In Vitro 조절 기전 모델링을 위한 DNA 컴퓨팅

남진우[O1] 정제균[1] 장병탁[1,2]
[1]서울대학교 생물정보학 협동과정
[2]서울대학교 컴퓨터공학부
{jwnam[O], jgjoung, btzhang}@bi.snu.ac.kr

DNA Computing for *In Vitro* Regulatory Machinery Modeling

Jin-Wu Nam[01] Je-Gun Joung[1] Byoung-Tak Zhang[1,2]
[1]Program in Bioinformatics, Seoul National University
[2]School of Computer Science and Engineering, Seoul National University

요 약

바이오네트웍 모델링은 유전자네트웍, 단백질네트웍, 대사회로, 신호전달회로네트웍등에 대하여 각 요소간의 관계를 그래프이론을 통하여 표현하는 작업을 말한다. 특히 조절네트웍의 모델링은 다양한 생물학적 실험 데이터로부터 단백질들간의 활성과 불활성 관계를 유추해내는 것을 말한다. 현재 조절네트웍 모델링을 위한 다양한 알고리즘들이 개발되어 있으나 응용적인 측면에서 유추된 네트웍은 활용성이 부족하다. 본 논문에서는 *In Vitro*상에서 DNA 컴퓨팅을 이용하여 간단한 연산을 수행함으로서 유전자 조절 기전을 모델링하고자 한다. 이러한 방법의 장점은 DNA컴퓨팅의 연산이 세포의 현재 또는 다음 상태를 *In Vivo* 상에서 구현되어 진단 등의 문제에 응용될 수 있다는 가능성을 제시해 준다는 것이다.

1. 서 론

살아있는 세포들은 수천에서 수만 개의 유전자를 포함하고 있고, 각각의 유전자는 한 개 혹은 그 이상의 단백질 생성에 관여하고 있다. 수많은 단백질들은 환경이나 조직의 성장에 따라 복잡한 경로(pathway)를 형성하는 조절 기작들에 관여한다. 이러한 조절 기작들을 이해하기 위해서는 수학적 모델링 또는 회로적 분석 방법이 필요하다.

현재 마이크로어레이(microarray)등의 출현은 정상 또는 다양한 처리(treatments)를 하거나 섭동(perturbation)에 의해서 동시에 수많은 유전자에 대하여 발현 양상의 측정이 가능하게 하였다. 유전자간의 조절 네트웍은 이러한 발현 패턴들의 실험결과를 통하여 구축될 수 있는데, 이러한 네트웍을 '유전자 네트웍' 이라고 한다. 네트웍의 유추는 기존에 연구된 Boolean network, Differential equation, Bayesian network (BN), Dynamic Bayesian network (DBN)등의 알고리즘들이 사용될 수 있다[1][2]. 이중에서 Boolean network은 가장 간단한 형태의 모델로서 유전자가 발현하여 단백질이 활성화되는지 또는 불활성화 되는지를 이진 코드로 표현하게 된다. 그 밖의 모델들은 더욱 복잡한 계산에 의하여 유전자간의 상호작용 관계를 추론하게 된다.

이러한 알고리즘들을 이용하여 생성된 유전자간의 상호관계는 정보의 활용 측면에서 실제 다양한 문제에 적용될 필요가 있다. 분자컴퓨팅[3]은 *In Vitro* 또는 *In Vivo*상에서 직접 연산을 수행할 수 있다는 장점을 가지고 있다. 따라서 여러 가지 연산의 수행은 특정 질병의 판별이나 예후를 유추해 낼 수 있다. 본 논문에서는 유전자간의 상호관계에 대한 데이터들을 근거로 DNA 컴퓨팅을 수행하여 세포상의 상태 변화를 감지할 수 있는 방법을 소개하고, 시뮬레이션을 통하여 가능성을 검증하기로 한다. DNA 컴퓨팅을 이용한 세포의 조절 상태 예측은 진단 등의 문제에 직접적으로 응용될 수 있는 가능성을 보여줄 것이다.

2. 조절네트웍의 기본구조

생물학적 네트웍의 모델링을 위한 가장 간단한 모델은 불리안 네트웍(Boolean Networks)이다. 간단한 예를 위하여 단백질-단백질 상호작용을 구성하는 조절네트웍을 가정해보자. 시스템은 특정 t시간에 단백질들을 대표하는 N개의 g_1, g_2, ..., g_N에 대하여 ON일때 g_i =1또는 OFF일때 g_i =0의 값을 가지는 이진 요소들이 서로 연결되어진 형태로 요약될 수 있다.

예를 들어, 그 요소들에 대하여 ON 값에 대해서는 kinase가 충분한 양으로 발현되어서 어느 정도 임계치(threshold)를 넘었을 때 활성화 상태를 나타내는 것을 의미하고, 반면에 ON 값에 대해서는 발현이 되지 않아서 임계치를 못 넘었을 때 불활성화 상태를 나타낸 것을 의미한다.

전체 네트웍관점에서 보면 네트웍 구조는 단백질들의 상호작용들(연결선)과 네트웍 요소에 할당된 지역함수들의 위상(topology)을 구성한다. 예를 들어, 네트웍 요소에 할당된 지역함수에는 OR, AND, NOT IF등이 있을 수 있다. 시간이 지남에 따라 특정 t시간의 네트웍 요소들 $g_i(t)$의 활성화 상태는 네트웍 상태 $S(t)$ = [$g_1(t)$, $g_2(t)$, ... , $g_N(t)$] 패턴으로 변하게 된다. 이러한 동적인 네트웍의 특성은 세포의 상태를 표현할 수 있다.

3. DNA 컴퓨팅을 이용한 유전자 조절 기전의 상태 예측

DNA 컴퓨팅을 이용한 유전자 조절 기전의 상태 예측을 위해서는 각 노드와 노드에 해당하는 연산자의 DNA 염기서열 디자인이 필수적이다. 우선 초기 상태에 해당하는 $S(0)$ = [$g_1(0)$, $g_2(0)$, ..., $g_N(0)$] 이진값이 들어오면 다음 상태인 $S(1)$ = [$g_1(1)$, $g_2(1)$, ... , $g_N(1)$]로 변하도록 하는 연산자의 디자인이 필수이다. 예를 들어, A, B, C, D의 단백질이 있을때 t+1 상태는 t 상태의 A, B, C, D 이진값에 의해 결정되는데, A는 C 와 D의 OR 연산자에 의해 다음상태가 정해지며, B는 B자신과 A의 OR 연산자에 의해 다음상태가 정해지고, C는 A와 B의 AND 연산자, D는 D 와 C의 NOT IF 연산자에 의해 다음상태가 정해진다.

DNA 서열 디자인에서 출력의 결과는 연산자의 겔 밴드(Gel

band) 패턴에 따라 결정되는데, 작은 두개의 밴드가 보이게 되면 False이고, 큰 한 개의 밴드가 보이게 되면 True를 출력하게 하여 쉽게 결과를 확인할 수 있도록 했다. 다음 상태의 전이는 1회 반응 후 결과에 따라 다음 이진 연산자에 해당하는 입력을 넣어주게 된다. AND 연산자는 반드시 A와 B가 1일 때만 true가 되도록 하기 위해 A, B 서열이 Sticky end를 만들어 AND연산자와 hybridization을 하게 되면 환형 형태의 DNA 구조가 만들어지게 했다. OR 연산자는 두개의 노드중 한 개만 들어와도 OR 연산자와 hybridization을 하여 환형 형태의 DNA 구조를 만들게 되며, NOT IF연산자는 D 노드만 들어왔을 때 NOT IF 연산자와 hybridization을 하여 환형 형태의 DNA구조를 만들게 했다. 이렇게 환형 형태의 구조에는 제한효소 사이트가 특이적으로 한 개씩 있게 디자인하여, 제한효소를 처리했을 때 환형 형태는 1개의 선형구조가, 환형(circle)을 이루지 못한 구조는 2개의 짧은 선형구조가 생산되도록 디자인하였다 (그림 1).

4. 시뮬레이션을 위한 디자인

4.1 DNA Sequence Design 예

A, B, C ,D 노드는 염기서열로 디자인하며 A, B 와 C, D의 염기서열 끝부분에는 서로 일부분 hybridziation할 수 있는 상보염기서열이 디자인 되어있다. 이 두개의 염기서열이 동시에 입력으로 들어오면 sticky end인 hybridzation 구조를 만들어, 연산자와 다시 hybridization을 동시에 하게 되어 환형 형태의 DNA 구조를 만들게 된다. 또한 OR 연산자에는 연산자 외에 보조 연산자 염기서열이 존재하여 두개의 노드중 한 개의 노드가 들어와도 환형 형태의 구조를 만들 수 있도록 도와준다.

A : 5’ AGGTCA　　C : 5’ GGCTAT
B : 5’ AGGTTG　　D : 5’ GGCTTA

위의 서열들은 A, B, C ,D의 각각의 노드에 대하여 디자인된 올리고들을 나타내고 있다.

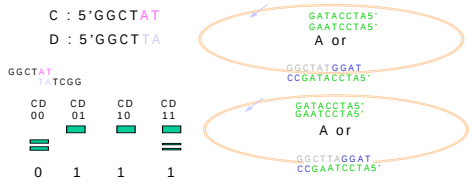

그림 1 유전자 조절 상태를 볼 수 있는 연산자와 노드의 디자인. (A OR 연산자)

A 노드는 우선 전 상태 C, D의 결과에 의해 영향을 받으며, C 나 D 둘 중 어느 것이 입력으로 들어오더라도 A의 다음 상태는 true로 결정된다. 그러므로 우리는 C 또는 D가 들어오더라도 연산자와 hybridization ligation을 통해 환형 형태를 만들도록 하기 위해, 보조 서열(auxiliary sequence)을 추가로 넣어, C 또는 D의 올리고와 sticky end를 만든 후 연산자와 hybridizaiton, ligation을 이루도록 하였다. 그러므로 제한 효소를 처리하면 백터 형태의 연산자에 단일(single) 제한효소 사이트에 작용하여, C 또는 D의 올리고가 입력되었다면, 큰 밴드(band) 한 개 보일 것이며, 아니라면 작은 것 두개가 보이게 될 것이다.

C 노드의 경우는 전 상태의 A, B에 의해 결정이 되며, A, B 모두가 들어 와야만 true가 되는 AND 연산자에 의해서 영향을 받는다. A, B의 올리고는 동시에 존재할 때 서로 상보적인 (complement) 서열 부분을 가지고 있어서 동시에 sticky end를 만들어 연산자와 hybridization ligation을 한 후 환형 형태로 만들어진다. 밴드의 형태 또한 A, B 입력이 모두 들어왔을 때만 큰 크기의 밴드가 나타난다.

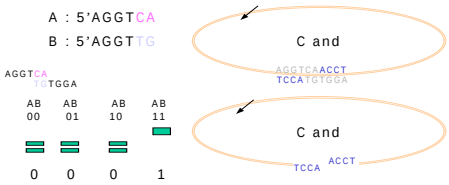

그림 2 유전자 조절 상태를 볼 수 있는 연산자와 노드의 디자인. (C AND 연산자)

4. 2 전체적인 실험과정

우선 한 개의 튜브 안에 A, B, C, D 연산자와 OR 연산자를 위한 보조서열을 동시에 넣은 후 초기상태 $S(0) = [g_1(0), g_2(0), , g_M(0)]$의 상태 값에 해당하는 서열 A, B, C, D를 튜브 안에 넣어준다. 즉 상태가 1(true)이면 그 상태에 해당하는 서열을 튜브에 넣어주고, 만약 state가 0(false)이면 넣어주지 않는다. 이렇게 hybridization에 참여한 B, C의 서열과 연산자 서열간의 랜덤 반응(random interaction)과 hydrogen bond interaction에 의한 specific hybridization을 이루게 된다. 우리는 이 결과를 검출(detection) 하기 위해, 제한효소 A', B', C', D'을 처리한 후 나온 서열 product를 gel eletrophoresis과정을 거치면 밴드 크기와 개수에 따라서 결과를 알 수 있다. 우선 큰 크기의 밴드 하나만 보인다면, 환형 형태의 결과에서 나온 것이므로 true를 의미하는 것이며, 반대로 작은 밴드 두개가 보인다면 선형 형태의 결과에서 나온 것이므로 false를 의미하게 된다. 이렇게 나온 결과를 검출한 후 그 결과에 맞게 다음 상태를 결정하기 위한 연산을 다시 진행시키게 된다. 이때 모든 material들은 다시 충전하게 된다. 이 실험의 전과정은 Lap On a Chip 형태로 구현될 수 있으며, 더 나아가서는 *in-vivo* 상태에서 구현할 수 있는 가능성이 있다.

5. 시뮬레이션

5.1 시뮬레이션 설정

컴퓨터 시뮬레이션은 Cell Cycle중에서 APOPTOSIS과정을 모델링하였다. 이러한 Cell Cycle에 관련된 상태 공간 및 attractor는 Wuensche에 의해 제한된바 있다[4]. 실제 모든 가능한 상태 공간은 조합의 급증으로 인한 엄청난 크기를 가지고 있다. 하지만 모든 상태들이 똑같이 안정화되어 있는 것은 아니고 네트웍 상태들이 지역적으로 제한되어 있고 불안정하다. 따라서 불리안 함수가 수행될 때 주로 이웃하는 상태에만 영향을 준다.

단백질의 상태들은 불안정한 상태의 체인을 따라 바뀌기 때문에 각 상태들은 계속해서 업데이트 된다. 이러한 현상을 trajectory라고 정의하고 있다. 상태 공간은 유한하기 때문에 결국에는 수렴하게 되고 루프형태로 나타날 수 있다. 따라서 $S(t+1) = S(t)$로 같은 상태를 유지하게 될 것이다. 이러한 상태에서는 trajectory를 끌어당기기 때문에 이를 attractor라고 부른다. 구체적으로는 APOPTOSIS과정 중에서 상태 [0,1,0,1] -> [1,1,0,1] -> [1,1,1,1] -> [1,1,1,0]을 모델링하기로 한

다.

컴퓨터 시뮬레이션을 위한 파라미터로서 상태공간에 참여하는 단백질 수는 4개, 올리고 종류는 총 12개, 반응횟수는 10^7, 올리고 개수는 각 올리고당 10^7, 올리고 총수는 12 X 10^7로 설정했다. 사뮬레이션의 흐름은 먼저 올리고들을 튜브에 넣고 올리고 군(oligo population)이 만들어 졌을 때 두 올리고들이 랜덤하게 선택되어서 Hybridization하는지 그렇지 않은지를 판단한 후, 튜브에 있는 올리고 양을 조절하게 되는 과정을 나타낸다. 여기서 랜덤하게 올리고들을 선택할 때 시간에 따른 올리고 양의 분포를 고려하였다. 시간에 따른 각각의 종류에 대한 올리고 양의 변화는 이러한 시뮬레이션을 통하여 모두 다르게 나타날 것이다.

5.2 시뮬레이션 결과

그림 3은 APOPTOSIS과정에 대한 컴퓨터 시뮬레이션 결과를 보여주고 있다. 전체적으로 상태는 4단계의 전이를 거치고 있다. 각 그래프에서 X축은 올리고의 종류(총 12개)를 나타내고, Y축은 반응 횟수를 나타내고 있으며, Z축은 hybridization되어 소모되는 올리고 양의 개수를 나타내고 있다. 각각의 상태들에 대한 소모되는 올리고 양의 변화들이 전체적으로 유사한 결과를 나타내고 있다. OR 불리안 함수는 비교적 많이 사용되는 반면에 AND나 NOT IF 불리안 함수는 비교적 사용되지 않고 있다. 이러한 현상은 당연한 결과로써 AND나 NOT IF로 인하여 ON이 될 확률은 1/4이기 때문에 hybridization될 확률이 그만큼 작다고 할 수 있다.

그림 4는 각 상태에서 다음 상태로 전이되는 과정에서 hybridization된 양이 어떻게 변화하는가를 보여주고 있다. 각 상태의 초기에는 급격하게 반응이 일어나서 hybridization되는 양이 많다가 점점 줄어드는 추세를 보이고 있다. 이는 일반적으로 다른 생물학적 반응에 대한 시뮬레이션에서도 보여지는 현상이다. 연산에 따라 다음 상태가 결정되는데 그림에서 보여지는 바와 같이 각 상태에서 A, B, C, D 각각의 단백질의 량은 hybridization되는 양과 동일하다. 예를 들어, 0101->1101로 가는 상태 전이는 A 단백질이 약 5.7 X 10^4 개가 hybridization에 의해서 활성화되고, B 단백질은 약 6.5 X 10^4 개, D 단백질은 약 2.5 X 10^4개가 활성화된다.

주로 C단백질과 D단백질의 활성화 양은 비교적 A와 B단백질에 비해 적은데 이는 각 상태의 조건 또는 연산자에 영향을 받기 때문이다. 예를 들어, C와 D가 활성화되기 위해서는 AND와 NOT IF 연산자가 사용되는데 이는 OR 연산자에 비해서 반응이 일어나기 어렵다. 실제 실험상에서는 연산자에 의한 양의 차이는 무시될 수도 있다. 상태도 자체가 불리안이기 때문에 어느 정도의 양으로 hybridization되면 ON으로 설정되게 된다. 생체 내에서도 마찬가지로 시간이 지남에 따라서 유전자 발현양의 정도에 따라 활성화되거나 불활성화되는데 그 양에 있어서 어느 정도의 threshold 이상을 넘어야 활성화가 된다고 할 수 있다. 이런 점에서 생체내에서 일어나는 조절 기작의 모델링이 유사하다고 할 수 있다.

6. 결 론

본 논문에서는 유전자간의 상호관계에 대한 데이터들을 근거로한 DNA 컴퓨팅을 수행하여 세포상의 상태 변화를 감지할 수 있는 방법을 소개하고, 시뮬레이션을 통하여 가능성을 검증하였다. 제안된 방법이 보여주고 있는 가장 큰 장점들 중에 하나는 세포의 상태를 알 수 있게 하는 조절기전 모델링을 *in vitro* DNA 컴퓨팅으로 구현하면 실험 생물학적 응용을 위한 여러 가지 대안이 마련될 수 있다는 것이다. 또한 *in silico* 상에는 데이터 변형(transformation)이 일어나서 정보의 손실이 발생할 수 있지만 제안된 방법은 이러한 정보의 손실이 없이 원하는 결과를 얻을 수 있다.

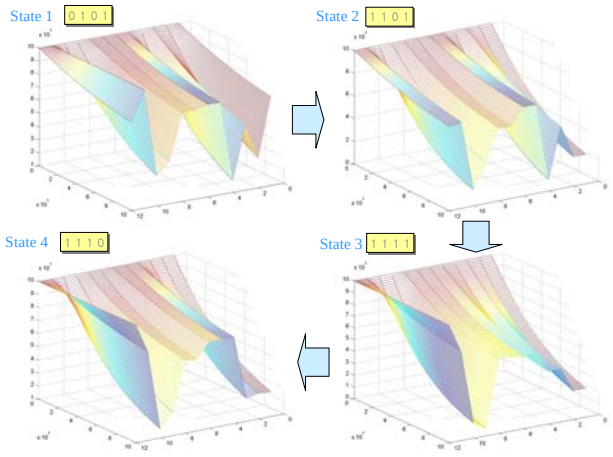

그림 3 APOPTOSIS 상태도의 시뮬레이션 결과

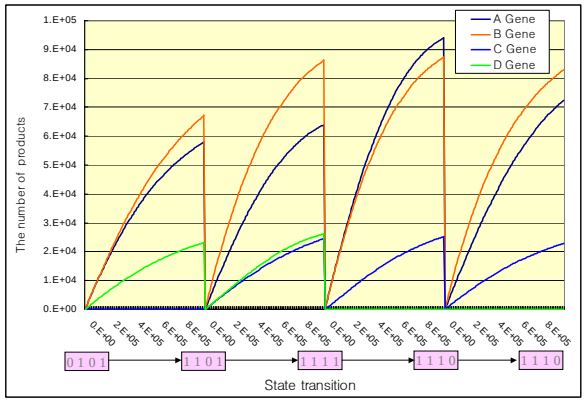

그림 4 각 State에 대한 hybridization된 양의 변화량 측정 결과

시간에 따른 유전자들의 상태를 토대로 유전자 네트웍 유추하는 방법 중에서 불리안 네트웍 모델은 가장 기본이 되는 접근법이다. 본 논문에서는 상태 전이의 예측 방법을 주로 다루었는데 반대로 유전자 네트웍을 유추하는 방법이 DNA 컴퓨팅 실험으로 가능함을 입증하게 된다면 좀더 복잡한 유전자 네트웍을 DNA 컴퓨팅으로 유추 방법에 핵심적인 기술을 제공할 것이다.

감사의 글

이 논문은 과학기술부의 국가지정연구실 사업과 IMT-2000 과제에 의하여 지원되었음.

참고문헌

1. Somogyi, R., and Sniegoski, C. A., Modelling the complexity of genetic networks: Understanding multigenic and pleiotropic regulation, *Complexity*, Vol.1, pp. 45-63.
2. Chen, T.., He, H. L. and Church, G. M, Modeling gene expression with differential equations, *Proc. Pac. Symp. Biocomput.*, pp. 17-28, 1999.
3. Adleman, L. M., Molecular computation of solutions to combinatorial problems, *Science* 266, pp. 1021-1024, 1994.
4. Wuenshe, A., Genomic regulation modeled as a network with basins of attraction, *Pac. Symp. Biocomput*, Vol 3, pp. 89-102, 1998.