

細胞アレイのための大規模画像処理システムの開発

富永大介^{1†)}、井口富久美²、堀本勝久¹、秋山泰¹

¹独立行政法人産業技術総合研究所、²株式会社情報数理研究所

細胞アレイや DNA マイクロアレイ、組織の顕微画像など、生命情報科学で対象とする実験観測の一次データが画像であることは多い。しかし観測対象や実験条件によって画像の様子は全く異なっており、一般には対象に特化した処理が必要である。画像が大量であれば処理の自動化も必須である。我々は、細胞アレイによる時系列画像から個々の細胞の活性と細胞形状を数値化し追跡するシステムを開発した。これにより、遺伝型と表現型を組み合わせたダイナミクスの大規模解析が可能となった。

Automated Image processing system for cell array observations

TOMINAGA Daisuke¹, IGUCHI Fukumi², HORIMOTO Katsuhisa¹,
AKIYAMA Yutaka¹

¹National Institute of Advanced Industrial Science and Technology,

²Information and Mathematical Science Laboratory Inc.

In many cases of biological observations such as cell arrays, DNA microarrays or tissue microscopy, primary data are obtained as photographs. Specialized processing methods are needed for each kind of photographs generally because they have very wide variety, and often needed automated systems for modern high-throughput observations which produce a huge amount of data. We developed a fully-automated image processing system for the cell array, high-throughput time series observation system for living cells, to evaluate gene expression levels and phenotype of each cell. It makes possible to analyze dynamics of life systems based on both genotypes and phenotypes of cells.

1. はじめに

たんぱく質の二次元電機泳動、DNA マイクロアレイ、細胞アレイなど、分子生物学において、実験結果が画像データとして得られるケースは多い。また近年開発された実験手法はハイスループット化が著しく、網羅的な観察を目的として大量の画像データが得られるようになってきている。現在、我々が研究対象としている細胞アレイ^[1-3] (図 1) では、スライドガラス上に 500~1500 個の小

さな穴 (スポット) をうがち、その中で細胞を培養し続けながら遺伝子発現の量や形態の変化を蛍光タンパクなどで観察する。たとえば 1500 スポットのスライドガラスを使い各スポットについて 15 分おきに 2 日間写真撮影で観察した場合、 $1500 \times (24 \times 60 / 15) \times 2 = 288,000$ 枚の撮影画像が得られる。この約 30 万枚の画像のそれぞれに対して、バックグラウンドのモデリングと減算、スポットノイズ除去、細胞の輪郭の認識、細胞の蛍光強度の積算を行い、また時系列撮影であることから、各細胞の追跡や形状変化の数

[†]) Email: tominaga@cbric.jp

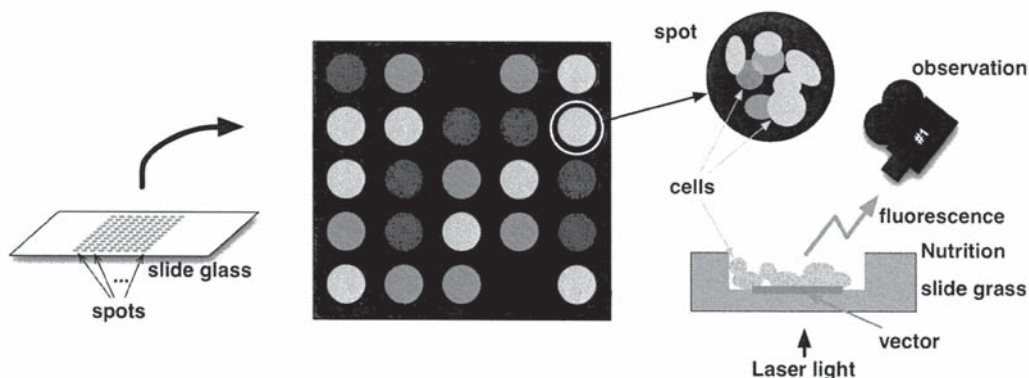


図1 細胞アレイによる生細胞の時系列観測。スライドグラス上の樹脂被服に直径数 μm の穴を開けガラスを露出させる（スポット）と、そこに細胞が吸着する。スポットにあらかじめ蛍光タンパクなどの遺伝子をベクターに組み込んで付着させておけば、そのベクターが細胞内、核内へと移行して発現し、発光する。特定の遺伝子と発現強度が比例するようにベクターを設計しておけば、その遺伝子の発現強度の変化を発光強度で観測することができる。

値化などを行う。

一般にこういった画像処理では、ノイズとそうでないものとの区別に経験的な判断を要することから、観測者による作業を必要とすることが多い。たとえば細胞の大きかなサイズや形状、バックグラウンド蛍光のおおよその強度などの知識を元に、観測装置に付属しているソフトウェアなどで実験条件に最適な処理パラメータの調整を行い、誤認識や外れ値の出現具合を見ながら半自動で処理を行う。こういった場合、対象としている実験に関する知識と経験が多いほど、またかける手間と時間が多いほど高品質な画像処理、つまりよりノイズを減らした精密な数値化ができるため、有能な研究者が長時間拘束されることがある。また画像をモニターで確認しながらパラメータ値の微調整を続けることで VDT 症候群などの健康被害をもたらすこともある。これまでの例では、我々の細胞アレイ画像では約 2 万枚の画像を一人の研究者が 1 ヶ月かかって処理しており、これが 30 万枚になった場合、二日間で得られる画像データの処理に 1 年かかることになる。細胞アレイによるハイスループットな網羅的観測能力を最大限に生かすためには、情報

科学による支援が不可欠である。

そこで、マニュアル操作を極力排除し、大部分の処理を自動で並列計算により行うシステムを開発した。ソフトウェアは米 NIH による画像処理システム ImageJ^[4]をベースに Java 言語 (version 1.4.2 以上) で実装され、Linux および Windows XP 上で動作を検証している。

2. 入出力

2.1 対象とする観測データ

細胞アレイ装置による、単色の時系列撮影画像を処理する。一回の実験で得られる複数の画像 (枚数は実験で使ったスライドグラス上のスポット数 \times 撮影回数) を入力単位とする。各画像は一つの直径数 μm のスポットを撮影した TIFF 形式の顕微画像であり (図 2)、写っているスポットの番号と何回目の撮影であるかをファイル名で保持する。倍率、露光時間などの情報は画像処理の結果には直接関係せず、別途ファイルで保持する。

2.2 ユーザーが設定するパラメータ

観測対象によって細胞の大きさは大きく異なるので、細胞の直径の上限と下限の数値

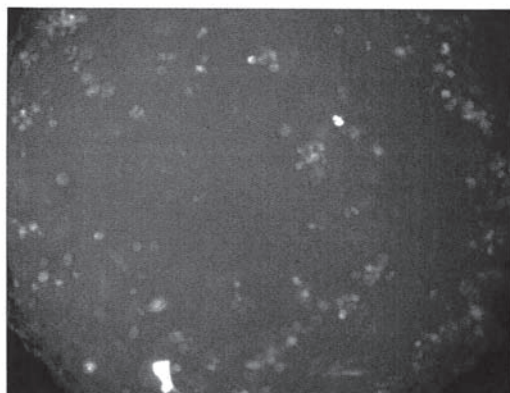


図2 スポット画像例。スポットは円形であり必ずしも全体が写っている必要はなく、スポット全体の様子が把握でき、かつ細胞の大きさが最適となるように撮影する。

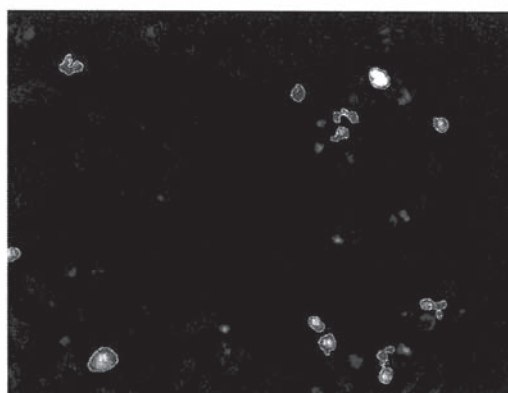


図3 細胞の認識例。図1の画像に対してバックグラウンド補正、閾値処理および輪郭抽出を行い、細胞の輪郭を図示し各細胞に番号を割り当てた画像。

表1 画像から抽出される特徴量

静止画	細胞の輪郭 各細胞内の輝度合計、平均輝度 各細胞の重心、真円率 画像全体での輝度合計、平均輝度 (スポット合計、スポット平均)
時系列処理	画像間の各細胞の対応

をソフトウェアに与える。他に、入力ファイルのあるディレクトリおよび出力先のディレクトリを指定する。

2.3 出力データ

1枚の画像から、画像中の明るさの合計と、それを細胞の面積合計で割った平均値(スポット合計、スポット平均)、各細胞についての明るさの合計と平均、各細胞の重心と真円度を出力し、ある時点の画像中の細胞がその直後に撮影された画像中のどの細胞に対応するかを判定する(表1)。出力はすべてテキストファイルに書き込まれ、スポット合計とスポット平均は各スポットごとのファイルに、各細胞についての計算値は各スポットごとのディレクトリ中の各細胞ごとのファイルに保存する。これらの数値をプロットすることで、細胞の挙動を概観し、ま

た各細胞を比較することで外れ値を除くことができ、信頼性を向上したデータに基づいた解析を行うことができる。また検証のため、入力ファイルの解像度を低くし、そこに認識した細胞の輪郭を描き加えた画像ファイルを出力する(図3)。

2.4 処理方法

画像を一枚ずつ読み込んで、1)バックグラウンド補正、2)閾値処理、3)輪郭抽出と輝度積算を行い、すべての画像の処理が終わってから画像間の細胞の対応を判定する(図)。

2.4.1 バックグラウンド補正

円形のスポットの撮影画像であることを考慮して、シェーディング補正^[5,6]によりバックグラウンド蛍光のモデリングを行い、それを差し引くことで細胞以外の部分の画素を取り除く。まず処理時間短縮のために画像を、縮尺で1/10に縮小する。縮小画像に半径2ピクセルのモルフォロジ・フィルタ(膨張/収縮フィルタ)^[6]でスムージングした画像をを元のサイズに戻し、戻すことによって発生するブロックノイズをガウシアン・フィルタ^[5,6]でスムージングする。これをバックグラウンド画像とする。元の画像が

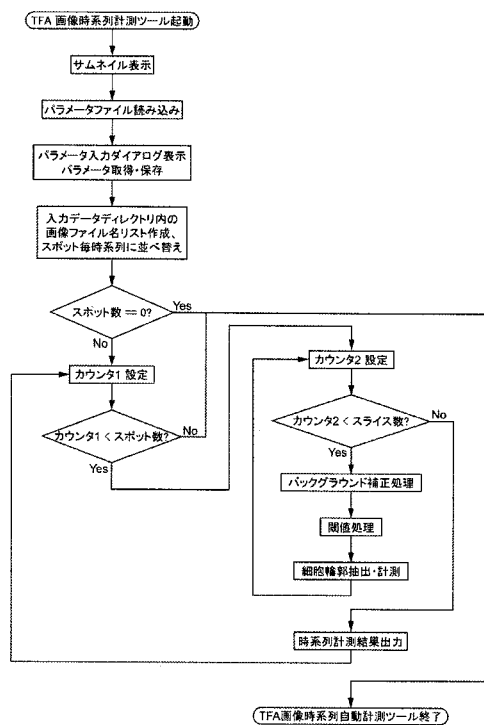


図 4 全体の処理の流れ。処理は各スポットごとに行われる (外側のループ)。各個の画像に対する処理は、バックグラウンド補正、細胞の認識、輝度積算と真円度計算の順に行われる (内側のループ)。

らバックグラウンド画像の輝度を差し引き、差し引いた画像のすべてのピクセルについて、その輝度にバックグラウンド画像の平均輝度をに加えることで、補正済み画像を得る。

2.4.2 閾値処理

細胞と背景を分離するために、補正画像の全体から閾値を差し引く。まず最大エントロピー法で閾値の初期値を計算する。この値を画像全体から差し引くと暗く写っている細胞が取り除かれてしまうため、これから画像全体の輝度分布の標準偏差を引いた値を閾値とする。

2.4.3 細胞の輪郭抽出

画像全体で、上述の閾値より輝度の高いピ

クセルを 1、低いピクセルを 0 とすることで二値化を行う。1 になったピクセルを対象に、その輪郭線を追い閉曲線を作ることで細胞の輪郭とする^[5,7]。

閉曲線には番号を振り、一つの閉曲線を一つの細胞であると判断する。各閉曲線をバックグラウンド補正画像に複写し、そこで細胞内の輝度値を積算する。一枚の画像内のすべての細胞の、細胞内のすべてのピクセルの輝度値合計をその画像の輝度値とする。また画像の輝度値を細胞の合計面積 (ピクセル数) で除したものを平均輝度とする。

画像内の各細胞についても輝度値を計算し、同時に重心と真円率 ($4\pi \times \text{面積} / \text{周囲長}$) も計算する。重心は輝度値による重みづけは行わず、閉曲線だけから計算される。

2.4.4 時系列情報処理

各画像での細胞の重心の位置を元に、複数の画像間で同じ細胞と思われるものの対応をとり、細胞ごとに輝度値、平均輝度、重心、真円率の時系列データを構成する。

時間的に連続する二枚の画像を比較し、時刻が前の画像にある細胞と、その重心位置に最も近く、判定距離内に重心がある後の画像内の細胞が、同じ細胞であると判断される。判定距離は、前画像での細胞と同面積の円の半径の 2.5 倍とする。このため、運動量の大きな細胞を観測する場合は、撮影間隔を短くする必要がある。後の画像に対応する細胞がない場合は、その細胞は消えたとされるが、さらに後の画像で条件に合致する細胞が現れた場合は同じ細胞であるとみなされる。逆にそれまで細胞がなかったところに細胞が現れた場合は、新しく現れたものとされる。

3. 実装

米 NIH による ImageJ v. 1.37 を改良して前節の処理法を実装した。シェーディング

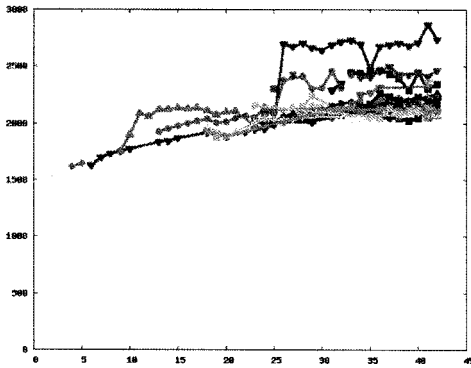


図5 スポット内に認識されたすべての細胞の平均輝度のプロット。横軸は時間 (h)、縦軸は蛍光強度 (無次元)。細胞数は 56 個。

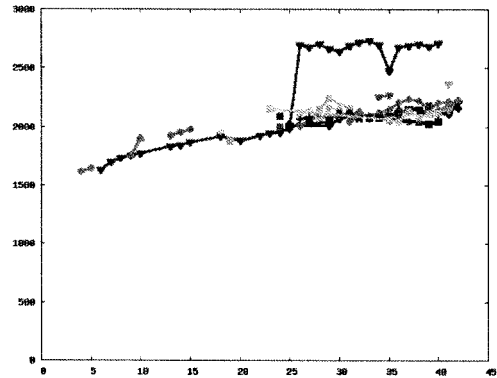


図6 真円度の高い細胞を死んだものとして除いた平均輝度のプロット。横軸は時間 (h)、縦軸は蛍光強度 (無次元)。細胞数は 29 個。

補正、ガウシアン・フィルタ、モルフォロジ・フィルタ、閾値処理、輪郭抽出は ImageJ に実装されているものを利用した。バックグラウンド補正、閾値処理、真円率の計算、時系列処理は ImageJ のプラグインとして Java 言語でプログラムを作成した。

プログラムは Java 1.4.2 以上の実行系のネイティブ・スレッドおよび、UNIX 系の OS では複数のプロセスによる並列実行に対応するよう作成した。

4. 実行例

432 スポット × 43 枚の細胞アレイの画像を処理することで作成したプログラムを検証した。

実験では HeLa 細胞に対して 12 種類のベクターを導入し、それによる影響を 9 種類のレポータータンパクで観測した。代表的な画像を図 2、それを処理した画像を図 3 に示した。この場合実験条件が $12 \times 9 = 108$ 通りあり、各条件について 4 つのスポットを割り当てた。これは、その 4 スポットを統計処理することで観測値の信頼性を向上することを意図している。スポットの合計数は $108 \times 4 = 432$ であり、各スポットについて、1 時間おきに 42 時間撮影した(撮影は合計 43

回)。画像枚数は合計で $432 \times 43 = 18576$ である。プログラムは 8 CPU core の SMP 構成の計算機上で 8 プロセスで並列実行し、すべての画像を処理し時系列データを作成するのに約 8 時間を要した (Dual-core Opteron 886 / 2.4GHz / 1MB cache × 4 CPU (8-core SMP 構成), メモリ 32GB, J2RE 1.4.2, Linux kernel 2.6.5)。

各スポットでは平均して 51.1 個の細胞が認識された (標準偏差 19.4)。一つのスポットについての、各細胞の平均輝度のプロットを図 5 に示す。

この実験では、死んだ細胞はガラス表面からはがれ、表面張力で円形になることを考慮し、一度真円度が高く (0.8 以上に) なった細胞は死んだ細胞であると判断して除去することで、生細胞だけの挙動を追った。そのプロットを図 6 に示す。

5. 考察

開発したソフトウェアを使うことで、細胞アレイ上のスポット内の各細胞について、生細胞中の 9 種類の遺伝子の活性と、細胞の形状を同時に測定、追尾することができた。細胞は死んでいても蛍光を失わないため (数時間の半減期で指数的に減少していく)、蛍光

だけを観測した場合には、遺伝子の活性が生きた細胞と死んだ細胞の両方から計算されてしまう恐れがあるが、形状から細胞の生死を判定することによってその問題を回避し、より信頼性のより数値化を可能にした。

分子生物学、あるいは生物学全般においては観測対象によって画像の特徴が非常に大きく異なるため、画像が大量にある場合でもその処理は半自動化に留まるのが一般的である。つまり細胞アレイの場合ならたとえば、細胞がどのように認識されたかを目で見て、認識されるべきなのにされていないものがあれば、そこをクリックすることで認識をやり直す、またノイズを細胞とご認識している場合にもそこを指定するといったことを、すべての画像に対して行わねばならない。ここで開発したソフトウェアでも認識できない細胞はあるが、細胞の大きさを指定することと、バックグラウンド補正を施すことでノイズを誤認識することは少ないことを画像で確認した。これにより質の高い数値化ができていていると考える。これは、プログラム内では数値データの正規化は行っていないのに図 5、6 で各細胞の平均輝度の値、および折れ線のプロファイルがよく一致していることが示している。

その一方で、目を見た場合に明らかに確認できる細胞もノイズと同時に除去してしまうため、暗い細胞は形状が明確な場合でも認識できない。これを解消するためには、閾値処理だけでなく微分フィルタ、FFT フィルタ、ウェーブレットなどによる処理も今後検討する必要がある。

同じ処理を、撮影に用いた実験装置に付属するソフトウェアで行った場合に 1 枚の画像に約 30 秒かかるとすると、今回のデータでは約 155 時間、一日 8 時間とすると 19 日かかる計算になる。観測が 2 日弱で終了することを考えると、画像処理がボトルネックに

なってしまうことになる。細胞アレイのハイスループット性を損なわないためには、少なくとも観測時間よりも早く処理を終える必要があり、今回開発した処理システムでは並列実行によりそれを達成している。SMP 計算機ではなく単一の CPU コアにより実行した場合は、同じ処理に 60 時間程度かかると想像され、実験観測に追いつかない。さらにノイズフィルタや輪郭抽出にインテリジェントなアルゴリズムを導入すると計算時間が大幅に増加することが予想されるため、並列実行による高速化は不可避であり、SMP 環境だけでなくグリッド環境による実行にも対応すべきであると考えている。

6. 備考

この研究は独立行政法人新エネルギー・産業技術総合開発機構 (NEDO) による研究委託「モデル細胞を用いた遺伝子機能解析技術開発/細胞アレイなどによる遺伝子機能の解析技術開発」により行った。

7. 参考文献

- [1] Junaid Ziauddin, David M. Sabatini, "Microarrays of cells expressing defined cDNAs", *Nature*, **411**, pp. 107-110 (2001).
- [2] Tomohiko Yoshikawa, et. al, "Transfection microarray of human mesenchymal stem cells and on-chip siRNA gene knockdown", *Journal of controlled release*, **96**, pp. 227-232 (2004).
- [3] Beate Neumann, et. al., "High-throughput RNAi screening by time-lapse imaging of live human cells", *Nature methods*, **3**, 5, pp. 385-390 (2006).
- [4] <http://rsb.info.nih.gov/ij/>
- [5] Gonzalez Woods, *Digital Image Processing*, Prentice Hall (2003).
- [6] 酒井幸市, デジタル画像処理入門, コロナ社 (1997).
- [7] 石田隆行編, C 言語で学ぶ医用画像処理, オーム社 (2006).