



確認テストの解答

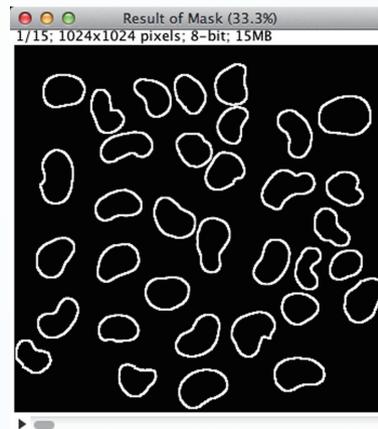
【問題①】 連載第7回で、核膜タンパク質の輝度の測定を行った。このときに使ったサンプル[NPC1.tif]は、時系列であり、核膜タンパク質が核膜に結合していく過程をとらえたものである。連載第7回の方法を時系列に拡張することで、輝度の変化を測定することができる。今回の連載での知見と合わせて、測定し、解析・考察せよ。

【答え】

次のような手順で測定を行うことができる。基本的には連載第7回の手順をスタックで行えばよい。

1. **[NPC1.tif]**を開く。この動画は、サンプル画像のプラグインをインストールしていればメニューから開くことができる。もしできない場合は、サポートサイトの連載第11回のページからダウンロードしてほしい (<https://sites.google.com/site/imagejpp/articles/session11>)。
2. **[Split Channels]**でチャンネルを分割し、核膜タンパク質のチャンネル (Channel 1, 緑) と、核のシグナルのチャンネル (Channel 2, 赤) の2つのスタックにする。
3. まず、核のチャンネルで作業する。
 1. **[Plot Z-axis Profile]**で、蛍光褪色の様子を確認する。 **[Plot Z-axis Profile]**を使うため、まず**[Image > Properties...]**を実行、Slicesの項を15に、Framesを1に差し替える。元は逆になっており、それが正しいのだが、あくまでも輝度の変化を得るためにこのようにする。この動画では褪色はとても少ない (ピクセル値1以下) なので、とりあえず補正をする必要はない、と判断する。後に二値化がうまくいかないようであればあらためて考える。
 2. **[Gaussian Blur...]** (sigma=1) で若干ぼかし、ノイズを取り除く。スタックの処理の場合、コマンドを実行した後で“Process all [n] images?...”と聞いてくる。スタックの画像すべてを処理するかどうか、という質問である。OKをクリックする。以下のステップでも同様。
 3. **[Auto Threshold]** Liのアルゴリズムで二値化する。核以外にも小さなゴミが分節化されてしまっている。また、核にも小さな穴が空いている。以下の2ステップでこれらを取り除く処理をする。“White objects on black background”と“Stack”にチェック。
 4. **[Process > Binary > Open]**で小さなオブジェクトを取り除く。
 5. **[Process > Binary > Options...]**でBlack Backgroundをチェック。 **[Fill Holes]**でオブジェクトに空いている穴を埋める。
 6. **[Duplicate...]**で複製する。“Duplicate Stack”にチェック。
 7. 複製スタックに3回**[Erode]**処理
 8. 複製元スタックに3回**[Dilate]**処理
 9. **[Image Calculator...]**で複製元から複製スタックを引く。出力は緑の分節化画像 (図1)。

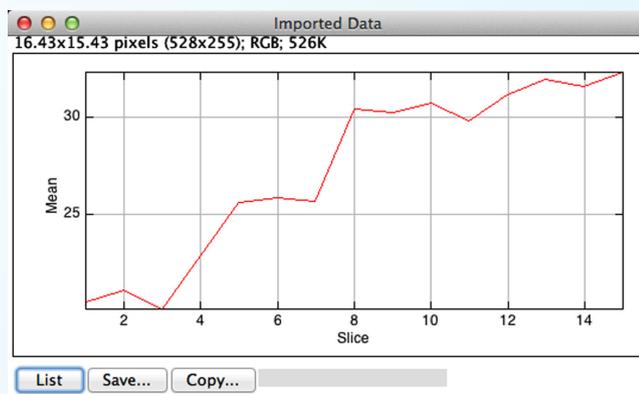
10. [Create Selection]続いて[Make Inverse]で、縁を選択領域 (ROI) にする。
11. [Add to Manager]で、選択領域をROI Managerに登録する。ROI Managerのウィンドウが開く。
12. [Select None]で選択領域を消去。
13. [Next Slice]ないしは“>”のキーで次のフレームに移行。
14. 10~13のステップをすべてのフレームで繰り返す。手で作業するのは大変なので、コマンドレコーダでコマンドを記録、Script Editorにペーストして“Run”を実行すると速いだろう。



■図1

4. 核膜タンパク質のチャンネルのスタック (緑)に戻ろう。まず、蛍光褪色の様子をチェックする。[Plot Z-axis Profile]を使うため、[Image > Properties]を実行、Slicesの項を15に、Framesを1に差し替えてから[Plot Z-axis Profile]を行う。
5. 蛍光強度の変化を眺めると、19近くから15近くまで下がっている。[Histogram]で背景の輝度を確かめると、9である。この変化はかなり大きいので補正を行う。
6. [Bleach Correction]で、背景の輝度を9と入力。補正したスタックで以下の測定を行う。
7. [Set Measurements...]で、測定項目を設定。Mean, Min & Max, Integrated Densityなど。Stack Positionをチェックすることを忘れない。
8. [Clear Results]で、これまでの測定結果をクリアし、Resultsの表を空にする。
9. 核膜タンパク質のチャンネルのスタック (緑)に戻り、ROI Managerにリストされた選択領域をシフトキーを押しながらすべて選択し (あるいはctrl-A, OSXではcmd-A)、“Measure”のボタンをクリックする。

10. Resultsの表をチェックしてみよう。“Mean”の列を見ると、平均輝度が増えていく様子がわかるはずである。



■図2

11. 今回の本文でインストールしたプラグイン“Plot Results”でx軸が“Slice”, y軸が“Mean”のプロットを行う。図2にあるように、核膜タンパク質が結合する過程を定量することができた。同じ測定を核膜の選択領域ではなく、例えば核の内部に当たる部分でも測定

し、逆に減る様子を定量化することもできる。余力のある方はこの測定を行い、Rなどで2つの領域の輝度変化をプロットしてみよう。

【問題②】①の解析をマクロで自動的に行えるようにせよ。

【答え】

コマンド・レコーダーから直接マクロを生成すると以下ようになる。ただし、二値化画像から選択領域をROI Managerに登録する部分は冗長なのでforを使ったループに差し替えてある。また、最後の[Plot Results]を使った部分はエラーを吐くことがあるので、コメントアウトした。

```
run("Split Channels");
run("Gaussian Blur...", "sigma=1 stack");

//selectWindow("C2-NPC1.tif");
//selectWindow("C1-NPC1.tif");
run("Auto Threshold", "method=Li white stack use_stack_histogram");
run("Open", "stack");
run("Fill Holes", "stack");
run("Duplicate...", "title=C2-NPC1-1.tif duplicate range=1-15");
run("Erode", "stack");
run("Erode", "stack");
run("Erode", "stack");
selectWindow("C2-NPC1.tif");
run("Dilate", "stack");
run("Dilate", "stack");
run("Dilate", "stack");
imageCalculator("Subtract create stack", "C2-NPC1.tif","C2-NPC1-1.tif");
for (i = 0; i < nSlices; i++){
    run("Create Selection");
    run("Make Inverse");
    roiManager("Add");
    run("Select None");
    run("Next Slice [>]");
}
selectWindow("C1-NPC1.tif");
run("Bleach Correction", "correction=[Simple Ratio] background=9");
run("Set Measurements...",
"area mean standard min integrated stack redirect=None decimal=5");
run("Clear Results");
roiManager("Select", newArray(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14));
```

```
roiManager("Measure");
//run("Plot Results", "plot=[Imported Data] x-values=Slice y-values=Mean style=Line color=Red gridlines gridlines labels labels major major");
```

このコードは<http://goo.gl/3gTkkg>にあるので、Webからコピーしてほしい。

さて、このままでもよいのだが、多数の画像が登場してどの画像に対して処理を行っているのかわかりにくい。また、新規に作成される画像も多く、自動的に閉じるほうが結果を見やすい。getImageID()と、selectImage(ID)のマクロコマンドを利用してより安定な作動をするコードにしたのが以下である。また、数値形態演算による核の縁の分節化をforループに差し替え、さらにスタックの計測の際のフレームの指定をsetSlice(n)コマンドを使うことで、より安定に作動するようにした。

```
orgtitle = getTitle();
run("Split Channels");
selectWindow("C1-"+orgtitle);
clid = getImageID();
selectWindow("C2-"+orgtitle);
c2id = getImageID();

selectImage(c2id);
run("Gaussian Blur...", "sigma=1 stack");
run("Auto Threshold", "method=Li white stack use_stack_histogram");
run("Open", "stack");
run("Fill Holes", "stack");
run("Duplicate...", "title=C2-NPC1-1.tif duplicate range=1-15");
c2erodeid = getImageID();
for (i = 0; i < 3; i++)
    run("Erode", "stack");
selectImage(c2id);
for (i = 0; i < 3; i++)
    run("Dilate", "stack");
imageCalculator("Subtract create stack", c2id, c2erodeid);
edgeid = getImageID();
selectImage(c2id);
close();
selectImage(c2erodeid);
```

```
close();
selectImage(edgeid);
for (i = 0; i < nSlices; i++){
    setSlice(i + 1);
    run("Create Selection");
    run("Make Inverse");
    roiManager("Add");
    run("Select None");
}
selectImage(edgeid);
close();
selectImage(clid);
run("Bleach Correction", "correction=[Simple Ratio] background=9");
correctedid = getImageID();
selectImage(clid);
close();
run("Set Measurements...",
"area mean standard min integrated stack redirect=None decimal=5");
run("Clear Results");
selectImage(correctedid);
roiManager("Select", newArray(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14));
roiManager("Measure");
//run("Plot Results", "plot=[Imported Data] x-values=Slice y-v
alues=Mean style=Line color=Red gridlines gridlines labels lab
els major major");
```

この最終的なコードは<http://goo.gl/4GhaBB>にある。