다수의 목표 유전자에서 진화연산을 이용한 Oligonucleotide Probe 선택

신기루○, 김선, 장병탁

서울대학교 바이오지능 연구실

{krshin○, skim, btzhang}@bi.snu.ac.kr

Oligonucleotide Probe Selection using Evolutionary Computation in Large Target Genes

Ki-Roo Shin○, Sun Kim, Byung-Tak Zhang

Biointelligence Laboratory, School of Computer Science and Engineering, Seoul National University

요 약

DNA microarray는 분자생물학에서 널리 사용되고 있는 실험 도구로써 크게 cDNA와 oligonucleotide microarray로 나뉘어진다. DNA microarray는 일련의 DNA 서열로 이루어진 probe들의 집합으로 구성되며 알려지지 않은 서열과의 hybridization 과정을 통해 특정 서열을 인식할 수 있게 된다. Oligonucleotide microarray는 cDNA 방법과는 다르게 probe를 구성하는 서열을 제작자가 임의로 구성할 수 있기 때문에 목표 서열이 가지는 고유한 부분만을 probe 서열로 사용함으로써 비용절감과 실험의 정확도를 높일 수 있다는 장점이 있다. 그러나 현재 목표 유전자 서열에 대해 probe 집합을 생성하는 결정적인 방법은 존재하지 않으며, 따라서 넓은 해 공간에서 효과적으로 최적 해를 찾아 주는 진화 연산이 probe 선택을 위한 좋은 대안으로 사용될 수 있다[1,2]. 그러나 진화연산을 이용한 probe 선택방법에 있어서 인식하고자 하는 목표 서열의 개수가 많아질 경우, 해 공간의 크기가 커짐으로 인해 문제점이 발생할 수 있다. 따라서 본 논문에서는 다수의 목표 유전자 서열을 대상으로 한 probe 선택 방법에 있어서 보다 효율적인 진화연산 접근 방법을 소개한다. 제시된 방법은 인식하고자 하는 목표 서열의 일부를 선택해 이를 probe 집합의 후보로 사용하며, 유전 연산자를 이용한 진화과정을 통해 최적에 가까운 probe 집합을 찾는다. 본 논문은 GenBank로부터 유전자 서열을 대상으로 제안된 방법을 실험하였으며, 축소된 목표 서열만을 이용해 probe 집합을 선택하더라도 적합한 probe 집합을 찾을수 있었다.

1. 서 론

DNA microarray는 분자생물학 분야에서 널리 사용되고 있는 실험 분석 도구로써 일련의 DNA 서열로 이루어진 점(spot)들이 유리 표면 위에 놓여진 형태를 가진다. 이때 일련의 DNA 조각들을 probe라고 하며, 서열이 알려지지 않은 유전자와 이 probe와의 결합 과정(hybridization)을 통해 특정 서열을 인식할 수 있게 된다. DNA microarray는 크게 cDNA 칩과 oligonucleotide 칩으로 구별되며 oligonucleotide 방식은 cDNA와 다르게 제작자가 probe의 서열을 디자인할 수 특징을 가진다. 따라서 oligonucleotide microarray의 장점은 제작자가 구별하고자 하는 유전자의 서열 구성을 보고 타 유전자와 중복되거나 유사한 서열이 있을 경우, 이를 피해 특정 유전자만을 인식하는 DNA 서열, 즉 probe를 선택할 수 있다는 것이다. 따라서 oligonucleotide microarray에서 인식하고자 하는 목표 유전자에 대해 이를 구별할 수 있는 특정 probe들을 선택하는 것은 중요한 문제가 된다. 한편 microarray를 제작하는 데 있어서 필요한 probe의 수, 즉 microarray상의 점(spot)의 개수는 임의의 알려지지 않은 서열을 구별하는데 필요한 hybridization 횟수와 같으며, 이는 곧 제작비용과 연결된다. 그래서 가능한 많은 목표 유전자 서열을 구별하면서 소수의 probe로 구성되는 microarray를 구성하는 것이 최적의 방법이라고 생각할 수 있다.

일반적인 probe 선택 방법은 인식하고자 하는 목표 유전자를 선택하고 각 probe의 길이를 선택하는 것에서부터 시작된다. 그 다음 목표 유전자 서열을 대상으로 정해진 probe 길이에 따라 sliding window방식으로 뽑아내어 후보 probe집합이 구성되며, 이렇게 선택된 probe 집합 중에서 목표로 하는 것이 아닌 다른 유전자와의 결합(cross hybridization)이 일어날 가능성 있는 후보들은 제외가 된다. 마지막으로 실험에 영향을 주는 기타 요소들을 고려해 목표 유전자와의 hybridization이 일어날 가능성이 높은 probe가 최종적으로 선택되게 된다[3,4]. 한편, probe 집합 선택을 위한 학문적인 연구도 꾸준히 수행되어 왔다. Herwig et al.[5]은 probe 선택을 최적화 문제로 보고 엔트로피 최대화 문제를 이용한 기법을 제시하였고, Li. et al.[6]은 free energy 및 melting temperature를 기준으로 한 probe 선택 기법을 제시하였다. Bourneman et al.[7]은 probe 집합 선택을 probe의 개수를 고정한 상태에서 가능한 많은 목표 서열을 구분하는 probe 집합을 찾는 문제와 고정된 목표 서열에 대해서 이를 구분할 수 있는 최소 크기의 probe집합을 찾는 문제로 나누어 풀고자 하였으며, 각 문제를 위해 simulated annealing과 Lagrangian relexation을 이용한 동적 프로그램 방법을 제안하였다. 또한 Tobler et al.은 probe 선택을 위해 naive Bayes, 신경망, 결정트리 등을 이용한 사례를 연구하였다[8].

본 논문에서는 oligonucleotide microarray에서의 probe 집합 선택 문제를 다루며, 일정한 개수로 이루어진 probe 집합에 있어서 다량의 목표 유전자를 가능한 많이 구별하도록 하는 probe 집합을 찾는 진화연산 기법을 소개한다. DNA microarray는 그 제작과정에 영향을 주는 물리, 화학적인 요인으로 인해 현재까지 probe 선택을 위한 결정적인 방법이

발견되지 않고 있으며, 큰 해 공간에서 효율적으로 최적해를 찾는 진화연산이 최적의 probe 집합을 찾기 위한 좋은 대안으로 사용될 수 있다[1,2]. 그러나, 진화연산을 이용한 접근 방법에 있어서 인식하고자 하는 목표 서열의 개수가 많아질 경우, 그에 따라 해 공간의 크기가 커짐으로 인해 문제점이 발생할 가능성이 있다. 따라서 본 논문에서는 목표 유전자의 일부를 샘플링해 probe 서열 후보로 사용하여 가능한 해 공간의 크기를 줄이면서도 그 성능을 일정하게 유지하는 기법을 소개하고자 한다. 제시된 방법은 GenBank로부터 얻어진 1158개의 유전자 서열을 대상으로 실험을 하였으며 그 결과를 분석하였다. 2장에서는 probe 선택 문제를 설명하며, 4장에서는 probe 선택을 위해 본 논문에서 제시하는 진화연산 알고리즘을 소개한다. 5장에서는 목표 유전자 서열를 바탕으로 한 실험 및 결과를 분석한다.

2. 문제 정의

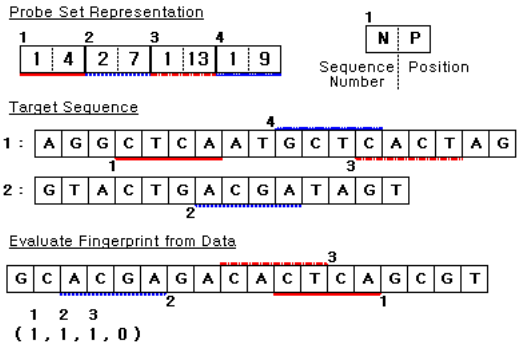

그림 1. probe 집합 표현과 fingerprint 예

Fingerprint는 한 유전자 서열에 대해 probe 집합 P의 각 probe들이 부분 서열로 존재하면 1, 아니면 0으로 표현한 벡터이다. 그리고 서로 다른 유전자 서열이 다른 fingerprint를 가지면 둘은 서로 구별할 수 있게 된다. 그림 1은 probe 집합에 대한 표현 방법과 fingerprint에 대한 예를 보인 것이다. 그림 1의 첫 번째 그림은 probe 집합을 나타낸 것으로써 샘플링된 목표 유전자의 번호와 그 유전자 안에서 probe의 시작위치를 나타내주는 위치값의 쌍으로써 한개의 probe가 표현되어진다. 즉, 길이가 4인 probe라고 가정했을 경우, 그림의 첫 번째 probe (1,4)는 1번 목표 유전자의 4번째 위치를 시작점으로 하는 'CTCA'를 가리키고 있는 것이다. 한편, 그림 1의 아래 그림은 제시한 probe 집합에 대해 fingerprint 결과를 예로 보인 것이다. 데이터 서열안에 4번 probe를 제외한 모든 probe의 DNA 서열과 일치하는 구간이 있으므로 fingerprint 벡터값은 (1,1,1,0)으로 표현됨을 알 수 있다. 참고로 원칙적으로는 DNA 염기의 결합이 A-T, G-C 형태로 제한되지만 그림 1에서는 편의상 각 염기는 같은 염기와 결합하는 것으로 표기하였다.

본 논문은 일정 개수로 이루어진 probe 집합에서 목표 유전자를 가능한 많이 구별하도록 하는 probe 집합을 찾는 문제를 풀고자 하며, 이는 목표 유전자에 대해 각기 다른 fingerprint를 갖게 하는 probe 서열들을 찾는 문제로 다시 정의할 수 있다. 따라서 일정 개수의 probe 집합에 대해 목표 유전자를 모두 구별할 수 있는 최적해가 존재한다면, 그 probe 집합의 서로 다른 fingerprint 벡터는 모두 목표 유전자의 개수만큼 존재하게 된다.

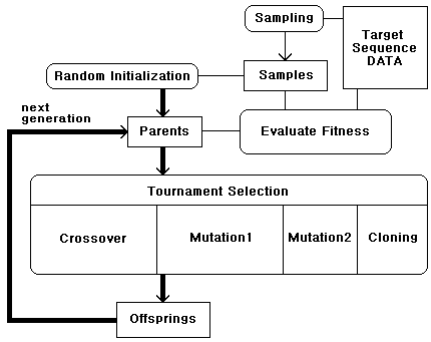

그림 2. 진화 연산을 이용한 probe 선택 시스템

3. 진화 연산을 이용한 Probe 선택 방법

본 논문에서 제시한 probe 선택을 위한 진화 연산 기법 전체적인 구성도를 그림 2에 나타내었다. 먼저 전체 목표 유전자 서열에서 probe 서열로 사용할 일부 유전자를 샘플링한다. 샘플링되는 유전자의 개수는 probe 집합 개수의 2배수로 정하였으며, 이는 목표 유전자에 존재하는 모든 경우의 조합을 probe 후보로 사용하지 않더라도 샘플링을 통해 보다 적은 시간에 최적해를 찾을 수 있는 충분한 조합을 만들어 낼 수 있기 때문이며, 다음 장의 실험을 통해 그 결과를 나타내었다. 샘플링된 목표 유전자 서열을 바탕으로 진화가 이루어지며, 초기해는 샘플링된 서열의 일부로 임의로 선택하는 방법으로 구성된다. 각 개체는 모든 목표 유전자를 대상으로 한 fingerprint 벡터 값을 기준으로 그 적합도가 평가된다. 부모해는 tournament selection에 선택되며 이를 바탕으로 1-포인트 교차(40%), 변이 1(40%), 변이 2(10%) 및 복제(10%)를 적용하여 다음 세대의 해집합을 생성한다(그림 3).

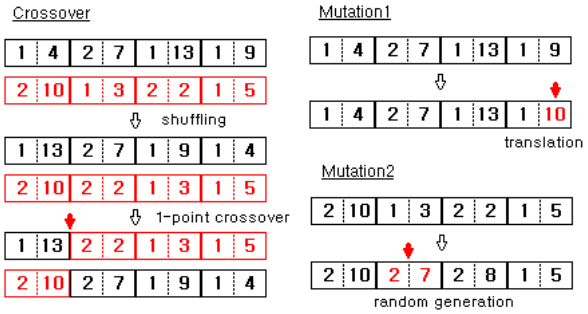

그림 3. GA Operator

4. 실험 결과

실험에 사용한 데이터는 GenBank(NCBI)로부터 얻은 1158개 유전자 서열이다. Probe의 길이는 8, population size는 500이며, 진화 세대수는 probe 개수에 따라 점점 증가시켜 수행하였으며 실험을 통해 적절한 수렴 시기를 측정하여 정하였다. 예를 들어, 14개의 probe에 대해서는 50세대, 22개의 probe에 대해서는 100세대까지 진화를 시켰다.

표 1은 probe 후보 유전자의 샘플링 크기에 따른 성능 변화를 정리한 것이다. 제한적으로 14개의 probe 집합에 대하여 3, 7, 14, 28, 42개 유전자를 샘플링하여 각각 10번씩 실험하였으며 그 평균과 표준 편차를 구하였다.

개수	평균	표준편차
3	845.9	108.89
7	868.1	115.29
14	877.7	110.61
28	873.8	83.56
42	880.7	80.4

표 1. 샘플링 개수에 따른 성능 변화

샘플링 크기가 커질수록 점차 성능이 좋아지다가 샘플링한 유전자 개수가 14 이상이 되면 평균 적합도가 점점 수렴하는 것을 알 수 있다. 특히 최고 적합도의 경우, 그 변화가 거의 없음을 확인할 수 있었다. 샘플링 개수가 7 이하인 경우에도 좋은 결과를 얻은 경우가 있었지만 안정적으로 찾지는 못하였으며, 작은 sampling일수록 편차가 급격하게 심해졌다. 또한 샘플링 크기에 대해 수렴 속도가 기하급수적으로 늘어나는 경향을 보였는데 이는 샘플링된 유전자의 크기가 증가함에 따라 가능한 probe 서열 후보의 집합, 즉 해공간이 늘어나기 때문에 발생하는 문제이다.

샘플링 되는 유전자의 수를 너무 작게 하면 일부 좋은 결과를 얻는 경우가 있음에도 불구하고 수렴하지 않는 경우가 많기 때문에 평균적으로 좋은 결과를 기대하기는 힘들다. 반대로 샘플링 크기를 너무 크게 하면 계산 시간이 증가하고 수렴하는 속도가 느려지기 때문에 비교적 빠른 시간내에 적절한 해를 찾기 어려워지게 된다. 따라서 실험에 사용한 1148개의 목표 유전자에 대해 샘플링 개수는 선택하고자 하는 probe 개수의 2배수 이하에서 결정하였으며, 그림 4 및 그림 5에 2배수 샘플링 통한 실험 결과를 나타내었다. 제시한 알고리즘의 적합도는 클러스터 관점에서 볼수 있고 이는 그림 4에 나타낸 결과와 같다. Probe 개수가 증가함에 따라 구별가능한 유전자의 개수가 늘어남을 알 수 있고, 20개의 probe에 대해서는 목표 유전자의 개수에 상당히 근접함을 볼 수 있다. 그림 5는 본 실험과 같은 데이터를 사용한 Borneman et al.[7]의 결과와 비교한 것으로써 비슷한 성능을 보인다는 것을 알 수 있다.

5. 결 론

본 논문에서는 oligonucleotide microarray에서의 probe 선택 문제에 있어서 목표 유전자의 샘플링을 통한 진화연산을 적용함으로써 목표 유전자의 크기가 증가함에 따라 발생되는 문제점을 해결하고자 하였다. 후보 probe 서열을 생성하기 위한 샘플링 방법은 해 공간의 크기를 제한함으로써 상대적으로 빠른 시간안에 좋은 성능의 probe 집합을 선택하는데 도움을 주었다.

한편, 실제 DNA microarray에서 사용하는 probe의 길이는 20 이상인데 반해 본 논문의 실험에 사용한 probe의 길이는 8로써 실제 microarray 제작과는 아직 거리가 있다. Probe의 길이가 20 이상이 되는 경우 DNA 서열로 만들어지는 가능한 probe 조합의 수가 기하급수적으로 증가하기 때문에 본 논문에서 제시한 샘플링 방법에 의한 효과가 감소할 수 있다. 따라서 향후 probe의 길이가 증가함에 따른 성능 변화를 관찰하고 발생할 수 있는 문제점을 개선하기 위한 연구가 필요하다.

감사의 글

본 연구는 교육부 BK21-IT 사업, 국가지정연구실(NRL) 사업 및 산업자원부 차세대신기술사업에 의하여 일부 지원되었음.

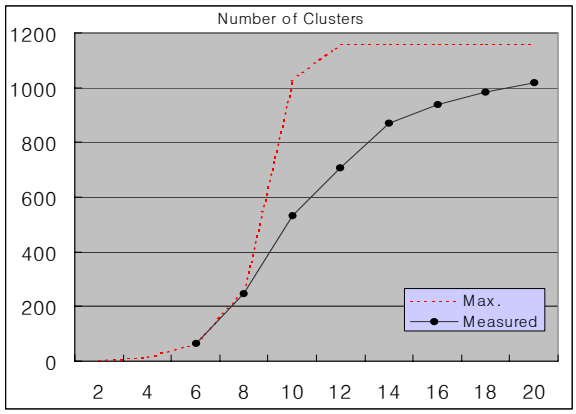

그림 4. probe 개수에 따른 cluster 개수

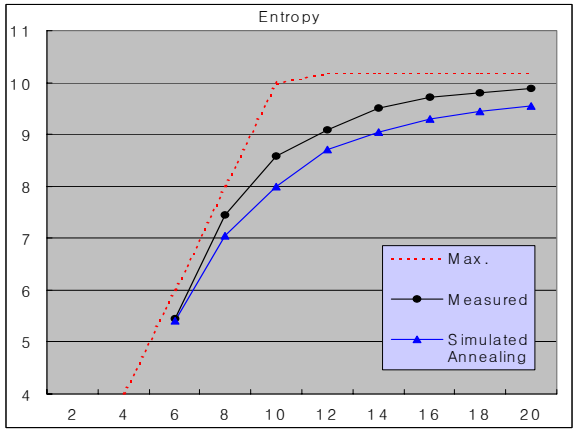

그림 5. probe 개수에 따른 entropy

참고 문헌

[1] 김선, 장병탁, 유전 알고리즘을 이용한 DNA Microarray의 Probe 선택, *한국퍼지 및 지능시스템학회 춘계 학술발표논문집*, pp. 183-186, 2002.

[2] 김선, 장병탁, Oligonucleotide Microarray의 Probe 선택을 위한 진화적인 접근 방법, *한국데이터마이닝학회 추계 학술대회 논문집*, pp. 140-147, 2002.

[3] http://www.olympus.co.jp/Special/specialE.html

[4] Tolstrup, N., et al., Optimal Design of Oligos for Micro Array Gene Expression Profiling, *The 10th International Conference on Intelligent Systems for Molecular Biology*, Poster, 2002.

[5] Herwig, R., et al., Information Theoretical Probe Selection for Hybridisation Experiments, *Bioinformatics*, 10, pp. 890-898, 2000.

[6] Li. F. and Stormo, G. D., Selection of Optimal DNA Oligos for Gene Expression Arrays, *Bioinformatics*, 17, pp. 1067-1076, 2001.

[7] Borneman, J., et al., Probe Selection Algorithms with Applications in the Analysis of Microbial Communities, *Proceedings of the 9th International Conference on Intelligent Systems for Molecular Biology*, pp. 39-48, 2001.

[8] Tobler, J. B., et al., Evaluating Machine Learning Approaches for Aiding Probe Selection for Gene-Expression Arrays, *Proceedings of the 10th International Conference on Intelligent Systems for Molecular Biology*, pp. 164-171, 2002.