부터 나쁜 관계에 있는 유전자 수가 많지 않음을 추정 할 수 있으며 4절에서 실제 실험을 통하여 이를 확인한다.

나쁜 관계에 있는 유전자를 찾기 위한 전처리 과정에 대한 문제를 아래와 같이 정의한다.

정의 3: 최대 일치 부분문자열 찾기 문제

임의의 두 유전자 g_i, g_j의 크기 m인 부분문자열 p, q중 일치하는 문자 수가 가장 큰 부분문자열을 찾는다.

$$\arg\min_{p\in g_i, q\in g_j} HD(p,q)$$

최대 일치 부분문자열을 찾는 문제에 대해 q-gram을 이용하여 어느 정도의 손실을 허용하지만 빠른 시간 내에 일치 부분을 추정하는 근사 알고리즘을 제시한다.

3. 1 실시간으로 q-gram의 개수를 계산하여 최대 일치 영역을 추정하는 근사 알고리즘

일치하는 문자 수가 가장 많은 크기 m인 부분문자열을 추정하는 방법으로 q-gram의 개수를 이용할 수 있다. 우선 개별 유전자의 서열을 그림 1의 a)처럼 q-gram 단위의 해시값으로 변환하고 해시값을 색인으로 하여 문자열내의 위치정보를 입력한 4^q크기의 위치정보 테이블을 생성한다. 또한 최대 일치 영역을 찾고자 하는 두 유전자에 대해 $|g_1| + |g_2| - 2q + 1$개의 크기 $m - q + 1$인 원형 큐들을 생성한다.

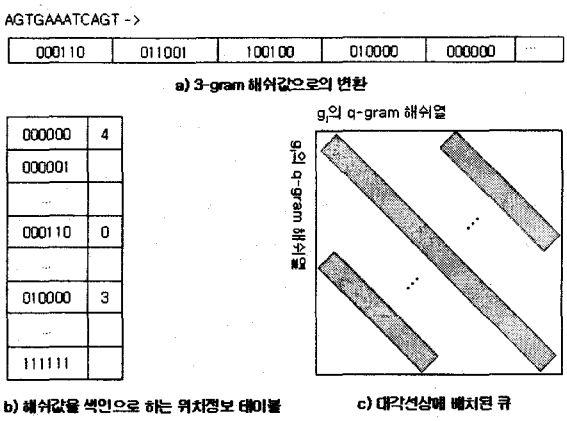


그림 1. 필요한 자료구조

이들 자료구조를 이용하여 일치하는 문자의 수가 가장 많은 크기 m인 부분문자열을 추정하는 절차는 다음과 같다.

1) 해시된 g_1의 q-gram 단위의 해시값 h_i 각각에 대해 이 값을 색인으로 하는 g_2의 위치정보 테이블의 q-gram들의 위치 j를 $i-j$번째 큐에 삽입하고 큐의 q-gram 개수를 1증가시킨다.

2) 새로 삽입된 q-gram과의 위치 차가 m이내인 q-gram의 위치들만이 큐 내에 유지되도록 삭제를 통하여 갱신한다.

이들 자료구조를 이용하여 q-gram의 개수를 계산하는 의사코드를 그림 2에 제시한다. 이렇게 계산된 q-gram의 수를 바탕으로 최대 수의 q-gram을 가진 영역이나 상위 $a(a > 1)$개 이상의 q-gram을 가진 영역만을 대상으로 일치되는 문자 수를 정확히 계산(이를 검증이라 한다)하여 두 유전자의 관계를 판별한다.

그림 1의 자료구조들을 생성하는 데는 선형시간만이 요구된다.

하나의 q-gram과 일치되는 상대편 유전자의 q-gram 수의 기대값은 $1/4^q * (|g| - q + 1)$ 이고 이 q-gram들은 원형큐에 한 번씩만 삽입, 삭제된다. 따라서 유전자 전체 길이의 합이 n일 때 알고리즘의 시간복잡도는 $O(\frac{1}{4^q}n^2) + T_v$이다. T_v는 검증에 요구되는 시간으로 검증 요건에 따라 다르다. 최대 수의 q-gram을 가진 영역을 대상으로 하는 경우엔 상수 시간이 되나 그 밖의 경우엔

정확성의 요구 정도에 따라 다르다. 실제 검증 영역의 개수는 표 3에 제시하였다.

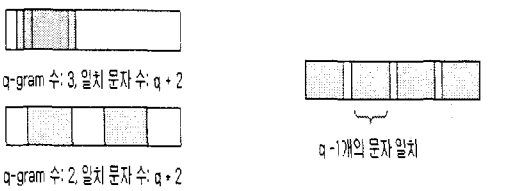
```
Algorithm BadGeneJudgementbySimpleQ-gramCounting
Input:  g1의 해시열 h, g2의 위치정보 테이블 T, m, k,
        검증요건 w
Output: g2가 g1과 나쁜 관계인지 여부
Begin
For each hi ∈ h
    For Thi 의 모든 j
        Insert(Queuei-j , j)
        Qgram_Counti-j ← Qgram_Counti-j + 1
        For j - fronti-j > m
            Delete(Queuei-j)
            Qgram_Counti-j ← Qgram_Counti-j - 1
        if Qgram_Counti-j > w
            if Verify(i, fronti-j, reari-j) = 1
                return 1 // 나쁜 관계
            else
                return 0 // 좋은 관계
End

Verify(i, fronti-j, reari-j )
Begin
i, fronti-j, reari-j 로부터 g1의 처음 위치 left 계산
For g1의 left, i와 g2의 fronti-j, reari-j를
   부분문자열로 갖는 모든 후보 probe쌍
   if 후보 probe쌍의 일치 문자 수 > m - k
      return 1
   return 0
End
```

그림 2. q-gram의 개수를 이용하여 최대 일치 영역을 추정하는 알고리즘의 의사코드

3.2 q-gram으로부터 일치되는 문자 수를 실시간으로 계산하여 최대 일치 영역을 추정하는 근사 알고리즘

q-gram의 개수만을 이용한 알고리즘은 그림 3의 a)같은 경우에 q-gram의 개수와 일치 하는 문자 수가 비례하지 않아 검증이 필요한 영역을 잘못 추정하는 문제점이 있다.



a) q-gram의 개수와 일치 문자 수가 비례하지 않는 경우 b) q-gram으로 일치 문자 수의 추정이 어려운 경우

그림 3. q-gram과 문자 수

따라서 큐 내에 갱신되는 q-gram으로부터 일치되는 문자 수를 실시간으로 추정하면 수행 시간을 크게 증가시키지 않으면서도 보다 정확한 추정이 가능하다.

```
Algorithm BadGeneJudgementbyMatchCounting
Input: g1의 해시열 h, g2의 위치정보 테이블 T, m, k,
       검증요건 w
Output: g2가 g1과 나쁜 관계인지 여부
Begin
For hi
      For Thi 의 모든 j
      Insert(Queuei-j , j) // i-j 번째 큐에 j 삽입
      if previous_reari-j + 1 = j
        Match_Counti-j ← Match_Counti-j + 1
      else
        Match_Counti-j ← Match_Counti-j + q
      For j - fronti-j > m
        if fronti-j + 1 = next_fronti-j
          Match_Counti-j ← Match_Counti-j - 1
        else
            Case 1: fronti-j + q - 1 < j + q- m
                Match_Counti-j ← Match_Counti-j - q
            Case 2: fronti-j + q - 1 ≥ j + q- m
                Match_Counti-j ← Match_Counti-j
                   - (j + q- m - fronti-j + 1)
                Create (fronti-j - j + m)-gram
                Queuei-j _front ← (fronti-j - j + m)-gram의
                                  문자열 위치
      Delete(Queuei-j) // i-j 번째 큐의 front 삭제
      if Match_Counti-j > w
        If Verify(i, fronti-j, reari-j) = 1
          return 1  // 나쁜 관계
        else
          return 0  // 좋은 관계
End
```

그림 4. q-gram을 이용하여 실시간으로 일치 문자 수를 추정하는 알고리즘

방법은 q-gram의 개수를 이용한 알고리즘과 유사하다. 다만 삽입과 삭제 시에 아래의 절차에 따라 일치되는 문자 수를 계산한다.

1) 새로 삽입되는 q-gram의 문자열 내 위치가 바로 이전에 삽입된 q-gram의 위치와 차이가 1 이면 일치되는 문자 수를 1 증가시키고 그렇지 않으면 q만큼 증가시킨다.

2) 새로 삽입되는 q-gram의 문자열 내 위치와 m 이상이 되는 q-gram들을 제거한다. 이때 제거되는 q-gram이 바로 다음 제거될 q-gram과 문자열내의 위치 차가 1 이면 일치되는 문자 수를 1 감소 시키고 그렇지 않으면 q만큼 감소시킨다.

3) 새로 삽입되는 q-gram의 문자열 내 위치와 m을

중심으로 걸치는 q-gram들은 걸치는 q-gram이 하나 남을 때까지 제거하고 1씩 감소 시킨다. 마지막 남은 q-gram은 m을 벗어난 영역을 제거하고 남는 부분만을 gram으로 새롭게 만들어 큐에 남겨 놓는다. 이때 제거된 영역만큼을 일치 문자 수로부터 감소시킨다.

단순 q-gram의 개수를 이용한 방법과 유사한 검증 요건으로 검증 절차를 거쳐 유전자 간의 관계를 판별한다.

이 알고리즘의 시간복잡도도 단순히 q-gram의 개수만을 이용한 알고리즘과 같이 $O(\frac{1}{4^q}n^2)+T_v$ 이다. 다만 일치되는 문자 수를 계산하는데 시간이 추가적으로 요구되나 전체 수행 시간에 미치는 영향은 매우 작다.

그림 4에 q-gram으로부터 일치 되는 문자 수를 실시간으로 추정하는 알고리즘의 의사코드를 제시한다.

4. 실험 결과 및 분석

제시된 알고리즘은 C언어로 구현되었으며 1GB 메모리와 Xeon 2.45GHz에서 실험하였다. 실험은 6343개의 유전자로 구성된 약 8.9 x 10^6 크기의 효모에 대하여 실행하였으며 probe의 크기는 50으로 하고 불일치 허용 임계치는 12로 하였다.

표 1과 2는 네 가지 검증 기준에 대해 6, 7-gram으로 제안한 두 가지 알고리즘의 실제 수행시간과 정확도를 측정한 것이다. 실제 수행시간은 프로그램의 시작부터 종료 시까지의 시간을 나타낸다. 하지만 여기엔 나쁜 유전자들을 찾는 전처리 과정의 시간만 고려되었고 이들을 대상으로 각 유전자들의 probe들을 선정하는데 드는 시간은 제외하였다. 왜냐하면 probe를 선정하는 데는 현재 제안되어 있는 알고리즘이 모두 가능하기 때문이다. 단순기법으로 수행한 결과는 효모의 경우 5분 정도의 시간이 소요되었다. 따라서 probe를 선택하는 전 과정에 요구되는 시간은 전처리 과정에 소요되는 시간에 5분을 더한 시간을 넘지 않는다.

알고리즘의 성능 측정을 위하여 효모에 단순기법을 사용하여 각 유전자들에 대한 나쁜 관계에 있는 유전자들을 모두 구하였다. 실제 유전자 사이에 필요한 비교회수는 10009번으로 각 유전자에 대해서 평균 2번을 넘지 않는다. 이는 6343개의 유전자 사이의 전체 비교회수가 $(6343\times6342)\div2$ 임을 보면 필요한 비교는 훨씬 적다는 것을 알 수 있다. 그리고 정확도는 제안한 알고리즘에 의해 구해진 유전자 수를 단순기법에 의해 구한 유전자수로 나누어 구하였다. 제안한 알고리즘은 언제나 판별을 위해 검증을 거치기 때문에 유전자에 대한 나쁜 관계에 대한 판단은 언제나 옳기 때문이다.

표 1. q-gram의 개수를 이용한 알고리즘의 결과

	6-gram		7-gram	
	정확도 (%)	수행시간 (분)	정확도 (%)	수행시간 (분)
최대 q-gram	82.9	51	-	-
> 5	96.7	37	91.7	14
> 4	97.9	39	94.5	15
> 3	98.8	44	96.4	16

표 2. q-gram으로부터 일치된 문자 수를 계산하는 알고리즘의 결과

	6-gram		7-gram	
	정확도 (%)	수행시간 (분)	정확도 (%)	수행시간(분)
최대 일치 문자 수	84.4	48	83.2	29
> 20	89.2	40	84.4	17
> 15	96.1	40	93.4	17
> 10	98.9	53	97.2	18

표 3. 문자 수를 계산하는 알고리즘을 사용한 경우의 검증 횟수

	6-gram		7-gram	
	전체	평균	전체	평균
> 20	224,364	0.01	91426	0.004
> 15	4,962,923	0.24	2,660,956	0.13
> 10	194,070,297	9.64	38,209,137	1.89

표 4. 기존 알고리즘들과의 비교

Li and Stromo	Rouillard, Herbert, and Zuker	Sung and Lee	제안알고리즘 (97 % 이상)
4 일 (24-mers)	1일 (50-mers)	49분 + α (50-mers)	18 + 5 분 (50-mers)

q-gram에 의한 최대 일치 영역의 추정은 q의 값이 1이 아닌 이상 그림 3의 b)와 같은 경우엔 일치되는 문자 수를 세더라도 완전한 정확성을 보장하지는 못한다. 하지만 불일치 문자들이 크지 않은 q의 값으로 계산하지 못할

정도로 균등하게 분포할 확률은 매우 낮다. 또한 열역학적 관점에서 볼 때 불일치가 균등하게 분포한 probe는 같은 수의 불일치를 가지고 불균등하게 분포한 probe보다 결합반응이 발생할 가능성이 훨씬 낮다[8]. 표1, 2를 보면 6,7-gram의 경우 모두 최대 수의 q-gram을 가진 곳이나 q-gram에 의해 계산된 일치 문자 수가 가장 큰 곳만을 검증한 경우는 90%를 넘지 못한다. 하지만 일정한 크기의 q-gram수나 일치 문자 수를 갖는 영역을 검증하면 100%에 근접한 정확성을 확보할 수 있다. 표3은 문자 수를 계산하는 알고리즘을 사용한 경우의 실제 검증 횟수를 나타내고 있다. 표를 보면 두 유전자를 비교할 때의 검증 영역은 6-gram의 경우 일치 문자 수가 10보다 큰 경우도 평균 10회를 넘지 않고 있다. 표4는 이제까지 발표된 효모의 유전자에 대한 probe 선택 알고리즘의 시간을 비교한 것이다. 그리고 제안하는 알고리즘의 경우는 정확성이 97% 이상인 경우로 한정 하였다. 현재는 효모에 대한 실험만이 진행된 상황이지만 인간과 같은 대규모의 유전자 집합을 가진 생물에 대해서는 시간상의 격차는 보다 커질 것이다.

제시된 실험의 결과들에서 알 수 있듯이 q의 크기나 검증 영역과 같은 파라미터의 선정은 요구되는 정확성과 시간의 상충관계를 적절히 고려하여 이루어져야 한다.

5. 결론

probe의 specificity 판별 문제는 DNA칩의 설계에서 중요한 문제이다. 지금까지 여러 가지 알고리즘들이 제시되었지만 대규모의 유전자 집합을 대상으로 현실성 있는 시간에 이를 판별하는 데는 문제를 가지고 있다. 본 논문에서는 특정 유전자의 후보 probe들을 나쁜 probe로 만드는 유전자들은 많지 않다는 확률적, 실험적 근거를 토대로 개개 probe가 아닌, 유전자 자체를 선별하는 전처리 알고리즘들을 제시하였다. 이 알고리즘은 q-gram을 사용하여 비교되는 두 유전자의 일치되는 문자 수가 가장 큰 영역을 추정하여 검증함으로써 빠른 시간 내에 선별을 가능하게 하였다.

앞으로 q-gram 선정과 이에 따른 손실률의 관계에 보다 정밀한 확률적 분석이 필요하고 대규모의 유전자 집합에 대한 실험도 필요하다. 또한 최대 일치 부분문자열 찾기 문제에 대한 정확성이 보장되는 알고리즘의 개발도 앞으로의 과제이다.

참고문헌

[1] C. Debouck and P. N. Goodfellow. *DNA microarrays in drug discovery and development.* Nature Genetics, 21:48-50, 1999.

[2] L. Gasieniec, C. Y. Li, P. Sant, and P. W. H. Wong. *Efficient Probe Selection in Microarray Design.* IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology. 2006.

[3] L. Kaderali and A. Schliep. *Selecting signature oligonucleotides to identify organisms using DNA arrays.* Bioinformatics, 18(10):1340-1349, 2002.

[4] L. Li and G. Stormo. *Selection of optimal DNA oligos for gene expression analysis.* Bioinformatics, 17(11):1067-1076, 2001.

[5] D. J. Lockhart, H. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, and E. L. Brown. *Expression monitoring by hybridization to high-density oligonucleotide arrays.* Nature Biotechnology, 14:1675-1680, 1996.

[6] U. Manber and G. Myers. *Suffix arrays: a new method for online string serarches.* SIAM Journal of Computing, 22(5):935-948, 1993.

[7] D. W. Mount. *Bioinformatics Sequence and Genome Analysis(2nd ed).* CSHL Press, 2004.

[8] S. Rahmann. *Rapid large-scale oligonucleotide selection for microarrays.* Proc. IEEE Computer Society Bioinformatics Conference, 1:54-63, 2002.

[9] S. Rahmann. *Fast and sensitive probe selection for DNA chips using jumps in matching statistics.* Proc. IEEE Computational Systems Bioinformatics, 2:57-64, 2003.

[10] J. J. SantaLucia, H. T. Allawi, and P. A. Seneviratne. *Improved Nearest-Neighbor Parameters for Predicting DNA Duplex Stability.* Biochemistry, 35:3555-3562, 1996.

[11] W. Sung and W. Lee. *Fast and accurate probe selection algorithm for large genomes.* Proc. IEEE Computational Systems Bioinformatics, 2:65-74, 2003.