

ヒトと機械で画像を観る

誌名	蚕糸・昆虫バイオテック = Sanshi-konchu biotec
ISSN	18810551
著者名	加藤,輝 林,茂生
発行元	日本蚕糸学会
巻/号	89巻1号
掲載ページ	p. 7-13
発行年月	2020年4月

農林水産省 農林水産技術会議事務局筑波産学連携支援センター

Tsukuba Business-Academia Cooperation Support Center, Agriculture, Forestry and Fisheries Research Council
Secretariat



特集「定量生物学事始め」

ヒトと機械で画像を観る —生物画像を定量的に解析する—

加藤 輝^{1,*}・林 茂生²

¹ 大学共同利用機関法人自然科学研究機構生命創成探究センター, 基礎生物学研究所

² 理化学研究所 生命機能科学研究センター

定量的な生物学と聞いて何が思い浮かぶか。

生物学において、定量的な解析といえば、アレイを用いた遺伝子発現量の解析などが浮かぶ。

これは単純にはある目的となる表現型を単一の数量として記述可能なものとして計測することができるため、非常に扱い易い表現であるということが出来るかもしれない。他方で、伝統的な生物学の方法論を踏襲する学生実習時代を思い返すと、「生物学の基本、まずはスケッチ」と言われ、顕微鏡を覗きつつ実験ノートにウニやら、葉などを精緻に写しとるつもりが自身の絵心の無さに悶絶するといったような惨憺たる過去が蘇る。そのような経験が物語る通り、生物学は「画像」を中心に据えている側面が大きい。これまでは、野生型と変異型、それぞれ「同じ」条件で撮影した画像のうちから尤もらしい2枚を選び出して並べるだけの「一目瞭然」型の記述で大抵の雑誌は納得してくれたが、近年はどうも事情が異なり、「違いを定量的に説明しなさい」というコメント付きで再提出を要求されるようになった。確かに、「チャンピオンデータ」と揶揄されるような方式で恣意的に選び出した尤もそれらしい2例を並べるだけでは客観性に乏しく、かと言ってこれまで撮影した観察事例全てを一様に並べ立てるのでは単なる資源の無駄遣いとなってしまふ。つまり、「定量的解析」とは、雑誌の紙数を節約しつつ、観察の客観性を担保するために編み出された策なのかもしれない。

それでは、これまで矢印で示して「ここが変だ」と言い張っていた記載を改め数値で記述するにはどうすれば良いのか。実は、定石と言えるような方法論は存在しない。対象・可視化方法が研究事例ごとに異なることもあって、具体的に何を測りとるべきかを論じることが困難なほど多様性に富んでいることがその原因である。このような状況に拠ってか「何か違いを（あるはずだから）計測して」といった依頼が舞い込む事例は多い。このような依頼について、まさか背景の暗さの違いを定量的に解析した結果を返すわけにも行かないので、まずは研究の背景などから議論を行い、計測を行う対象の見極めを行うことが常である。稀に、計測方法についてかなり具体的な指定がなされる場合もあるが、これについても、実際の計測方法としては煩雑だったり、あるいは「違い」を記述するためには少々不適切だったり、やはり多少の擦り合わせを要する。このような経験から、生物画像の定量的解析は、計測対象を選択、単純化して、これを数値に置き換える手続きを最も簡単に行えながらも、目視による直感的な理解を裏打ちできるような手法を探る作業であると言える。

このような抽象化は、例えば形態を記述するような方法として単回帰を含むような単純なモデルへの当て嵌まりの良さ、といったような数値化を含む。それでは、この種の抽象化を行うためには、どのような技法があるのだろうか。過去には、カウンタ片手に目視で計数したり、ディスプレイに定規を貼って、という荒技を目撃したことがある。このように、「気合と根性」で画像を計測する方法はしかし、再現性と客観性において問題が生じる可能性がある。計測した結果が後日改めて計測した結果と大きく異なることもあれば、深夜遠くなる意識と闘い

*〒444-8585 愛知県岡崎市明大寺町字西郷中 38
生命創成探究センター 生物画像情報解析グループ
E-mail: kkat@orion.ac.jp

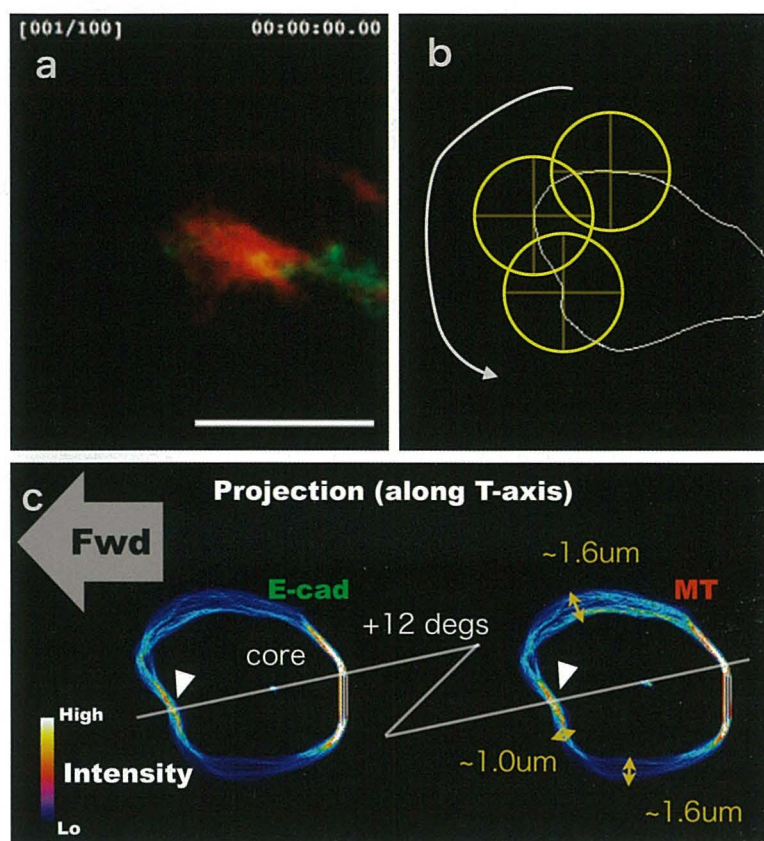


Fig. 1. 気管先端細胞における E-Cadherin と微小管の共局在解析

E-Cadherin (緑) と微小管 (赤) をそれぞれ可視化した気管先端細胞のタイムラプスイメージングデータ (a) について、細胞辺縁部における共局在性を解析したもの。b. 計測方法。c. 計測した辺縁部の平均輝度について、輝度の時間平均を投影したもの。E-Cadherin と微小管の局在の類似性が高いことがわかる。スケールバーは 10 μm 。

ながら計測した結果に無意識の願望が入り込む隙を排除するのは難しかったりするかもしれない。心理的な葛藤をも含むであろう振幅を除くには、計測において厳密な条件を設定し、それを忠実に解釈し実行する道具立てが必要となるだろう。この種の単純労働を完遂する上で、適切にプログラムされた計算機は実行効率と再現性において人間を圧倒する。つまり、画像計測はプログラミングと切っても切れない縁にある。

画像解析においてプログラミングが威力を発揮した事例としてここで挙げているのは、ショウジョウバエ胚の細胞辺縁部における DE-Cadherin と微小管マーカである tau の蛍光タンパク融合分子の共発現条件下で撮影した動画について、細胞質を辺縁に向かい投射する微小微小管の先端部と細胞表面に局在する DE-Cadherin との共在性について解析した例である (Fig. 1)。それぞれのマーカが放つ微小な蛍光を用いて細胞全体の形態を抽出し、その輪郭を抽出する。そして、得られた輪郭線に沿って円形の ROI (Region of Interest; 関心領域) を走らせ、領域内の平均輝度を都度計測することで細胞辺縁部における 2 つのマーカの輝度分布を調べる、という仕様である。

この種の計測は、プログラムの実装といった観点からはそれほど工数を要することなく実現可能ではあるがしかし、手動で、ことさら動画全てのフレームについて行うのは、かなり骨の折れる作業であり、輪郭線の抽出といった段階から音を上げたくなるような労苦となる。画像解析プログラミングと聞いて、画像認識のための新たなアルゴリズム開発のような事例を想像される向きも多いかと思われるが、ここで挙げたような単純な手続きを延々と繰り返すような事例でも、プログラミングが大きな力を発揮していることがわかる。

本稿では、生物画像データ解析について比較的初心者を対象に、オープンソースの生物画像データ解析基盤である ImageJ (FIJI) を用いた簡易なプログラミング法の導入について概説しつつ、より本格的な画像解析プログラムの実装技法と、得られたデータ解析の流れについて簡単に説明する。

画像を俯瞰する

観察画像を定量的に解析する上で、まずは定量する対象を確実に目視で捉えられるようにしておくことが望ま

しい。特に、共焦点レーザー顕微鏡のような光学切片像を積み重ねたような画像形式は確認に手間を要する。この種の積層画像は、観察対象が立体的であり、その配置などについて定量的な解析を施す場合には非常に有用なデータ形式ではあるが、あいにく表示装置が2次元に限定したディスプレイであるため観察像の全貌を俯瞰するにはやや扱い辛い。そのため、最大輝度投影（3次元のピクセル値のうち、Z軸上の値について最大値を求め2次元平面に投影する方法（Maximum Intensity Projection・MIP）が多用される。この時、投影する光学切片の範囲の調整など、その見かけについて調整を要する他に、この種の顕微鏡につきもののノイズはその出現確率を下げるため、投影前に除去しておく必要がある。単純な中央値フィルタ（注目画素とその近傍の画素値を輝度で並べ替え、その中央部を占める値を以って注目画素を置き換える手法。median filter）を適用するだけでもかなりの効果を認めることができる場合がある。また、観察対象の配置によっては、撮像面を用いた投影像が必ずしも最善の再構成法ではない場合もある。このような場合は、視点を断面に置き換えた上で再構成を行う必要が生じる（Fig. 2, 3a）。なお、これらの画像操作や可視化は、顕微鏡に付属するソフトウェアや、さきに挙げた ImageJ などのフリーソフトウェアによっても実現することが可能である。

画像解析ことはじめ

画像解析のためのプログラミングをまずは手っ取り早く習得するためにはどのような基盤が最適だろうか。実は、プログラム自体はそれほど難しいものではない。通常のプログラム言語は、1. 逐次、2. 分岐、そして3. 繰り返しといった3つの要素が最低限組み込まれている。これは、プログラムが書かれた順を追ってその内容を実行すること、条件によって処理を切り分けたり、あるいは処理の有無を決定すること、そして任意の処理を繰り返し実行することに相当する。言うなれば、「道中、赤信号なら止まり、青信号になれば進むを目的地に至るまで繰り返す」という手続きについて明文化したものであると言える。生物画像解析においても、このような手続きは画像を操作する機能の呼び出しの手順を定めたものとして記述可能である。オープンソースの生物画像解析として最もよく知られている ImageJ や、その派生物である Fiji といったアプリケーションは、マクロ言語と呼ばれる簡易プログラミング環境を提供しており、用途によってはかなり複雑な解析工程を実装することも可能である。マクロ言語とは、ImageJ 本体を操作すること

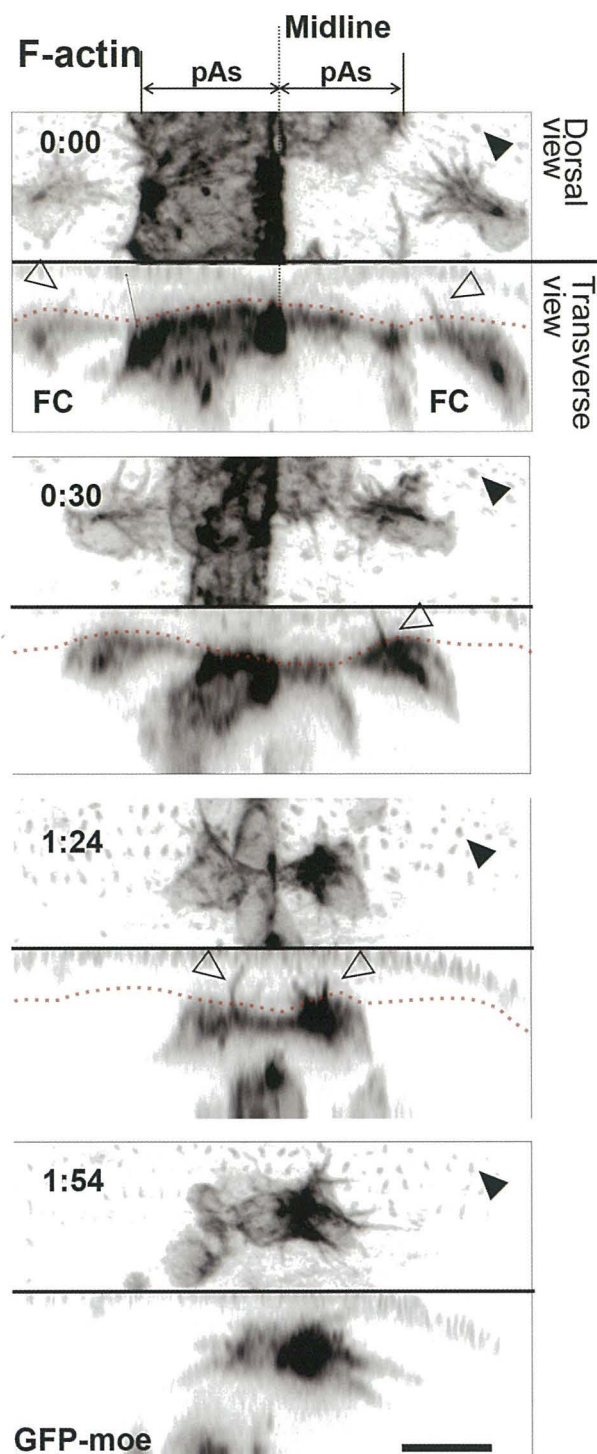


Fig. 2. 気管先端細胞の F-actin イメージング
気管先端細胞の移動について、撮影面からの俯瞰図 (XY-Z) ならびに横断面 (XZ-Y) の再構成像として示すことで、より仔細に細胞動態を捉えることができる。スケールバーは 10 μm 。

のみを目的として設計されたプログラム言語であり、マクロで記述したコードは ImageJ 本体によって逐次解釈され、実行される。余談ではあるが、画像を取り扱うような処理系ではこのようなプログラミング環境を同梱す

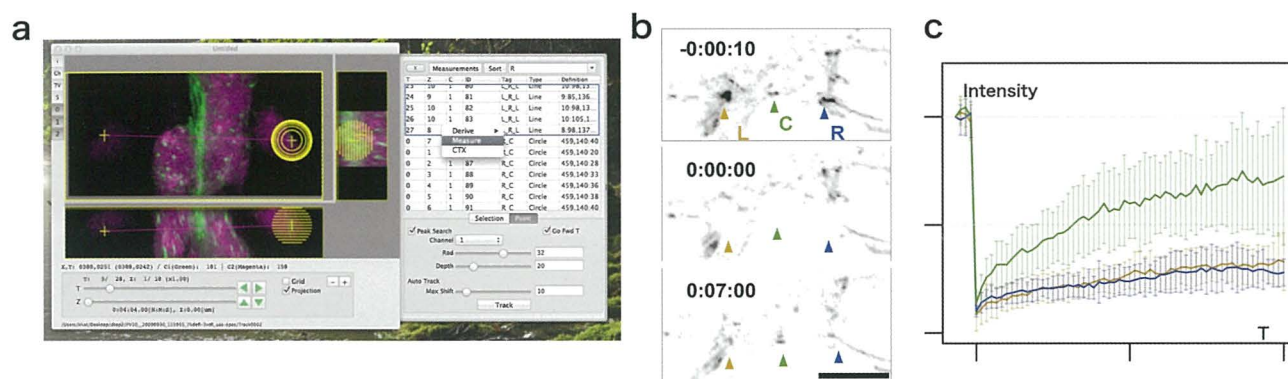


Fig. 3. 手作業による特徴抽出のための GUI アプリケーション

a. 横断面の再構成像の作成機能に加え、効率的に画像特徴を採取するための機能ならびにユーザインターフェイスを実装している。b. FRAP 実験。撮影開始時 (-0:00:10) に見られる 3 個の DE-Cadherin-GFP により標識した細胞間接着部位について、レーザーにより蛍光を褪色 (0:00:00)、その回復過程 (0:07:00) を開発したアプリケーションにより計測する。時間は (時:分:秒)。c. FRAP について実施した解析例。目視による特徴採取の手続きを最適化することにより、多数のサンプルについて同様の解析を行うことが可能となった。スケールバーは 10 μm 。

ることでユーザによる作業の自動化を図ることが多い。翻って考えると、画像処理自体、面倒かつ退屈な単純作業の繰り返しが多い、という事情を伺うことができる。ImageJ のマクロでは、そのプログラムの内容は、ImageJ のメニュー項目 (例えば File -> New -> Image... といったような一般的なソフトウェアに共通して存在するアプリケーションの構造) を、文字列として起こしたようなものが実行すべき順に羅列してあるような見立てとなっている。実際に、Plugins -> Macros -> Record... を選択すると、「Recorder」なるウィンドウが開き、以降 ImageJ 上におけるほぼ全ての操作が文字列としてウィンドウ上に列挙され、これを保存、実行するだけでプログラムとして動作する。この Macro Recorder は、開いた画像に対して、自分がどのような処理・操作を施したのかを後日確認するためのログとして利用するという用途で用いることも可能である。画像処理プログラムを実装する上で、ImageJ が持つ機能をそのまま全て利用できるという点でこれは非常に効率が良い。というのも、冒頭の例で挙げたような「任意の画像の読み込み」、「任意の座標にある特定の形状の ROI を置き」、「その領域内の輝度を計測する」といった工程全てについて、プログラムするには膨大な手間を要するためである。ImageJ マクロの文法は、一見すると C と類似しており、非常に単純である。特に、分岐や繰り返しといった制御構造などは C の記法と同様のものが採用されている。また、ImageJ の機能は関数の呼び出しとして実装する。

ただし、マクロは ImageJ に予め実装されている機能 (メニュー項目) を継ぎ接ぎするには非常に効率的な環境ではあるが、ピクセル毎の操作を施すような実装は実行速度の点でやや力不足であることは否めない。また、

解析の途上で得られたデータを保持するための高度な機構を持たない点も、大規模なアプリケーションを実装するにあたって障害となる。このような点を満足する基盤として、ImageJ ではプラグイン機構が提供されている。プラグインは、GUI (Graphical User Interface。画面をマウスで操作するあれやこれや) や様々なデータ表現方法を含む Java 環境が提供する豊富な機能を利用しつつ、高速に動作するプログラムを Java 言語により実装することが可能である。

プラグイン開発をはじめとする本格的なプログラミングを実施する際に、非常に参考となるのが ImageJ 自身のソースコードである。参照用に整理されたコードがウェブ上で公開されており (<https://imagej.nih.gov/ij/developer/source/>)、画像工学の教科書に掲載されているようなアルゴリズムの数式での表現が実際に機能するプログラムとして書き下されている様を見ることで効率的に学習を進めることができる。

より効率的な画像解析ソフトウェアの開発

画像解析では、画像を構成する個々のピクセルを相手として独自に定義した演算を行う事例が多く、その結果計算量が膨大となる場合がある。また、ImageJ などの汎用のソフトでは実現が難しいような操作性を必要とするソフトウェアを要する場合、独自にプログラムを実装する必要性が生じる。以下に、それらの事例について、参考文献 (Kato et al., 2016) をもとに紹介する。なお、ここで挙げた例は、実行効率の観点から C ならびに Objective-C を開発言語として用いている。

手作業による画像特徴取得ソフトウェアの実装

in vivo イメージングのような混み入った状況下での顕微鏡観察画像を対象とした計測は、目視による手作業に頼らざるを得ない場合がある。ここでは、DE-Cadherin-GFP の *in vivo* FRAP について紹介する。FRAP (Fluorescence Recovery After Photobleaching, 蛍光回復実験) は、レーザ照射により蛍光タンパクの蛍光をいったん褪色させた後、照射部位における蛍光の回復過程を測定することでタンパク質の動態を解析する手法である。この種の実験を導入することで、細胞間で接着を構成する部位ごとの Cadherin 分子の流入量の差を解析することが可能となる。ただ、この実験系では、蛍光タンパクを発現するショウジョウバエ胚生体を直接共焦点レーザ顕微鏡を用いて観察しているため、対象とする細胞が動くこと、そして目的外の細胞においても蛍光タンパクの発現が見られること、そしてそもそも褪色直後は計測対象部位の蛍光が大幅に減弱する。このため、測定対象を機械的に決定することができない。さらに、レーザーによる褪色処理を複数の領域に施すという実験の性質を併せ、ここでは手作業での特徴取得作業を最適化するアプリケーション開発を行った (Fig. 3a)。開発にあたって定めた主な仕様としては;

1. キーボードによる一文字入力によって Z・T (光学切片深度・時間) を含む空間を自在に移動できること。
2. 深度方向に向かって伸びた円筒状かつその深度と半径を任意の値に設定可能なマウスカーソルを定義し、その円筒内部において最も輝度の高い voxel (pixel に対し、深度を持つ画素を voxel と定義する) の座標を記録する機能を有すること。
3. 特徴 (ここでは褪色させた領域) の識別を確実に行うため、XY-Z ならびに XZ-Y, ZY-X の各方向に視点を置いた任意の範囲の MIP 投影画像を終始表示可能であること。

を設定した。

このような仕様を満たすアプリケーションを適用することにより、対象を確実に見極めつつもある程度 (円筒の範囲内で) 特徴を自動的に探索することが可能となり、たいへんな労力の削減となる。

このソフトウェア、仕様上は点の座標を記録するだけであるが、点を中心とした任意の径の球状の ROI を設定することで点を含む一定領域における平均輝度の測定を可能とする。また、複数の点を定義し、その間の距離の時間変化を計測することも可能となる。このように手

作業での測定はとにかく人手による入力を単純化し、かつ実測時により高次の特徴を派生させることで労力を低減しつつ効率化することで、より多くの観察事例を対象とした測定値を取得する工夫が必要であると考えられる (Fig. 3b)。

不明瞭な画像特徴の動態解析

細胞内の微小管の伸長の動態を可視化する方法として、その + プラス端に特異的に結合する EB1 と GFP との融合タンパクを発現させる実験法がある。細胞内の微小管の配置が疎であれば、個々の EB1 による輝点を追跡することで微小管の配向や伸長速度を計測することができるが、ショウジョウバエ気管の先端に位置する細胞では微小管は密に存在しているため、個別の輝点を連続した時間スライスで同定することが困難である。このような状況で細胞内の微小管の動きを解析するため、Particle Tracking Velocimetry (PTV) という手法を実装し、解析に供した (Fig. 4)。PTV は、動作原理としては非常に単純で、画像を多数の正方形領域 (基盤の目状) に分割し、各々の矩形領域における画素のパターンに特徴があれば、次の時点でその模様が何処に移動したか、周辺を探索する、という手続きを全ての領域について延々と繰り返すだけのものである。

このような手法は個々の輝点の軌跡を追跡する方法とは原理が異なるため、輝点の移動についてその始点と終点を知ることは難しいが、天気予報で観る風の流れ図のように観察対象を複数の区分けした各領域における「物質の流れの平均的な特徴」を知ることができる。このような手法は、密に配置する多量の物質の動態を解析する上で非常に強力な手法ではあるが、その一方で領域の移動先の探索には領域間の類似性の指標として正規化相互相関係数を用いているために計算量が非常に多い。ただ、この種の演算はひたすら同じことの繰り返しのため、並列実行することで効率化を図ることが可能である。

並列化の手法としては pthreads のような CPU のコアを対象としたものや、GPGPU のように GPU を利用したものがある。後者は、多数の画素を実時間で描画する必要があるディスプレイ表示用の補助演算装置を汎用計算に応用した事例が起源であり、現在では深層学習での応用事例が衆目を集めている。GPU を計算に用いる場合、nVidia 製の装置を対象とするのであれば CUDA C といったような専用のプログラミング環境を必要とする。また、GPU の内部構成に配慮した方が演算効率が良かったり、GPU へのデータの送受信の手間が必要だったりすることもあり、実装には非常に手間がかかるが、PTV の実

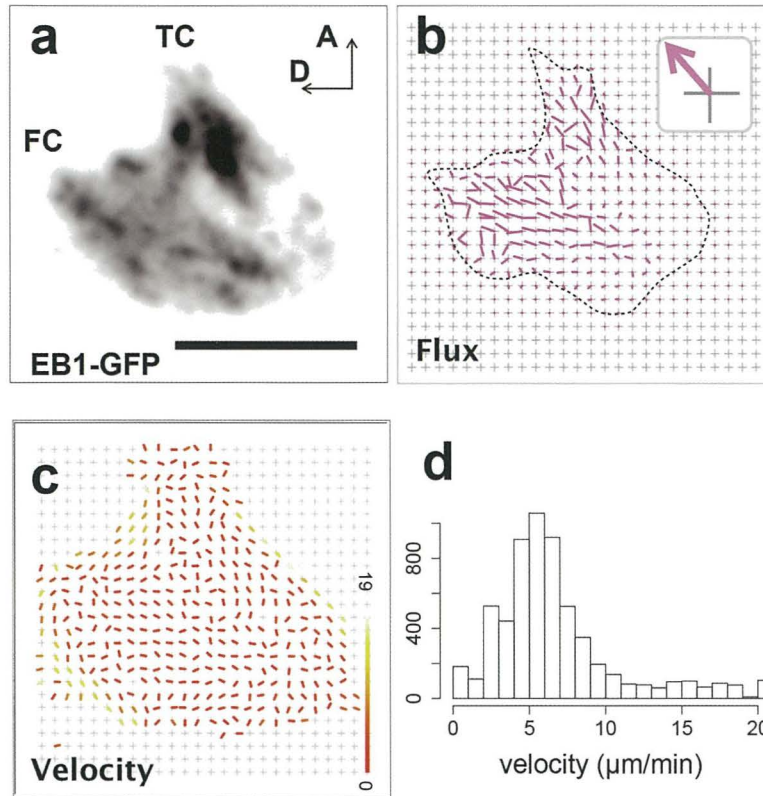


Fig. 4. 気管先端細胞における微小管伸長動態のPTV解析

a. 気管の先端部の1細胞で微小管+端結合因子EB1のGFP融合タンパク質を発現させ、共焦点レーザー顕微鏡によりタイムラプス観察したもの。b-c. 細胞内の各領域における相対的な流量 (b) と速度の分布 (c)。d. 全領域における速度の分布。スケールバーは10 μm。

装についてはCPU (Intel Core i7, 4コア8スレッド) に比しておよそ14倍程度の高速化を達成した (nVidia Tesla K20c)。さらなるチューニングもおそらく可能であるのだろうが、正直精魂尽き果てるような種の仕事となること、そして少々複雑な計算となるとさらに実装自体が困難となるため、現在は専らさきに挙げた pthreads を用いたCPU並列化による実装を実施している。pthreadsは、多種のOSを標準化するPOSIX規格に準拠した並列化プログラミングモデルのため、例えばMacとLinuxとの間でソースコードを書き換える必要のないプログラムを実装することも可能である。筆者はプログラム開発と試験をMacのXcodeで行い、ソースコードをIntel Xeonプロセッサを4基搭載したLinuxマシンに転送、ビルドを行い本番に供している。他にも、pthreadsに類似しているがさらに使い勝手の良いOpenMPといった並列化基盤を用いることでも、プログラムの効率的な実行が可能となる。

データ処理

画像特徴を数値に置き換えたデータの統計的処理などはエクセルなどの表計算ソフトやRに代表されるよう

な統計解析ソフトウェアを用いて行った方が堅実であると考えられる。単純に平均値を求めるだけのコードであっても、自身で実装した場合バグを紛れ込ませる余地は大きい。そのため、統計解析用の環境にデータを移行するべく、画像解析ソフトウェアが求めた値をCSVやTSV (Comma-もしくはTab-Separated Values) ファイルとして書き出したり、あるいはオープンソースとして広く普及しているファイル指向型リレーショナルデータベースであるSQLite3のファイルとして保存したりしている。後者については、他の一般的なデータベースシステム (PostgreSQL) などと異なり、データベースを構成する一切切をOSに依存しない形式の単一ファイルとして保存できるなど、小規模なデータの読み書きを行う基盤として非常に使い勝手が良い。また、SQL (Structured Query Language) と呼ばれるデータへの問い合わせ言語をサポートしているため、データセットに対して選択的なデータの読み込みや更改などを行うといった操作が可能である。これは、一般的な表計算ソフトウェアであれば、ある任意の条件を満たす行だけを表の中から選択的に読み込んだり他の表の値との組み合わせによる結果を取得し、処理した結果を新たに表に書き込む、といった

ような操作に相当する。前述の R においても、SQLite3 データベースへの読み書きをサポートしているため、画像解析ソフトウェアで考える全ての数値データをとりあえずデータベースに書き出しておいて、後に R で SQL を用いて必要なデータを読み込んで解析し、その結果をデータベースに書き出す、といったような使い方が可能である。SQLite3 は、C をはじめとする多くのプログラミング言語を対象としたインターフェイスが用意されており、元の観察画像に対して R で処理した結果を可視化するというような作業についても、効率よく行うことが可能である。

数値処理の結果は、表計算ソフトや R などのプロットとして最終的に表現されることとなるが、その段階に進む前に、何らかの方法を用いて必ず計測対象となった画像上に重ねて表示し、測定などの処理に異常が生じていないことを確認する必要がある。画像特徴は、数値化した途端に一人歩きを始める。数値そのものを眺めていても、何らかの誤りを内包していることに気付くのは非常に難しい。測定すべき対象の違い、ソフトウェアの不具合など、異常が生じる要因はそこかしこに存在するため、常に可視化による確認を行うよう心がけることが重要となる。

おわりに

自戒の念を込めて。ソフトウェアを開発し、定量的な解析に供することは労力を低減し、かつ実装によっては測定の客観性を向上させる。が、山と積み上がった観察画像を対象とした手作業も、コツコツとこなせばいつか完遂する。開発したソフトウェアを適用できれば数分で終わるであろう解析も、その実装期間が「気合と根性」による手作業に要する時間を大幅に上回るようではその有り難味は霧散する。したがって、開発難度、実装期間

の見積もりと、測定対象がどの程度の量的規模で、機械的な処理に適しているのかについてその塩梅を慎重に見積もっておく必要がある。

以上、駆け足ではあったが、生物画像を解析する方法ならびに過程について概説した。画像解析においてここで解説しなかった大事な要素として解析対象となる画像の品質がある。自然科学研究機構・生命創成探究センターならびに基礎生物学研究所では、解析に適した画像データを取得する方法や、本稿で挙げたマクロによるプログラミング技法について実践的に習得する場として例年「生物画像データ解析トレーニングコース」を開催している。詳細はコースウェブサイト (<https://bioimageanalysis.jp/>) に掲載している。また、本稿の内容に則した雑誌記事について、参考文献として付記した。

引用文献

Kato K, Dong B, Wada H, Tanaka-Matakatsu M, Yagi Y, Hayashi S (2016) Microtubule-dependent balanced cell contraction and luminal-matrix modification accelerate epithelial tube fusion. *Nat Commun* 7, 11141. doi: 10.1038/ncomms11141.

参考文献

Kato K, Hayashi S (2008) Practical guide of live imaging for developmental biologists. *Dev Growth Differ* 50, 381-390. doi: 10.1111/j.1440-169X.2008.01029.x.
加藤輝 (2018) クローズアップ実験法. 生物画像処理・解析を加速する ImageJ マクロプログラミング. *実験医学* 36 (9), 1513-1520.
加藤輝 (2019) イメージングデータ解析よろず相談所. 大量画像解析のための自動化手法・応用編. *実験医学* 37(14), 2346-2353.
小山宏史, 加藤輝 (2019) イメージングデータ解析よろず相談所. 顕微観察画像の定量的情報解析とその落とし穴. *実験医学* 37 (16), 2772-2777.