

마이크로어레이 이미지 분석을 위한 계층적 그리드 정렬 알고리즘

(A Hierarchical Grid Alignment Algorithm for Microarray Image Analysis)

천 봉 경 † 진 희 정 † 이 평 준 † 조 환 규 ††

(Bong-Kyung Chun) (Hee-Jeong Jin) (Pyung-Jun Lee) (Hwan-Gue Cho)

요 약 마이크로어레이(microarray) 실험은 수백 개 혹은 수천 개의 유전자 발현(expression) 정보를 한 번에 실험을 할 수 있기 때문에, 유전자 비교분석에 있어서 획기적인 실험 방법이다. 먼저, 각 유전자의 발현량을 측정하기 위해서 마이크로어레이 실험 결과로 생성된 이미지를 분석해야 한다. 그러나 마이크로어레이 이미지는 많은 유전자들로 구성되어 있기 때문에 사용자가 일일이 수동으로 유전자인 스팟(spot)들을 분석하기는 어려운 일이다. 이와 같은 문제점을 해결하기 위하여 메타그리드(meta-grid)를 이용한 그리딩(griding) 방법과 자동 그리딩 방법이 제안되었지만, 여전히 이 방법들은 문제점을 가지고 있다. 예를 들어, 메타 그리딩 방법은 동일한 마이크로어레이 칩에서 생성된 이미지일 지라도, 실험 시 발생하는 오류로 인해 그리딩 시 사용자의 수작업이 필요하다. 자동 그리딩 방법은 생성된 마이크로어레이 이미지가 잡음이 많거나 발현율이 낮을 경우, 이미지 분석 작업을 수행하지 못할 수도 있다.

본 논문에서는 메타 그리딩 방법과 자동 그리딩 방법을 혼합한 새로운 그리딩 방법인 자동 메타 그리딩 방법을 위한 계층적 그리드 정렬(hierarchical grid alignment) 알고리즘을 제안한다. 자동 메타 그리딩 방법은 메타 그리드를 입력으로 하여, 계층적 그리드 정렬 알고리즘을 통해 메타 그리드를 마이크로어레이 이미지에 맞게 자동으로 정렬해주는 방법이다. 실험 결과, 본 논문에서 제안한 방법은 이전 방법들보다 훨씬 강건하고 신뢰할 수 있는 그리딩 결과를 제공하였다. 또한 같은 칩에서 생성된 다수의 이미지들을 동시에 분석하는 일괄 분석(batch analysis) 시 제안한 방법을 적용함으로써, 보다 신뢰할 수 있는 일괄 분석 환경을 제공해 줄 수 있다.

키워드 : 생물정보학, 마이크로어레이 이미지 분석, 자동 메타 그리딩, 계층적 그리드 정렬

Abstract Microarray which enables us to obtain hundreds and thousands of expression of gene or genotype at once is an epoch-making technology in comparative analysis of genes. First of all, we have to measure the intensity of each gene in an microarray image from the experiment to gain the expression level of each gene. But it is difficult to analyze the microarray image in manual because it has a lot of genes. Meta-gridding method and various auto-gridding methods have been proposed for this, but they still have some problems. For example, meta-gridding requires manual-work due to some variations in spite of experiment in same microarray, and auto-gridding may not carried out fully or correctly when an image has a lot of noises or is lowly expressed.

In this article, we propose *Hierarchical Grid Alignment* algorithm for new methodology combining meta-gridding method with auto-gridding method. In our methodology, we need a meta-grid as an input, and then align it with the microarray image automatically. Experimental results show that the proposed method serves more robust and reliable gridding result than the previous methods. It is also possible for user to do more reliable batch analysis by using our algorithm.

Key words : bioinformatics, microarray image analysis, auto-meta-gridding, hierarchical grid alignment

· 본 연구는 과학기술부 생물정보학연구개발사업(M105290000105N290000110) 지원으로 수행되었습니다.

† 학생회원 : 부산대학교 컴퓨터공학부
bkchun@pearl.cs.pusan.ac.kr
hjjin@pearl.cs.pusan.ac.kr

pjlee@pearl.cs.pusan.ac.kr
** 정 회 원 : 부산대학교 정보컴퓨터공학부 교수
hgcho@pusan.ac.kr
논문접수 : 2005년 4월 18일
심사완료 : 2005년 12월 6일

1. 서론

마이크로어레이(microarray)는 기계 자동화와 전자 제어 기술 등을 이용하여 직계는 수백 개에서부터 많게는 수만 개의 DNA를 유리 박편에 그리드(grid) 구조로 심어 놓은 것이다[1-3]. 이를 이용하여 유전체의 발현 영상을 탐색하거나, 동시에 수천 개의 유전자들 간의 상호작용을 관찰할 수 있다. 그림 1은 마이크로어레이 칩 제작과정 및 실험과정을 보여준다. 마이크로어레이 실험을 위한 DNA 클론(clone)들은 여러 가지 장기기관이나 개체에서 추출하고, PCR(Polymerase Chain Reaction) 방법을 통하여 증폭하여 DNA 프로브(probe)로 칩에 심는다. 그 후, 실험군(test)과 대조군(reference) 샘플로부터 mRNA를 추출하고, 각각의 mRNA로부터 cDNA를 만들어 초록색(Cye3-dUTP)과 빨강색(Cye5-dUTP) 형광 물질로 염색을 하여 DNA 프로브와 합성 반응을 시켜본다. 실험은 시간변화나 온도변화 등의 다양한 조건으로 하며, 실험 후 광학 스캐닝 단계를 거쳐서 마이크로어레이 이미지 파일을 얻어 낸다. 생성된 이미지는 마이크로어레이 이미지 분석 소프트웨어를 이용하여 스팟의 발현량을 측정하여 유전자 발현 정도를 알아낸다.

마이크로어레이 실험 결과로 생성된 이미지는 그림 2에서 보듯이 많은 스팟들이 그리드 구조를 형성하고 있다. 마이크로어레이 이미지는 스팟들이 그리드 구조로 모여 하나의 블록을 형성하고, 블록들은 다시 이미지 전체의 그리드인 칩박스(ChipBox)를 형성하는 계층적 구조를 지닌다. 일반적으로 이상적인 마이크로어레이 이미

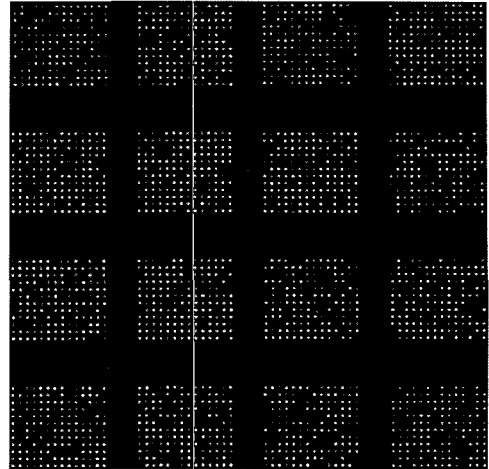


그림 2 마이크로어레이 이미지 예. 마이크로어레이 이미지는 4x4 블록과 각 블록은 12x14 스팟 그리드 구조로 구성되어 있다.

지는 다음과 같은 특징을 가진다[4].

- 모든 블록의 크기는 일정하다.
- 모든 블록 간의 간격은 일정하다.
- 블록 안의 스팟의 위치는 그 스팟이 속한 행과 열의 교차점 위에 위치한다.
- 스팟의 모양은 기하학적으로 원형이며, 모든 스팟들의 크기는 일정하다.
- 스팟의 위치는 같은 칩에서 실험한 모든 이미지들에 대해서 동일하다.

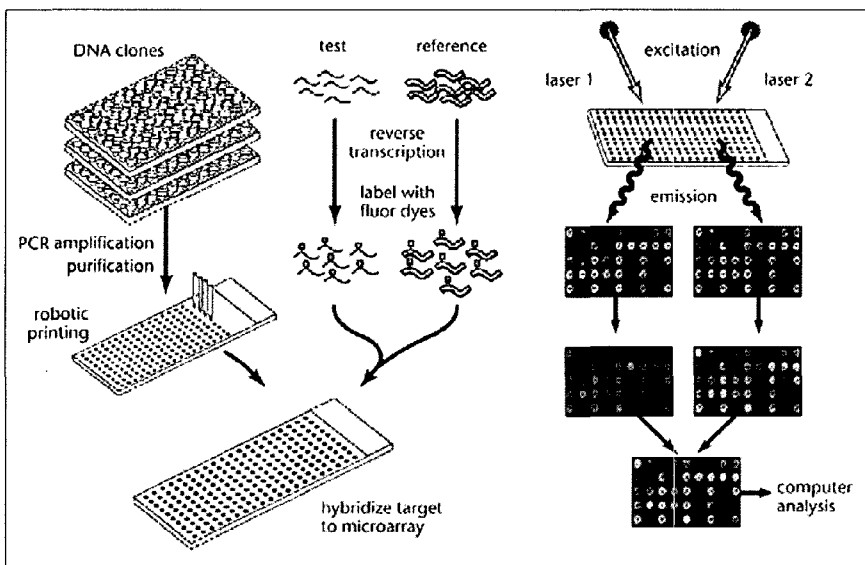


그림 1 마이크로어레이 칩의 제작 과정 및 실험과정[1]

하지만 대부분의 경우, 실험 또는 스캐닝 시 발생하는 오류로 인하여 마이크로어레이 이미지는 조금씩 이동되어 있거나 회전되어 있어 이상적인 특징을 모두 만족하기 어렵다. 이와 같은 문제로 인해 마이크로어레이 이미지를 자동으로 분석하는 것은 어려운 작업이다.

마이크로어레이 이미지내의 스팟들의 발현량을 측정하기 위해서는 각 블록과 스팟의 위치와 주소를 정확하게 결정하여야 하는데, 이러한 작업을 '그리딩(griding)' 이라고 하며, 스팟들의 발현량을 측정하는 것을 '강도 추출(intensity extraction)' 작업이라 한다. 초기에는 이 그리딩 작업은 사용자가 수동으로 모든 스팟들의 영역과 주소를 지정하여 분석하였으나, 기술이 발달함에 따라 마이크로어레이 내의 스팟들의 수가 증가하여 사용자가 수동으로 분석하기가 어렵게 되었다. 이와 같은 문제점을 해결하기 위해서 메타 그리딩(meta-gridding) 방법과 자동 그리딩(auto-gridding) 방법이 제안되었다.

메타 그리딩 방법은 마이크로어레이 이미지의 블록들과 스팟들의 위치 정보를 저장하고 있는 파일인 메타 그리드 파일을 이용하는 방법이다. GenePix[5]나 ImageGene[6]과 같은 대부분의 마이크로어레이 이미지 분석 프로그램에서 이 방법을 지원하고 있다. 그러나 이 방법은 메타 그리드 파일을 생성하기 위해 적어도 한번은 사용자가 수동으로 블록들과 스팟들의 위치를 지정하여 메타 그리드 파일을 생성해야 하는 단점이 있다. 또한, 동일한 칩에서 생성된 마이크로어레이 이미지들을 분석할 지라도, 실험 시 발생하는 오차로 인하여 이미지와 메타 그리드의 블록/스팟들의 위치들이 일치하지 않아 사용자가 수동으로 위치를 맞게 조정해야 한다.

자동 그리딩 방법은 사용자의 수작업 없이 자동으로 블록과 스팟의 위치와 주소를 결정하는 방법으로 많은 연구가 진행되었다[7-12]. Buhler[7]는 스팟의 모양과 위치에 대한 품질 제어 측정과 그리딩 방법을 연구하였다. 이 논문에서는 마이크로어레이 칩에서 생성된 이미지들의 평균적인 스팟의 위치를 미리 측정하고 이를 이용하여 스팟들의 주소를 결정하였다. 그러나 같은 종류의 칩에서 생성된 이미지에서만 가능하다는 단점이 있다. Jain[8]은 마이크로어레이 이미지의 행/열을 기준으로 히스토그램을 생성하고, 히스토그램의 분포 특성을 이용하여 블록과 스팟의 영역을 자동으로 그리딩하는 방법을 소개하였다. 이 방법은 이미지가 기울어 있거나 블록 내의 스팟의 발현율이 70% 이하가 될 경우, 올바른 그리딩 결과를 제공하지 못한다. 진회정[9]은 이미지에서 스팟으로 생각되는 컴포넌트(component)들을 찾아 이로부터 6균일 서열(regular sequence)을 생성하여 자동으로 그리딩하는 방법을 소개하였다. Kauer[10]은

역시 스팟으로 생각되는 컴포넌트들을 찾아서 일정 간격을 가지는 컴포넌트들로부터 그리드 구조를 생성하는 방법을 제안하였으며, 이 방법은 진회정 방법과 비슷하다. 일반적으로 자동 그리딩 방법들은 블록 간격과 스팟 간격의 차이가 뚜렷하거나 각 블록 내의 각 행/열에서 스팟의 발현율이 일정 비율을 유지해야하는 전제 조건을 만족하지 않을 경우, 그리딩 작업을 수행할 수 없다.

본 논문에서는 메타 그리딩 방법과 자동 그리딩 방법에서 발생하는 문제점들을 해결하기 위해서, 메타 그리딩 방법에 자동 그리딩 방법을 적용한 자동 메타 그리딩 방법을 위한 계층적 그리드 정렬(hierarchical grid alignment) 알고리즘을 제안한다. 이 알고리즘은 메타 그리드를 입력으로 하여, 마이크로어레이 이미지로부터 정렬의 기준 요소가 되는 목적 스팟(target spot)들을 찾은 후에, 이를 이용하여 칩박스, 블록, 스팟의 순서로 각 구조에 맞는 정렬을 계층적으로 수행하여 입력된 메타 그리드를 이미지에 맞게 정렬하는 알고리즘이다.

본 논문의 구성은 다음과 같다. 2절에서는 제안하는 계층적 그리드 정렬 알고리즘에 대해서 알아보고, 3절에서는 실험 데이터와 실험 결과에 대해서 알아보며, 마지막 4절에서는 결론을 논한다.

2. 계층적 그리드 정렬 알고리즘

계층적 그리드 정렬 알고리즘은 입력된 메타 그리드를 이미지에 맞게 자동으로 정렬 해주는 알고리즘이다. 이 알고리즘은 그림 3과 같이 전처리, 칩박스 정렬, 블록 정렬 그리고 스팟 정렬의 순서로 가장 큰 구조에서부터 작은 구조로 계층적으로 진행된다.

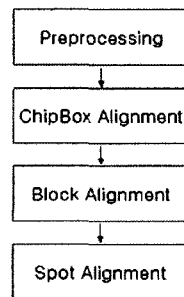


그림 3 계층적 그리드 정렬 알고리즘의 수행 단계

2.1 전처리단계

전처리 단계는 메타 그리드를 마이크로어레이 이미지에 정확하게 정렬을 수행하기 위한 기준점인 목적 스팟들을 찾는 단계이다. 이 단계는 크게 연결 컴포넌트 (connected component) 찾기 단계와 잡영 컴포넌트 필터링(noise component filtering) 단계로 구성된다.

연결 컴포넌트는 마이크로어레이 이미지에서 스팟으로 예상되는 컴포넌트로, 알고리즘 1을 이용하여 찾아낸다. 알고리즘 1은 마이크로어레이 이미지의 픽셀(pixel) 배열 값을 입력으로 하여, 처음부터 끝까지 목적 스팟과 배경(background)을 구분하는 경계값인 δ 와 비교를 한다. 만약 특정 위치 (i, j) 의 픽셀 강도값이 δ 보다 크다면 이 위치에서 *findConnected* 함수를 실행하여 목적 스팟의 영역을 찾는다. *findConnected* 함수는 (i, j) 위치를 기준으로 이웃 픽셀들의 강도값이 δ 보다 크다면 해당 픽셀을 목적 스팟의 영역에 추가하는 방식으로 목적 스팟 영역을 찾는 함수이다. *findConnected* 함수에서 더 이상 목적 스팟 영역에 추가할 픽셀이 존재하지 않을 경우, 이 함수를 종료하고 *add-spot-list*를 통해 찾은 목적 스팟을 *S*에 추가한다. 아직 검색하지 않은 나머지 이미지의 픽셀(pixel)들에 대해서 같은 과정을 반복한다.

알고리즘 1을 통해 찾은 연결 컴포넌트들에는 실제 스팟이 아닌 다양한 예러들이 포함되어 있기 때문에, 모든 연결 컴포넌트들이 계층적 그리드 정렬 시 기준점으로 사용되는 목적 스팟이 되는 것은 아니다. 연결 컴포넌트들은 그림 4에서의 같이 목적 스팟과 잡영 컴포넌트들로 구성된다. 따라서 연결 컴포넌트들에서 잡영 컴포넌트들은 필터링되어야 한다.

잡영 컴포넌트는 컴포넌트의 특성 및 필터링하는 방

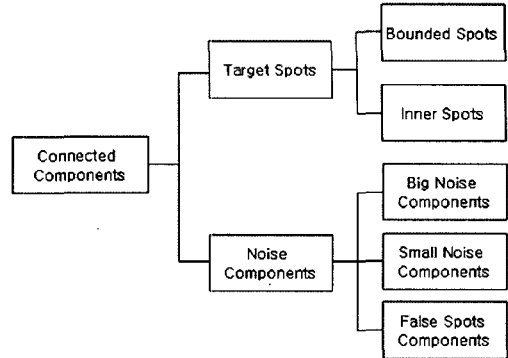


그림 4 연결 컴포넌트(connected component)의 구성요소. 목적 스팟(target spot)은 정렬 시 기준점으로 사용되는 컴포넌트를, 잡영 컴포넌트는 필터링되어야 하는 컴포넌트를 의미한다.

법에 따라 big/small 잡영 컴포넌트와 거짓 스팟 컴포넌트(false spot component)로 구분된다. Big/small 잡영 컴포넌트는 실제 스팟과 달리 원형이 아니거나, 실제 스팟의 크기와 일정 크기 이상 차이가 나는 컴포넌트를 의미한다. 이러한 컴포넌트들은 잡영으로, 크기와 둥근 정도(roundness)의 정보를 이용하여 제거할 수 있다. 본 논문에서는 다양한 실험을 통해 연결 컴포넌트들 중에서 실제 스팟과 비교하여 크기와 둥근 정도가 50%이

Algorithm: Connected Component Finding

Input: $I_{i,j}$, a pixel array of microarray image(width X height)

Output: *S*, a list of connected components

- *visited[i][j]* // an array to store visited pixel index
- *findConnected()* // a function to find connected component from (i, j) position
- *rect* // an area of connected component

```

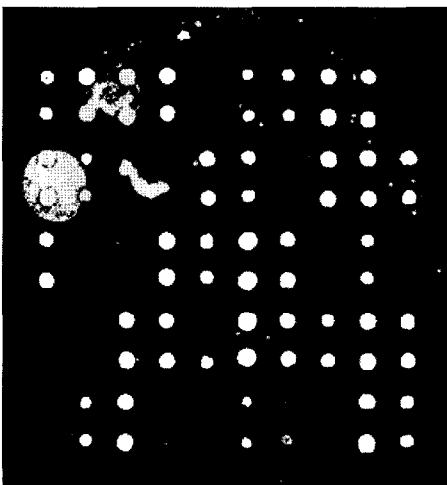
for i :=0 to width
  for j :=0 to height
    visited[i][j] = true;
    if  $I_{i,j} > \delta$  then
      Stack-push(i, j);
      findConnected(i, j, rect);
      add-spot-list(S, rect);
    endif
  endfor
endfor
    
```

알고리즘 1. 마이크로어레이 이미지에서 연결 컴포넌트를 찾는 알고리즘

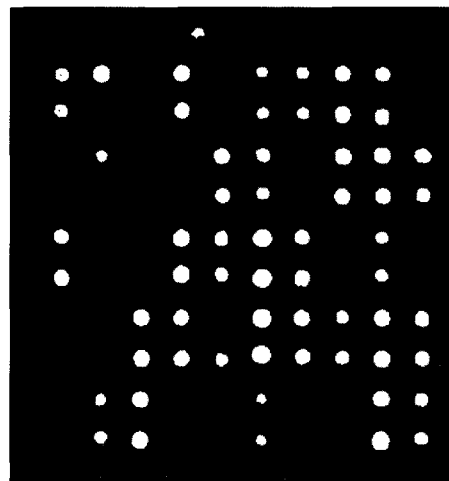
상 차이냐면 big/small 잡영 컴포넌트로 분류하여 제거 하였다. 그림 5는 big/small 잡영 컴포넌트 필터링을 수행하기 전과 후의 그림을 보여준다.

거짓 스팟 컴포넌트는 실제 스팟과 크기와 둥근 정도가 비슷하나, 그리드 구조에 포함되지 않는 컴포넌트를 의미한다. 이 컴포넌트는 그림 6(a)에서 보듯이 big/small 잡영 컴포넌트 필터링을 적용하더라도 제거되지 않는다. 이 컴포넌트를 제거하기 위해서는 추가적으로 블록 내 스팟들은 그리드 구조를 가진다는 특성을 이용

하여 필터링 작업을 수행한다. 이 필터링 작업은 각 연결 컴포넌트의 중심에서 수직선과 수평선을 생성하여, 만약 수직/수평선상에 임의의 다른 k 개의 컴포넌트들이 존재하지 않을 경우 거짓 스팟 컴포넌트로 생각하여 해당 컴포넌트를 제거한다. 이 필터링 작업과 이후 계층적 그리드 정렬 작업은 그리드 좌표계의 기울기가 0도라는 가정이 필요하기 때문에, ϵ 균일 서열[13]을 이용하여 그리드 좌표계를 0도인 이상적인 기울기로 변경한 후 수행한다. 그림 6은 거짓 스팟 컴포넌트 필터링을 수행

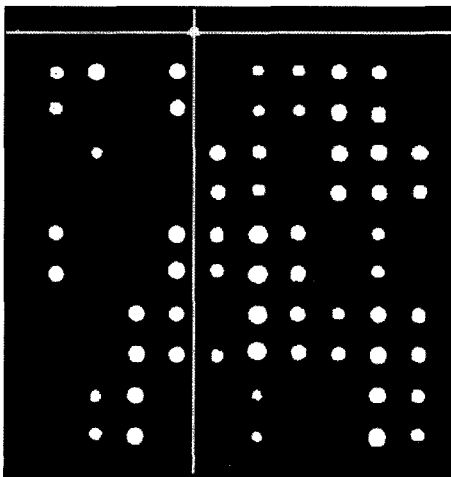


(a) Big/small 잡영 컴포넌트 필터링 수행 전

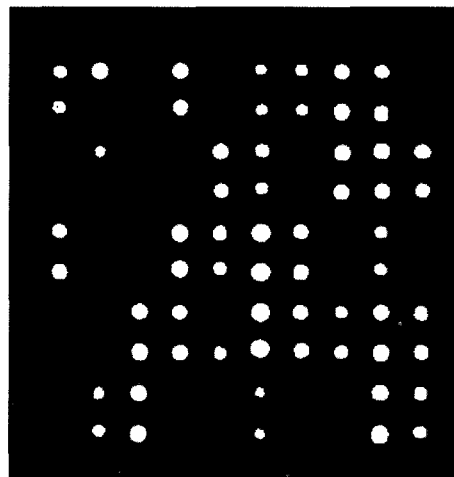


(b) Big/small 잡영 컴포넌트 필터링 수행 후

그림 5 Big/small 잡영 컴포넌트(noise component) 필터링을 수행하기 전(a)과 후(b)의 그림. 그림에서 노란색 도형들은 목적 스팟 또는 거짓 스팟 컴포넌트를, 회색 도형들은 big/small 잡영 컴포넌트를 의미한다.



(a) 거짓 스팟 컴포넌트 필터링 수행 전



(b) 거짓 스팟 컴포넌트 필터링 수행 후

그림 6 거짓 스팟 컴포넌트(false spot component) 필터링을 수행하기 전(a)과 후(b)의 그림. 그림 (a)에서 빨간색 선들은 필터링 수행을 위해 생성한 수직/수평선들, 그림 (b)에서 노란색 도형들은 목적 스팟을 의미한다.

하기 전(a)과 후(b)의 그림이다.

2.2 칩박스 정렬

칩박스 정렬 과정은 전처리 과정에서 찾은 목적 스팟들을 이용하여, 계층적 그리드 구조 중에서 가장 큰 영역인 칩박스를 정렬하는 과정이다. 칩박스는 모든 블록들을 포함하는 최소 경계 사각형(MBR: Minimum Bounding Rectangle)을 의미한다. 만약 칩박스의 각 변에 대해서 적어도 k 개 이상의 목적 스팟들이 존재한다면 칩박스를 찾는 작업은 아주 쉬운 작업으로 목적 스팟들로부터 최소 경계 사각형을 찾는 것과 동일하다. 알고리즘 2는 목적 스팟들을 모두 포함하는 칩박스인 $fChipBox$ 를 결정하는 알고리즘으로, 전체 목적 스팟 리스트인 S 에서 x, y 각각의 최소/최대 좌표를 결정하여,

```

Algorithm: fChipBox Finding
Input: S : a list of target spot components
Output: R : a fChipBox area

UINT min_x, min_y, max_x, max_y; // ChipBox Area

for i :=0 to n
    if (Si.x < min_x) then min_x = Si.x; endif
    if (Si.y < min_y) then min_y = Si.y; endif
    if (Si.x > max_x) then max_x = Si.x; endif
    if (Si.y > max_y) then max_y = Si.y; endif
endfor

return new Rect(min_x, min_y, max_x-min_x, max_y-min_y);
    
```

알고리즘 2. 마이크로어레이 이미지의 목적 스팟들로부터 fChipBox 영역을 결정하는 알고리즘

전체 목적 스팟들의 영역을 포함하는 MBR을 구하는 알고리즘이다. 이 후, 입력된 $mChipBox$ 를 이동하여 $fChipBox$ 에 맞게 정렬한다.

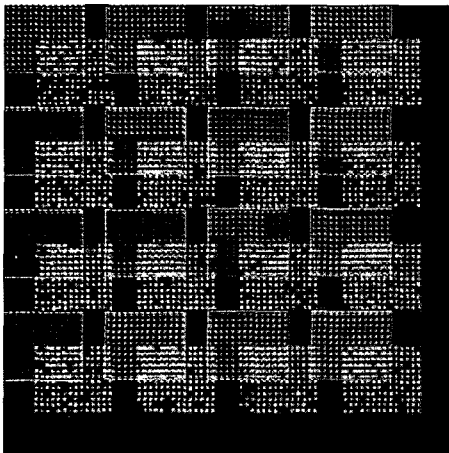
본 절 이후로 사용되는 용어들은 다음과 같은 규칙을 가진다. 입력된 메타 그리드의 요소들은 접두어 m 를 사용하며, 목적 스팟들로부터 찾은 영역들은 접두어 f 를 사용한다. 예를 들어, 위에서 사용한 $mChipBox$ 는 입력된 메타 그리드의 칩박스 영역을, $fChipBox$ 는 목적 스팟들로부터 찾은 칩박스 영역을 의미한다.

그림 7은 입력된 메타 그리드를 칩박스 정렬하기 전(a)과 후(b)를 보여준다. 그림 7(a)에서 녹색 도형은 입력된 메타 그리드를, 노란색 도형은 목적 스팟을 그리고 목적 스팟들을 둘러싸고 있는 빨간색 사각 영역은 $fChipBox$ 를 의미한다. 그림 7(b)는 칩박스 정렬을 수행한 결과이다.

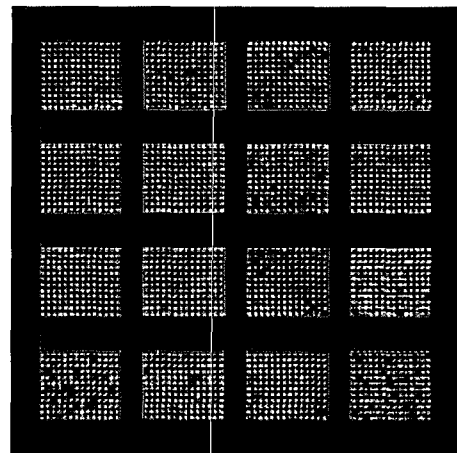
2.3 블록 정렬

칩박스 정렬 후 입력된 메타 그리드를 더 정확하게 정렬하기 위해 칩박스보다 세부영역인 블록 영역 별로 정렬을 수행한다. 블록 정렬 과정은 목적 스팟들로부터 실제 블록 영역인 $fBlock$ 을 찾는 단계 그리고 $fBlock$ 을 기준으로 $mBlock$ 들을 정렬하는 단계로 구분된다.

$fBlock$ 을 찾기 위해서 칩박스 영역을 각 블록을 포함하고 있는 영역인 블록 단위 영역으로 분할한다. 이 작업은 입력된 메타 그리드를 이용하여 블록의 크기와 블록간의 행/열 간격 정보를 계산한 후, 이 정보를 이용하여 칩박스 영역에서 블록 단위 영역을 계산할 수 있다. 이후, 각 블록 단위 영역의 해당 목적 스팟들을 이용하여 $fBlock$ 을 찾는다. $fBlock$ 을 찾는 방법은 앞 단계의



(a) 칩박스 정렬하기 전



(b) 칩박스 정렬하기 후

그림 7 입력된 메타 그리드를 칩박스 정렬하기 전(a)과 후(b). 녹색 도형은 입력된 메타 그리드를, 노란색 도형은 목적 스팟을 그리고 목적 스팟들을 둘러싸고 있는 빨간색 사각 영역은 fChipBox를 의미한다.

*fChipBox*를 찾는 방법과 동일하다. 그림 8은 2×2 블록 구조를 가지는 마이크로어레이 이미지를 입력된 메타 그리드 정보를 이용하여 블록 정렬을 수행하는 과정을 그린 그림이다. 그림에서 흰색 선들은 블록 단위 영역을, 녹색 도형들을 입력된 메타 그리드를, 노란색 도형은 목적 스팟들을 의미한다. 그리고 보라색과 빨간색 도형은 각 블록 단위 영역에서 찾은 *fBlock*을 의미한다.

*fBlock*이 결정되었으면 *fBlock*이 존재하는 블록 단위 영역 내의 *mBlock*을 *fBlock*에 맞게 정렬한다. 이 정렬 작업은 그림 8에서 보라색 영역인 *fGoodBlock*과 빨간색 영역인 *fBadBlock*에 따라 다른 작업을 수행한다. *fGoodBlock*은 블록 내 행/열의 스팟 발견율이 일정 비율 이상이어서 완벽하게 찾아진 *fBlock*이다. 따라서 *fGoodBlock*의 좌상단영역은 *mBlock*내의 $mSpot_{1,1}$ (1행 1열에 위치하는 스팟)과 매칭을 이룬다. 이 특성을 이용하여 $mSpot_{1,1}$ 과 *fGoodBlock*의 좌상단영역과의 좌표 차이만큼 *mBlock*을 이동하여 블록 정렬을 수행한다. *fBadBlock*은 *fGoodBlock*과 달리, 블록 내의 스팟 발견율이 낮아 불완전하게 찾아진 *fBlock*이다. 따라서 *mBlock*을 *fBadBlock*에 맞게 정렬을 하기 위해서 *fBadBlock*의 좌상단영역과 매칭되는 *mBlock*내의 *mSpot*의 인덱스(index)를 계산하여야 한다. 매칭되는 *mSpot*의 인덱스는 해당 *fBadBlock*과 같은 행과 열에 존재하는 *fBlock*들 중 *fGoodBlock*들의 좌표 정보를 이용하여 계산할 수 있다. 그 후, 계산된 인덱스의 *mSpot*을 *fBadBlock*의 좌상단영역으로 이동하여 블록 정렬을 수행한다.

$$SRowIndex = (fRowGoodBlock.Y - fBadBlock.Y) / SRowGap$$

$$SColIndex = (fColGoodBlock.X - fBadBlock.X) / SColGap$$

위의 식은 *fBadBlock*의 좌측 상단 영역과 매칭이 되는 *mBlock* 내의 *mSpot*의 인덱스를 계산하는 수식이다. 수식에서 *SRowIndex*와 *SColIndex*는 *fBadBlock*과 매칭되는 *mSpot*의 행/열 인덱스를, *SRowGap*과 *SColGap*은 *mSpot*들 간의 행과 열의 간격을 의미한다. 그리고 *fRowGoodBlock*은 *fBadBlock*과 같은 행에 존재하는 *fGoodBlock*을, *fColGoodBlock*은 *fBadBlock*과 같은 열에 존재하는 *fGoodBlock*을 의미한다. 그림 8에서 가로 점선은 *fRowGoodBlock.Y* 좌표를, 세로 점선은 *fColGoodBlock.X* 좌표를 의미한다.

2.4 스팟 정렬

블록 정렬 작업을 수행하였더라도, 그림 10(a)에서 보듯이 *mSpot*과 목적 스팟간의 중심 좌표가 어긋날 수 있다. 본 논문에서는 이러한 스팟간의 좌표 차이를 제거하기 위해 각 블록 내에서 스팟 정렬을 수행한다. 스팟

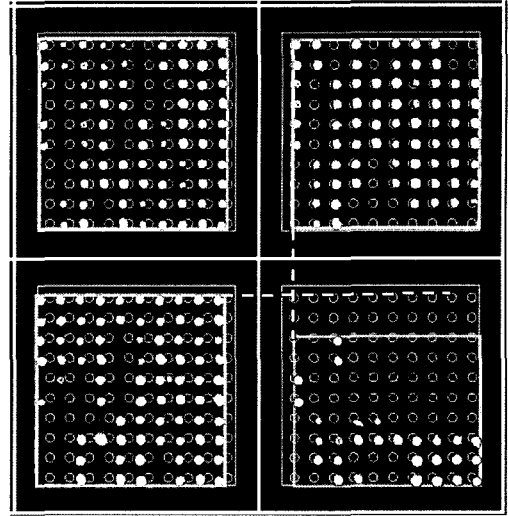


그림 8 2×2 블록의 구조를 가지는 칩박스 영역을 블록 단위 영역으로 분할한 화면. 흰색 선은 블록 단위 영역을, 녹색 도형은 *mBlock*을, 노란색 도형은 목적 스팟을, 보라색 선은 *fGoodBlock*을, 그리고 빨간색 선은 *fBadBlock*을 의미한다. 가로 점선은 *fRowGoodBlock*의 Y 좌표를, 세로 점선은 *fColGoodBlock*의 X 좌표를 의미한다.

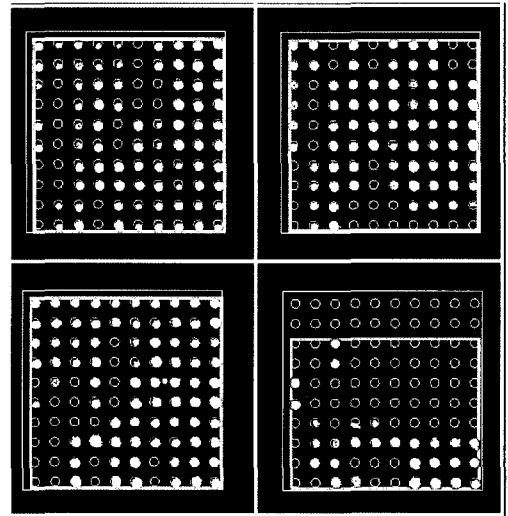
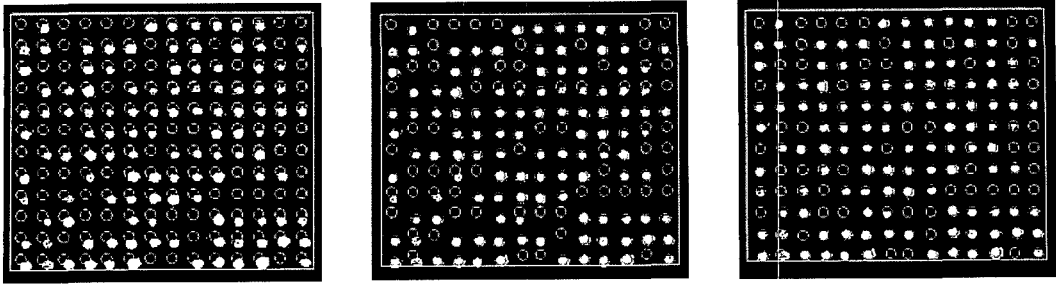


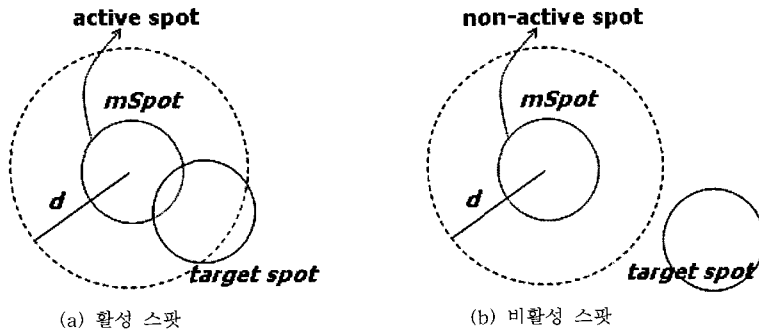
그림 9 그림 8을 블록 정렬을 수행한 결과 화면

정렬은 목적 스팟을 기준으로 *mSpot*을 정렬하는 작업으로, *mSpot*의 특성에 따라 활성 스팟(active spot) 정렬과 비활성 스팟(non-active spot) 정렬 단계로 구분할 수 있다. 그림 11은 활성 스팟과 비활성 스팟을 보여준다.



(a) 블럭 정렬 작업 완료 후 (b) 활성 스팟 정렬 (c) 비활성 스팟 정렬

그림 10 스팟 정렬의 진행 과정. 그림 (a)는 블럭 정렬 작업이 완료 후의 그림으로, 녹색 도형은 mSpot을, 노란색 도형은 목적 스팟을 의미한다. 그림 (b)는 활성 스팟 정렬 수행한 후의 그림을, 그림 (c)는 비활성 스팟 정렬을 수행한 후의 그림을 의미한다. 그림 (b), (c)에서 mSpot들 중 빨간색 도형은 활성 스팟을, 녹색 도형은 비활성 스팟을 의미한다.



(a) 활성 스팟 (b) 비활성 스팟

그림 11 활성 스팟과 비활성 스팟

활성 스팟은 mSpot의 중심 좌표를 기준으로 일정 거리 내에 목적 스팟이 존재하는 스팟으로, 이 mSpot의 좌표를 가장 가까운 목적 스팟의 좌표로 이동하여 정렬 작업을 수행한다. 본 논문에서는 다양한 실험을 통해 활성 스팟과 비활성 스팟을 구분하는 일정 거리 d를 스팟 지름 길이로 설정하였다. 그림 10(b)는 활성 스팟을 정렬한 결과이다. 활성 스팟 정렬 수행 후, 비활성 스팟들을 해당하는 행/열의 활성 스팟들의 좌표를 기준으로 정렬한다. 그림 10(c)는 비활성 스팟 정렬까지 수행한 결과이다.

3. 실험 결과 및 검토

계층적 그리드 정렬 방법을 평가하기 위해서 7종류의 마이크로어레이 칩(chip)에서 생성된 이미지 세트(set)들을 사용하였다. 표 1은 실험에 사용된 마이크로어레이 이미지들의 특징을 나타내며, 표에서 Chip은 사용된 마이크로어레이 칩의 종류, B와 S는 블럭과 스팟의 그리드 구조(행×열), Spot Dia.는 스팟 지름의 픽셀 길이, #Image는 각 칩 당 실험에 사용된 이미지 수를 의미한다. 칩 A부터 E까지는 국내 대학 연구실 및 기업에서 실험한 결과 이미지이며, 칩 F와 G는 스탠포드 마이크

표 1 실험 데이터의 특성. Chip : 마이크로어레이 칩의 종류, B : 블럭의 그리드 구조, S : 스팟의 그리드 구조, Spot Dia. : 스팟 지름의 픽셀 길이, #Image : 각 칩 당 실험에 사용된 이미지 개수

| Chip | B | S | Spot Dia. | Block Gap | Spot Gap | #Image |
|------|-------|---------|-----------|-----------|----------|--------|
| A | 4 × 4 | 12 × 14 | 14 | 98.58 | 12.20 | 5 |
| B | 4 × 4 | 10 × 10 | 16 | 90.75 | 18.90 | 8 |
| C | 4 × 4 | 18 × 18 | 14 | 34.13 | 8.53 | 8 |
| D | 4 × 4 | 10 × 10 | 16 | 91.66 | 18.81 | 8 |
| E | 4 × 4 | 18 × 19 | 14 | 30.10 | 7.62 | 8 |
| F | 4 × 4 | 18 × 18 | 16 | 18.58 | 7.50 | 12 |
| G | 2 × 4 | 26 × 26 | 10 | 29.83 | 5.84 | 6 |

로어레이 데이터베이스[14]에서 다운로드 받은 이미지이다.

표 1의 이미지 세트들에 대해서 사용자가 수동으로 직접 그리딩 작업을 수행한 이상적인 그리딩 결과와 3개의 다른 그리딩 결과와의 스팟의 평균 좌표 차이를 측정하였다(표 2). 실험 결과는 표 2와 같으며, Meta-Gridding은 메타 그리딩을, AutoGridding은 6 균일 서열을 이용한 자동 그리딩[9]을, Proposed Gridding은 본 논문에서 제안한 계층적 그리드 정렬을 이용한 자동 메타 그리딩을 의미한다. 실험 결과인 스팟의 평균 좌표 차이는 픽셀 거리를 기준으로 하여 유클리디언 거리(euclidean distance)로 측정하였다. 칩박스과 블록은 사용자가 개념적으로 정의한 것으로 스팟의 발현량 측정에 직접적으로 영향을 미치지 않기 때문에 본 논문에서는 칩박스과 블록의 좌표 측정을 고려하지 않았다.

표 2에서 알 수 있듯이 본 논문에서 제시한 방법은 전체 칩에 대해서 그리딩 작업 실패가 없을 뿐만 아니라, 이상적인 그리딩 결과와의 스팟의 좌표 차이가 1.17~2.33으로 가장 좋은 결과를 얻을 수 있었다. 그리고 이 차이점은 수동으로 결정한 이상적인 스팟 좌표의 오류로 생각 할 수도 있기 때문에 본 논문에서 제안한 그리딩 방법이 수동으로 정한 그리딩 방법보다 더 좋은 결과일 수도 있다. 반면, 메타 그리딩 방법은 이상적인 그리딩 결과와 적게는 스팟들의 지름 크기에서 많게는 수십 픽셀만큼 차이가 나기 때문에 신뢰할 수 없다. 그러므로 메타 그리딩 방법은 사용자가 수동으로 스팟들의 위치를 보정해 줘야하는 추가 작업이 필요하다. 자동 그리딩 방법은 블록 간격과 스팟 간격의 차이가 뚜렷하고 각 블록의 각 행/열에서 스팟의 발현율이 일정 비율을 유지해야하는 전체 조건을 지닌다. 마이크로어레이 칩 A와 F에서와 같이 자동 그리딩의 전체 조건인 블록 간격과 스팟 간격의 차이가 뚜렷하고 각 블록의 각 행/열에서 스팟의 발현율이 일정 비율을 유지한다면, 평균 2.80과 1.98로 좋은 결과를 보이고 있다. 그러나 이 결과는 본 논문에서 제안한 그리딩 결과인 1.39와 1.17보다는 좋지 않다. 또한, 칩 C와 E에서와 같이 발현율이

낮으면서 블록 간의 간격 차이가 스팟의 크기와 비슷하여 자동 그리딩의 전체 조건을 만족하지 못하는 경우, 표 2에서 보듯이 그리딩 작업을 수행하지 못하는 것을 알 수 있다. 다른 자동 그리딩 방법의 경우에도 이와 같은 전체 조건을 가정하기 때문에 본 논문에서 실험한 결과와 비슷한 결과를 보일 것으로 생각된다.

그림 12는 계층적 그리드 정렬 단계에 따른 그리딩 결과와 이상적인 그리딩 결과와의 평균 스팟 좌표 차이를 보여준다. 그림에서 x축의 'None'은 단순히 메타 그리딩만을 수행한 결과이며, 'ChipBox', 'Block', 'Spot'은 입력된 메타 그리드를 각각 칩박스 정렬, 블록 정렬, 스팟 정렬까지 수행한 결과를 의미한다. 그림에서 보듯이, 계층적 그리드 정렬이 진행됨에 따라 점차 이상적인 그리딩 결과에 근접하는 것을 알 수 있다.

그림 13은 마이크로어레이 이미지내의 스팟 발현율에 따른 계층적 그리드 정렬의 결과를 보여준다. x축은 이미지내의 스팟의 발현율을, y축은 이상적인 그리딩 결과와 계층적 정렬 그리딩 결과와의 스팟 거리 차이를 의미한다. 스팟 발현율이 10%일 때는 많게는 24.05에서 적게는 3.14차이를 보이며, 발현율이 20%일 때는 평균

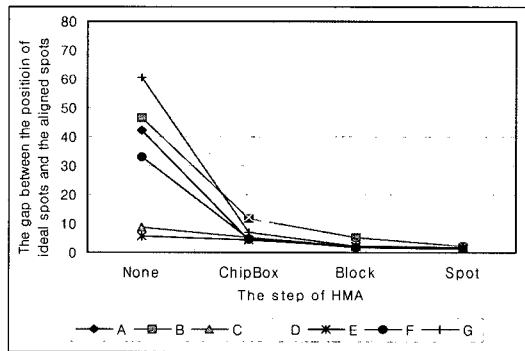


그림 12 계층적 그리드 정렬 단계 별 그리딩 결과와 이상적인 그리딩 결과와의 차이. X축은 정렬 단계를, Y축은 이상적인 그리딩과의 좌표차이를 의미한다.

표 2 각 칩에서 생성한 이미지들의 실험 결과. 표에서 셀(cell)은 각 칩에서 생성된 이미지들에 대해서, 각 그리딩 결과와 실제 사용자가 수동으로 수행한 이상적인 그리딩 결과와의 스팟의 평균 픽셀 좌표 차이를 의미한다. 표에서 Chip의 ()의 값은 실험에 사용된 이미지 개수를, fail은 실패한 이미지 수를 의미한다.

| Chip | MetaGridding | AutoGridding | Proposed Gridding |
|--------|--------------|------------------|-------------------|
| A (5) | 42.08 | 2.80 | 1.39 |
| B (8) | 46.66 | 14.27 (fail : 3) | 2.25 |
| C (8) | 8.72 | fail | 1.52 |
| D (8) | 72.24 | 13.01 | 2.33 |
| E (8) | 5.53 | fail | 1.92 |
| F (12) | 33.07 | 1.98 | 1.17 |
| G (6) | 60.36 | 3.37 (fail : 2) | 1.50 |

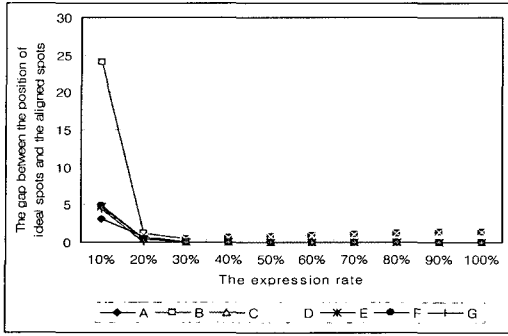


그림 13 마이크로어레이 이미지 내의 스팟 발현율에 따른 계층적 그리드 정렬 결과. X축은 스팟 발현율을, Y축은 발현율에 따른 이상적인 그리딩 결과와 계층적 정렬 그리딩 결과와의 차이를 의미한다.

적으로 0.58로 차이가 거의 없음을 알 수 있다. 즉, 마이크로어레이 이미지의 발현율이 20% 이상만 된다면, 신뢰할 수 있는 그리딩 결과를 얻을 수 있다.

실험 결과에서 알 수 있듯이, 본 논문에서 제안한 메타 그리드를 이용한 계층적 정렬 그리딩 결과는 사용자가 생각하는 이상적인 그리딩 결과와 차이가 거의 없으며, 자동 그리딩과 비교하여 보다 강건하고 정확한 그리딩 결과를 도출하는 것을 알 수 있다. 이와 같은 결과는 본 논문에서 제안한 방법이 자동 그리딩 방법과 달리 메타 그리드를 이용함으로써 블록의 크기, 블록/스팟간의 간격 등과 같은 정보를 추가적으로 이용할 수 있었기 때문이라고 생각된다.

4. 결론

마이크로어레이 실험은 수백 개 혹은 수천 개의 유전자 발현 정보를 한 번에 실험을 할 수 있기 때문에 유전자 비교분석에 있어서 획기적인 실험 방법이다. 마이크로어레이 실험 결과로 이미지가 생성되며, 이 이미지를 분석함으로써 유전자 발현량을 알 수 있다. 그러나 마이크로어레이 이미지는 수많은 스팟들을 포함하고 있기 때문에 사용자가 일일이 수동으로 분석하기는 어려운 일이다. 이와 같은 문제점을 해결하기 위해서 메타 그리드를 이용한 그리딩 방법과 자동 그리딩 방법이 제시되었지만 여전히 문제점들은 존재하였다. 본 논문에서는 이와 같은 문제점을 해결하기 위해서 메타 그리딩 방법에 자동 그리딩 방법을 적용한 새로운 그리딩 방법인 자동 메타 그리딩 방법을 제안하였다. 본 논문에서 제안한 그리딩 방법은 다음과 같은 특징을 가진다.

- 메타 그리딩 방법과 비교하여 보다 적은 사용자 수작업을 요한다. 동일한 칩에서 생성된 마이크로어레이

이미지를 같은 메타 그리드를 이용하여 분석할 지라도, 실험 시 발생하는 오차들로 인하여 메타 그리드와 마이크로어레이 이미지들 간의 오차가 존재한다. 따라서 사용자는 입력된 메타 그리드의 블록/스팟의 좌표와 크기를 수동으로 조정해 주어야 한다. 그러나 본 논문에서 제안한 계층적 그리드 정렬 방법을 이용하면 메타 그리드를 이미지에 맞게 자동으로 정렬해 주기 때문에 사용자의 수작업을 보다 더 많이 줄일 수 있다.

- 이전 그리딩 방법들보다 더 강건하고 정확한 그리딩 결과를 제공한다. 일반적으로 자동 그리딩 방법은 마이크로어레이 이미지의 발현율이 낮을 경우, 전체 그리드 구조를 다 생성하지 못해 그리딩 작업이 실패하는 경우가 종종 발생한다. 그러나 자동 메타 그리딩 방법은 실험 결과에서 알 수 있듯이, 이미지의 발현율이 20% 이상만 된다면 신뢰할 수 있는 그리딩 결과를 제공한다.
- 보다 신뢰할 수 있는 마이크로어레이 이미지 일괄 분석 환경을 제공한다. 마이크로어레이 실험 시 환경 및 기계적 요인으로 동일 칩에서 생성된 이미지일지라도 이미지들 간의 블록 및 스팟 좌표간의 오차가 생길 수 있다. 이와 같은 이유로 기존의 메타 그리딩 방법을 이용한 일괄 분석 작업은 사용자들이 거의 신뢰할 수 없었다. 그러나 자동 메타 그리딩 방법은 입력된 메타 그리드를 자동으로 이미지에 맞게 정렬해 주기 때문에 사용자가 신뢰할 수 있는 일괄 분석 작업을 가능하게 해준다.
- 본 논문에서 제안한 그리딩 방법을 적용한 마이크로어레이 이미지 분석 프로그램인 ArrayShop[15]을 개발하였다. ArrayShop은 제안한 그리딩 방법 외, 기본적으로 메타 그리딩 방법과 자동 그리딩 방법을 제공한다. 또한, 사용자가 신뢰할 수 있는 일괄 분석 환경뿐만 아니라 마이크로어레이 이미지가 얼마나 실험이 잘 되었나를 평가할 수 있는 품질 평가 결과도 제공해준다[16,17].

참고 문헌

- [1] David J. Duggan *et al.*, "Expression profiling using cDNA microarrays," *Nature Genetics*, Vol. 12, pp. 10-14, 1999.
- [2] Schena, M. and R.W. Davis, "Parallel Analysis with Biological Chips," *PCR Applications: Protocols for Functional Genomics*, pp. 445-456, 1999.
- [3] Schena, M., *Microarray Biochip Technology*, Eaton Publishing, 2000.
- [4] Ho-Youl Jung and Hwan-Gue Cho, "An automatic block and spot indexing with k-nearest neighbors graph for microarray image analysis," *Bioinformatics(Supple.)*, Vol. 18, pp. 141-151, 2002.
- [5] GenePix, <http://www.axon.com>.

- [6] ImaGene, <http://www.biodiscovery.com/imagene.asp>.
- [7] J. Buhler, T. Ideker, and D. Haynor, "Dapple: improved techniques for finding spots on DNA microarrays," *University of Washington CSE Technical Report UWTR, 2000-08-05*, 2000.
- [8] Ajay N. Jain *et al*, "Fully automatic quantification of microarray image data," *Genome Research*, Vol. 12, No. 2, pp. 325 - 332, 2002.
- [9] 진희정, 조환규, "균일 격자 구조 탐색을 이용한 마이크로어레이 반점 주소 결정 알고리즘," *정보과학회논문지*, 제31권, 제9·10호, pp. 514-526, 2004.
- [10] Gerhard Kauer and Helmut Blocker, "Analysis of disturbed images," *Bioinformatics*, Vol. 20, No. 9, pp. 1381-1387, 2004.
- [11] M. Steinfath *et al*, "Automated image analysis for array hybridization experiments," *Bioinformatics*, Vol. 17, No. 7, pp. 634-641, 2001.
- [12] Yang, Y. H *et al*, "Comparison of methods for image analysis on cDNA microarray data," *Computational and Graphical Statistics*, Vol. 11, pp. 108-136, 2002.
- [13] Gabriel Robins, Brian L.Robinson and Bhupinder S.Sethi, "On detecting spatial regularity in noisy images," *Information Processing Letters*, Vol. 69, No. 4, pp. 189-195, 1999.
- [14] Stanford microarray database, <http://genome-www5.stanford.edu>.
- [15] Bong-Kyung Chun *et al*, "ToMAS : software development toolkits for microarray analysis system," *Proc. Of ISMB*, Poster, 2004.
- [16] 김판규, 진희정, 조환규, "Quality measures for microarray design and experiment," *Proc. Of KSBI*, pp. 155-163, 2003.
- [17] Kuklin APA and Shams S, "Quality control in microarray image analysis," *Image & Microscopy Research*, Vol. 1, pp. 8-10, 2001.



이 평 준

2004년 부산대학교 정보컴퓨터공학부 졸업(학사). 2004년~현재 부산대학교 컴퓨터공학과 석사과정. 관심분야는 생물정보학, 컴퓨터 그래픽스



조 환 규

1984년 서울대학교 계산통계학과 졸업(학사). 1986년 한국과학기술원 전자계산학과 졸업(공학석사). 1990년 한국과학기술원 전자계산학과 졸업(공학박사). 1990년~현재 부산대학교 전기전자정보컴퓨터공학부 교수. 관심분야는 생물정보학,

컴퓨터 그래픽스



천 봉 경

2003년 부산대학교 정보컴퓨터공학부 졸업(학사). 2005년 부산대학교 컴퓨터공학과 졸업(공학석사). 2005년~현재 부산대학교 컴퓨터공학과 박사과정. 관심분야는 생물정보학, 컴퓨터 그래픽스



진 희 정

2000년 부산대학교 전자계산학과 졸업(학사). 2002년 부산대학교 전자계산학과 졸업(이학석사). 2002~2003년 국립보건원 유전체연구부 역학정보실 선임연구원 2003년~현재 부산대학교 박사과정. 관심분야는 생물정보학, 알고리즘 이론