Peptide Nucleic Acid(PNA)를 이용한 antisense 기법에 적용할 병렬 컴퓨팅용 Bioinformatics tool 개발

김성조[1,O], 전호상[2], 홍승표[3], 김현창[3], 김한집[3], 민철기[4,*]
[1]전자공학부, [2]정보 및 컴퓨터공학부, [3]생명과학과, [4]분자과학기술학과, 아주대학교
{rlat[O], jhsbeat, sp1020, genetics, hjkim}@ajou.ac.kr
*Corresponding author: 민철기, minc@ajou.ac.kr

Developing a Bioinformatics Tool for Peptide Nucleic Acid (PNA) antisense Technique Utilizing Parallel Computing System

Seongjo Kim[1,O], Hosang Jeon[2], Seungpyo Hong[3], Hyon Chang Kim[3], Han Jip Kim[3], Churl K. Min[4,*]
[1]Division of Electronic Engineering, [2]Division of Information & Computer Engineering, [3]Department of Biological Sciences, [4]Department of Molecular Science & Technology, Ajou University, Suwon 443-749, Korea
*Corresponding author: Churl K. Min, minc@ajou.ac.kr

Unlike RNA interference, whose usage is limited to eukaryotic cells, Peptide Nucleic Acid (PNA) technique is applicable to both eukaryotic and prokaryotic cells. PNA has been proven to be an effective agent for blocking gene expressions and has several advantages over other antisense techniques. Here we developed a parallel computing software that provides the ideal sequences to design PNA oligos to prevent any off-target effects. We applied a new approach in our location-finding algorithm that finds a target gene from the whole genome sequence. Message Passing Interface (MPI) was used to perform parallel computing in order to reduce the calculation time. The software will help biologists design more accurate and effective antisense PNA by minimizing the chance of off-target effects.

1.서 론

PNA는 원핵생물에는 응용이 가능하지 않은 RNA interference(RNAi)와는 달리 진핵생물 뿐만 아니라 원핵생물에서도 antisense 효과를 낼 수 있는 DNA 모방 물질이다[1]. 문제는 PNA 역시 표적 외의 유전자와 작용하는 off-target 효과를 초래할 수 있다는 것이다. 이미 효과적인 RNAi 실험을 수행하기 위해서 RNAi의 off-target 효과를 방지하는 여러 알고리즘 및 프로그램[2] 개발 관련 연구들은 많이 진행이 되었지만, 아직 PNA의 off-target 효과를 최소화하고 PNA를 이용하여 실험하고자 하는 생물학자들에게 최상의 서열 정보를 제공하는 Bioinformatics tool이 없는 것이 현실이다. 본 연구에서는 PNA와 DNA상의 상호관계를 바탕으로 알고리즘을 설계하고 off-target 효과를 방지하기 위해 최상의 서열정보를 제공하는 병렬 컴퓨팅용 프로그램을 개발하였다. 프로그램은 크게 두 가지 구성요소를 가지고 있다.

먼저 표적유전자 서열을 전체 게놈 서열로부터 분리해 내는 작업을 위해 전체 게놈 서열 중 표적 유전자서열의 위치를 찾는 효과적인 방법을 고안하였다. 실제로 유사 부분 서열을 찾는 경우, 범용적으로 Basic Local Alignment Search Tool(BLAST)[4,5,6]이 널리 알려져 있고, 뛰어난 성능을 보인다. 본 연구에서는 BLAST 보다 간단한 알고리즘을 이용하여 프로그램을 개발하였다. 알고리즘이 단순하기 때문에 유사부분을 찾는 다른 프로그램 개발에 쉽게 적용이 가능하다. 또한 유사도를 결정하는 기준에 있어서 여러 가지 고려해야 할 파라미터의 값들을 사용하는 유전서열 환경에 맞추어 자유롭게 구현이 가능하다.

두 번째 구성요소는 off-target 효과를 최소화하는 후보자 서열의 모색이다. 그리고 PNA와 DNA의 특이적인 상호관계를 반영하기 위해 통계학적으로 분석한 모델[3]을 적용하였다. 더 나아가 Message Passing Interface (MPI) 라이브러리를 사용하여 프로그램이 병렬 시스템에서 구동 될 수 있도록 하였다. 이로써 생물학자들에게 빠른 시간 내에 off-target 효과를 최소화하는 이상적인 PNA 서열정보를 제공하고자 하였다.

2. 프로그램 구현

2.1 전체 게놈서열로부터 표적 유전자의 분리

Alignment는 두 가지 이상의 서열에서 상동성이 높은 부분을 찾는 목적으로 사용된다. 이중 길이가 서로 다른 서열에서 가장 상동성이 높은 subsequence의 위치를 찾는 것을 local alignment라 한다. Local alignment의 결과를 antisense technique에 필요한 데이터로 활용하게 된다.

1970년 Needleman & Wunsch[7]의 Dynamic-programming 기법에 의한 Pairwise Alignment algorithm이 발표되고 Smith & Waterman[8]이 일반적인 길이의 gap에 대해 이 algorithm을 확장함으로써 서열간의 alignment를 위한 실용 가능한 프로그램들이 구현되었다.

Smith-Waterman 알고리즘의 storage complexity는 사용되는 서열의 길이가 m, n이라 할 때 O(mn) 크기의 메모리를 소요하게 된다. 대용량의 유전자 데이터가 많은 최근에는 m의 값이 Megabyte 단위를 넘는 예를 쉽게 볼 수 있다. m의 값이 Megabyte 단위를 넘어설 경우 O(mn)의 메모리 크기는 일반적으로 Gigabyte 단위를 넘게 되면서 물리적인 메모리의 크기 및 수행 시간에 있어서 문제들을 갖게 된다.

본 연구에서는 기존 Smith-Waterman의 알고리즘을 개선하였다. 전체 게놈서열의 길이를 m, 원하는 표적 유전자 서열의 길이를 n이라 가정하였다. 게놈서열상에서 표적 유전자의 정확한 위치를 찾는 것이 과제이다. 이 때 m을 n의 길이만큼 분할한 후에 순서대로 2n의 길이만큼의 구간을 선택했을 때 해당 표적 유전자 서열은 2n 길이의 구간에 항상 포함이 된다.

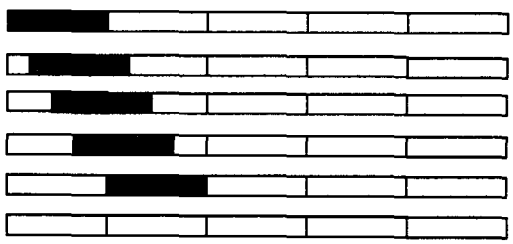

그림 1. 길이 m의 전체 게놈 속에 있는 길이 n의 표적 유전자의 배치 사례들

길이가 n인 표적 유전자의 서열이 전체 분할된 게놈서열상 존재할 경우의 수는 그림 1의 5가지이다. 이로써 게놈 서열상에서 표적유전자의 서열의 위치를 찾는 과정마다 $O(n^2)$ 의 storage complexity를 갖게 된다. 표적 유전자의 위치를 결정하기 위한 유사도의 하한선을 cut-off value로 표현한다. 또한 계산 시간을 단축하기 위해서 추가적인 파라미터가 설정 되었다. Dynamic programming의 배열의 값을 채우는 과정에서 기준이상으로 반복되는 mismatch는 해당 위치에서 표적유전자의 존재가능성이 희박하므로 해당 표적 후보의 연산을 종료하였다.

2.2 타겟 유전자로부터의 이상적인 타겟 스크리닝과 PNA와 DNA의 상호관계 반영

표적 유전자 서열의 길이를 m이라 하고 얻고자 하는 PNA 서열의 길이를 n 이라 가정하면 가능한 후보자의 개수는 m-n+1 이다. 각각의 PNA 후보의 off-target 효과를 분석하기 위해서는 게놈서열 중 표적 유전자의 서열은 그림 2와 같이 계산과정에서 제외시켜야 한다.

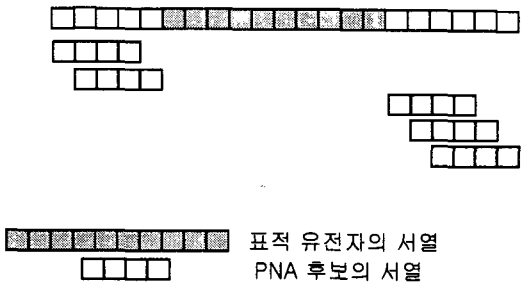

그림 2. n=4라고 가정했을 때 계산 과정의 도식화

PNA와 DNA의 상호작용은 [3]에서 이용한 모델을 사용하였다. PNA와 DNA반응에서의 평형상수는 식(1)과 같이 나타낼 수 있다.

$$P + T_{IJ} \overset{K_{IJ}}{\leftrightarrow} P \cdot T_{IJ} \qquad K_{IJ} = k_0^n f_1^I f_2^J = \frac{[P \cdot T_{IJ}]}{[P][T_{IJ}]}$$

$$k_0 = 8.6,\ f_1 = 10^{-2},\ f_2 = 10^{-3}\ [9] \qquad (1)$$

$$[P]_{total} = [P]_{free} + \sum_{I,J=0,0} \frac{K_{IJ}[P]_{free}[T_{IJ}]_{total}}{1+K_{IJ}[P]_{free}} + \sum_{I,J=0,0} \frac{K_{IJ}[P]_{free}[D_{IJ}]_{total}}{1+K_{IJ}[P]_{free}}$$

P: PNA, T: a target sequence, D: an off-target sequence (2)

또한, 전체 PNA의 반응양상은 식(2)와 같이 분류할 수 있다. 실험에 사용된 전체 PNA 가운데 각 범주에 포함되는 PNA의 양은 통계적 수치에 의존하게 된다.

3. 프로그램의 병렬화

전체 게놈의 길이를 a, 표적 유전자 서열의 길이를 b라 하고, 게놈 서열을 길이 b씩 겹치게 2b의 길이로 잘라준다. 그렇게 되면 2b 길이의 a/b-1개의 후보지역으로 분할된다. 해당 후보 지역을 Smith-Waterman 알고리즘을 적용하여 최대 유사성 점수와 위치를 구했다.

계산 노드의 개수를 n이라 할 때
1번 계산 노드는 1, 1+n, 1+2n, 1+3n, … 번째 서열
2번 계산 노드는 2, 2+n, 2+2n, 2+3n, … 번째 서열
……
n번 계산 노드는 n, 2n, 3n, … 번째 서열

이런 방법으로 a/b-1개의 구간에 동일한 양의 작업을 각 계산 노드에 분산시켜 주었다.

4. 병렬시스템 구축

표 1. 구축을 위한 H/W제원

Item	Quantity
2400 AMD CPU Opteron	16
ASUS SK8V Motherboard	16
512MB ECC PC2700	32
Netgear Gigabit Switch	1
Seagate 80giga HDD	16
Graphic Card	16

본 연구에서는 Rock-cluster[9]를 활용한 안정적인 시스템을 구축하였다. 개별 PC는 저장장치를 필요로 하며 Diskless환경과는 차이가 있다. Diskless 환경에서는 계산 노드의 경우 메인보드 및 CPU, 랜카드(내부랜)만으로 설정 및 작동이 모두 가능하지만 Rock-cluster를 이용한 시스템에서는 원활한 사용을 위해 그래픽 카드가 반드시 필요하였다.

5. 실험 결과

본 프로그램과 병렬 시스템을 이용해서 구체적인 예를 적용하여 보았다. 선택한 게놈은 *Escherichia coli* K12 이며 표적유전자는 16S ribosomal RNA(rRNA)였다. *E. coli* K12안에서 16S rRNA의 서열을 비교해서 97%이상 일치하는 구간을 찾게 설정해주었다. BLAST를 활용하였을 때 4639675bp의 *E. coli* K12에서 16S rRNA와 높은 일치도를 나타내는 부분의 시작 위치는 223771, 3939831, 4033554, 4164682, 4206170번째에서 나타났다. 본 연구에서 수행한 방법에서도 동일한 결과를 볼 수 있었다.

계산 노드 간의 작업이 상호 독립적으로 이루어져서 병렬화 효율이 뚜렷하게 향상되는 것을 볼 수 있었다. 1대의 계산 노드로 1169sec가 소요되었던 작업시간이 계산 노드가 증가되는 것에 정확하게 반비례하여 감소되는 것을 볼 수 있었다. 최종적으로 15대의 PC로 작업시에는 77sec가 소요되었고, 병렬화 시 생기는 통신부하 및 부하분산 불균형의 문제는 심각하게 발생하지 않았다.

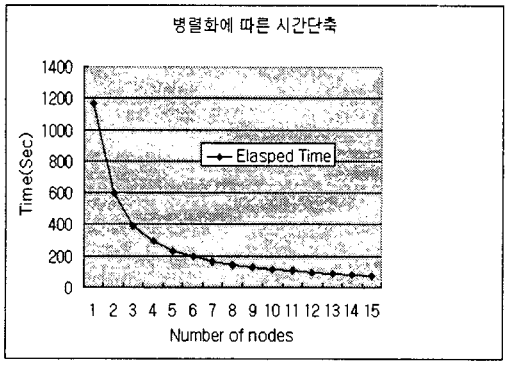

그림 3. 병렬화에 따른 시간단축

6. 결론 및 추가 연구

본 연구에서는 병렬 컴퓨팅 환경에서 구동이 가능하고 PNA의 off-target 효과를 최소화하는 서열 정보를 제공하는 프로그램을 개발하였다. 전체 게놈 서열 중 표적 유전자서열의 위치를 찾는 부분에서는 좀 더 빠른 실행 시간을 위한 알고리즘의 개선이 필요하며, 전체 게놈 및 표적의 특성에 따른 파라미터의 설정을 위한 경험적인 데이터의 축적이 필요하다. 후보자 서열의 모색에 있어서 열역학적인 요소를 적용하는 작업 역시 병행해 주어야 한다. 본 프로그램은 추후 PNA를 이용하는 생물학자들에게 좀 더 정확하고 효과적인 PNA 후보자 서열을 모색하는데 도움을 줄 것으로 기대되며, 실제 실험을 통해 얻어진 데이터를 통한 꾸준한 보완 작업을 필요로 할 것으로 사료된다.

7. 사사

본 연구는 산업자원부 기술혁신사업 (RTI04-03-05) 지원으로 수행되었습니다.

참고문헌

[1] Natalia Nekhotiaeva, Satish Kumar Awasthi, Peter E. Nielsen, and Liam Good, " Inhibition of Staphylococcus aureus Gene Expression and Growth Using Antisense Peptide Nucleic Acids" , *Molecular Therapy*, 10, 652-659, 2004

[2] Tomoyuki Yamada and Shinichi Morishita, " Accelerated off-target search algorithm for siRNA " , *Bioinformatics*, 21(8), 1316-1324, 2005

[3] Tommi Ratilainen, " A Simple Model for Gene Targeting" , *Biophysics Journal*, 81, 2876-2885, 2001

[4] Altschul, S.F. et. al.. 1990. Basic local alignment search tool. *J. Mol. Biol.* 215. 403-410.

[5] Karlin, S.,A, Dembo, and T. Kawabata, 1990, Statistical composition of high-scoring segments from molecular sequences, *The Annals. Of Statistics* 18, 571-581.

[6] Karlin, S., and S.F. Altschul, 1993. Applications and statistics for multiple high-scoring segments in molecular sequences. *Proc. Natl. Acad, Sci.* USA 90, 5873-5877.

[7] Needleman, S.B., and C.D. Wunsch, 1970. A general method applicable to the search for similarities in the amino acid sequences of two proteins. *J. Mol. Biol.* 48, 443-453.

[8] Smith, T.F., and M.S. Waterman, 1981. Identification of common molecular subsequences. *J. Mol. Biol.* 147, 195-197..

[9] http://www.rocksclusters.org