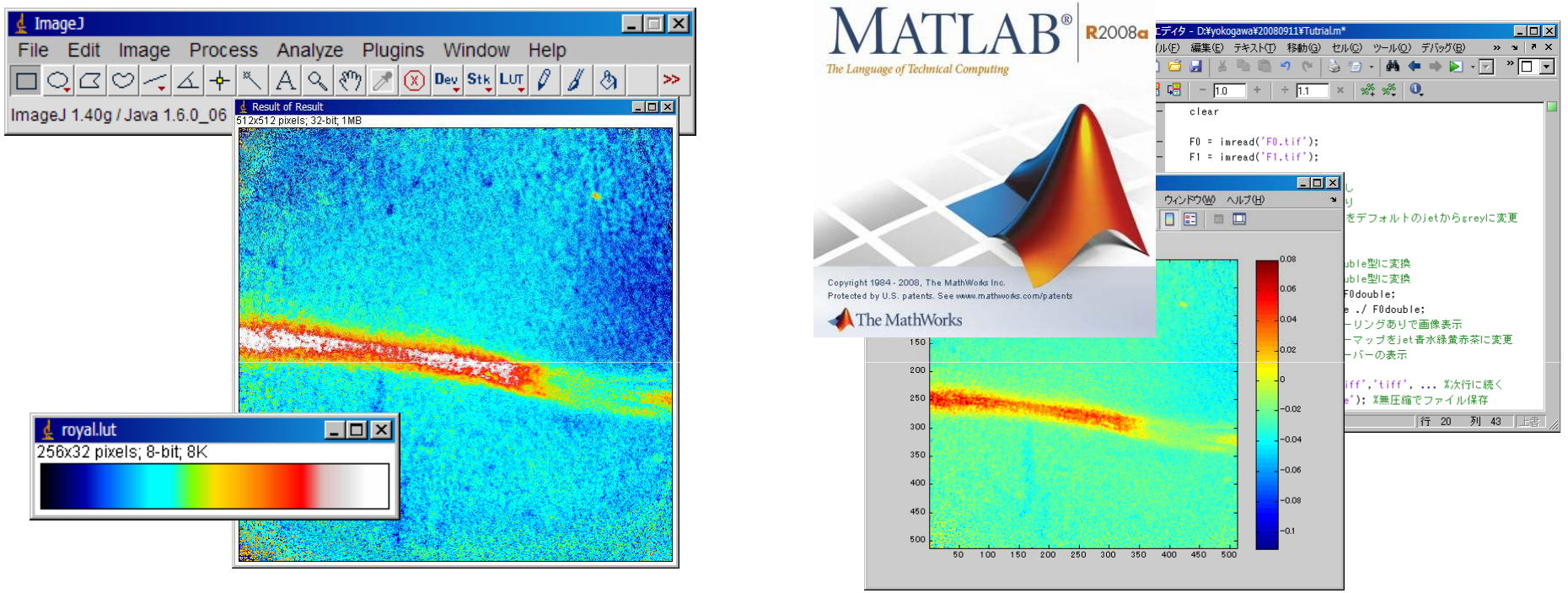
ImageJとMATLABを用いた定量的画像解析チュートリアル



大阪バイオサイエンス研究所

所長研究部 特別研究員 土居智和

tomdoi@obi.or.jp

10 January 2009

27 Dec 2008版

画像取得と保存時の注意

まずは綺麗なチャンピョンデータを得ないと解析方針が立たない

スケールバーの情報は失われるので、レンズ倍率はメモしておく。

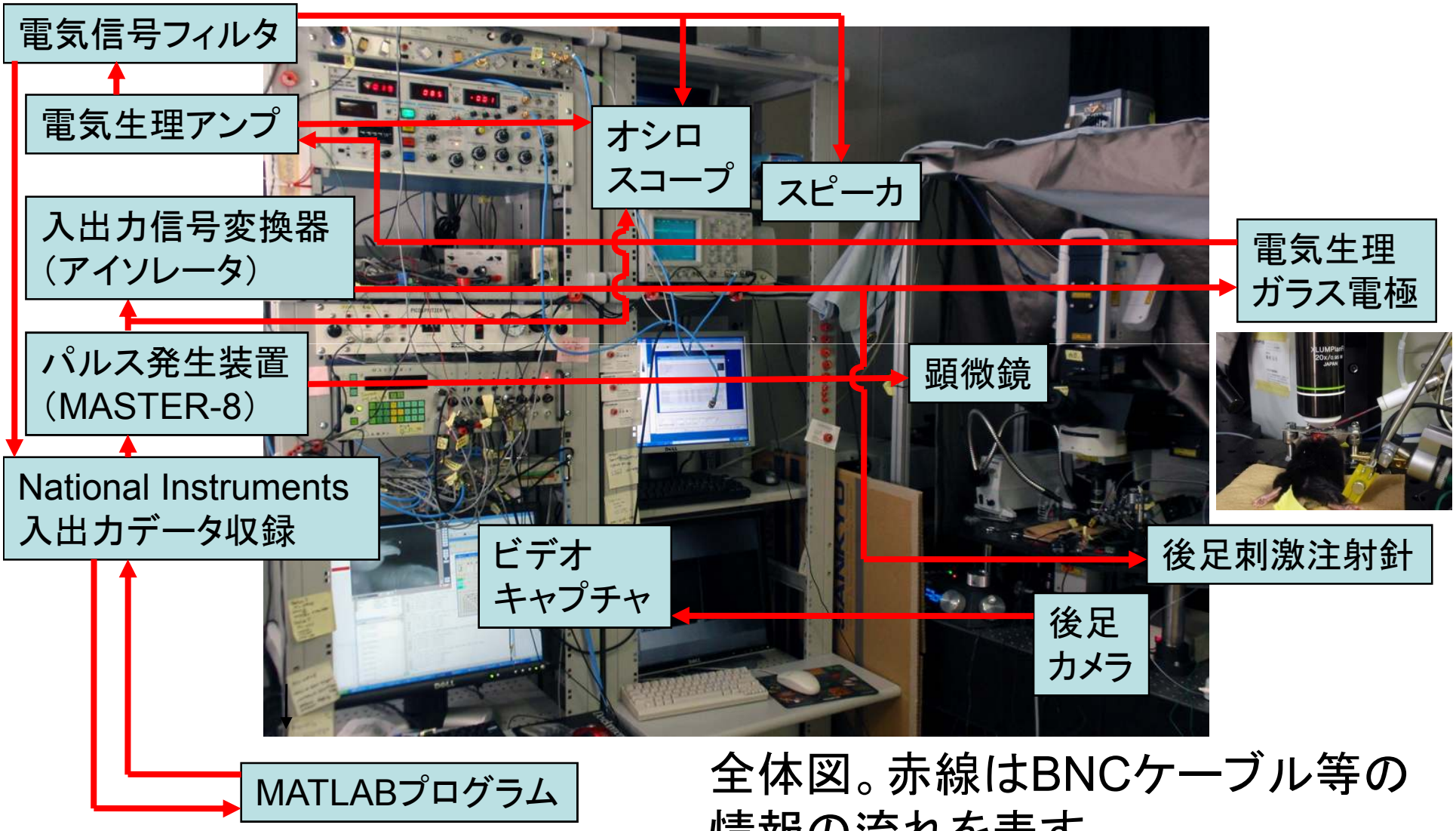
画像ファイルは、無圧縮(TIFF、BMP)または可逆形式(PNG)で保存。
不可逆圧縮(JPEG、GIF)は定量性が失われる。おすすめはTIFF。

時系列データは1枚1枚のタイムスタンプが正確に記録されるか事前に確認。全速力モードでは取りこぼし、タイムスタンプ詐称の可能性あり。

他の実験装置とタイミングを合わせるときは、
BNCケーブルをつないでTTL信号（トリガ）が
来たらスキャン開始するように顕微鏡を設定

BNCケーブル

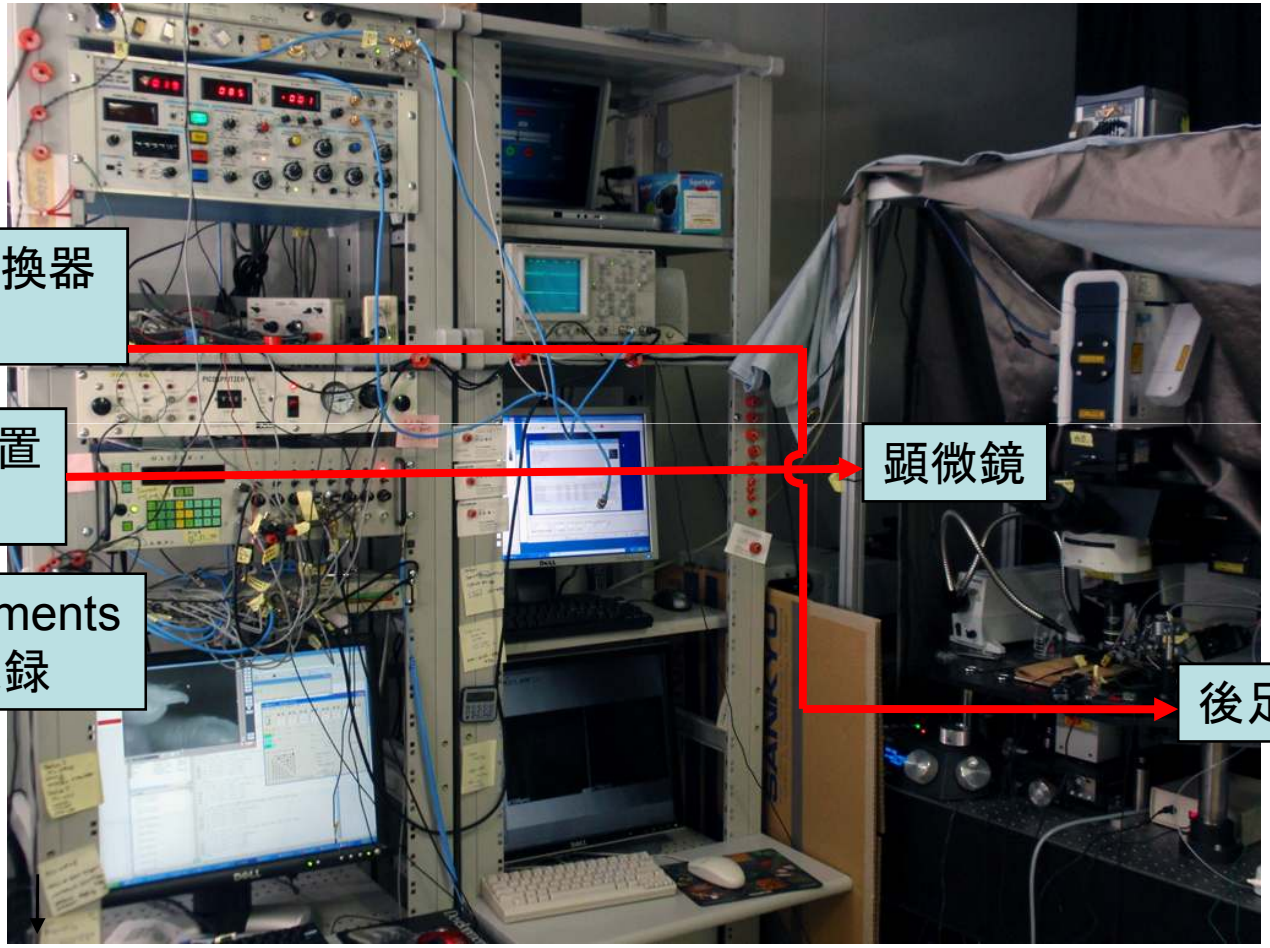
実際の画像取得セットアップ



全体図。赤線はBNCケーブル等の情報の流れを表す。

実際の画像取得セットアップ

マウス後足に電気ショックを与えながら、
脳細胞Ca2+蛍光経時変化と電気活動を同時記録



入出力信号変換器
（アイソレータ）

パルス発生装置
（MASTER-8）

National Instruments
入出力データ収録

MATLABプログラム

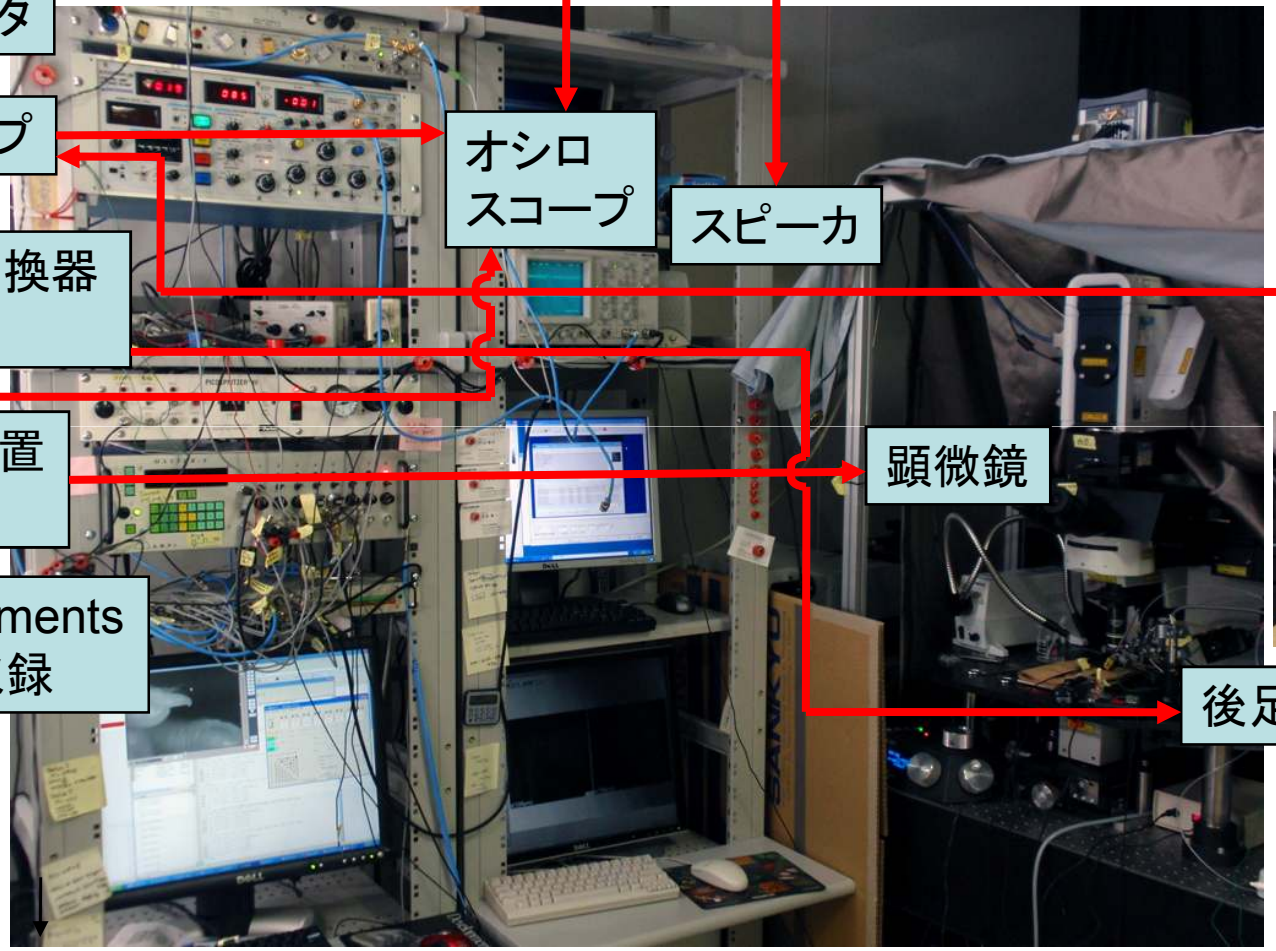
顕微鏡

後足刺激注射針

MATLABプログラムがタイムスケジュールを作り、トリガを顕微鏡と刺激装置に送る

実際の画像取得セットアップ

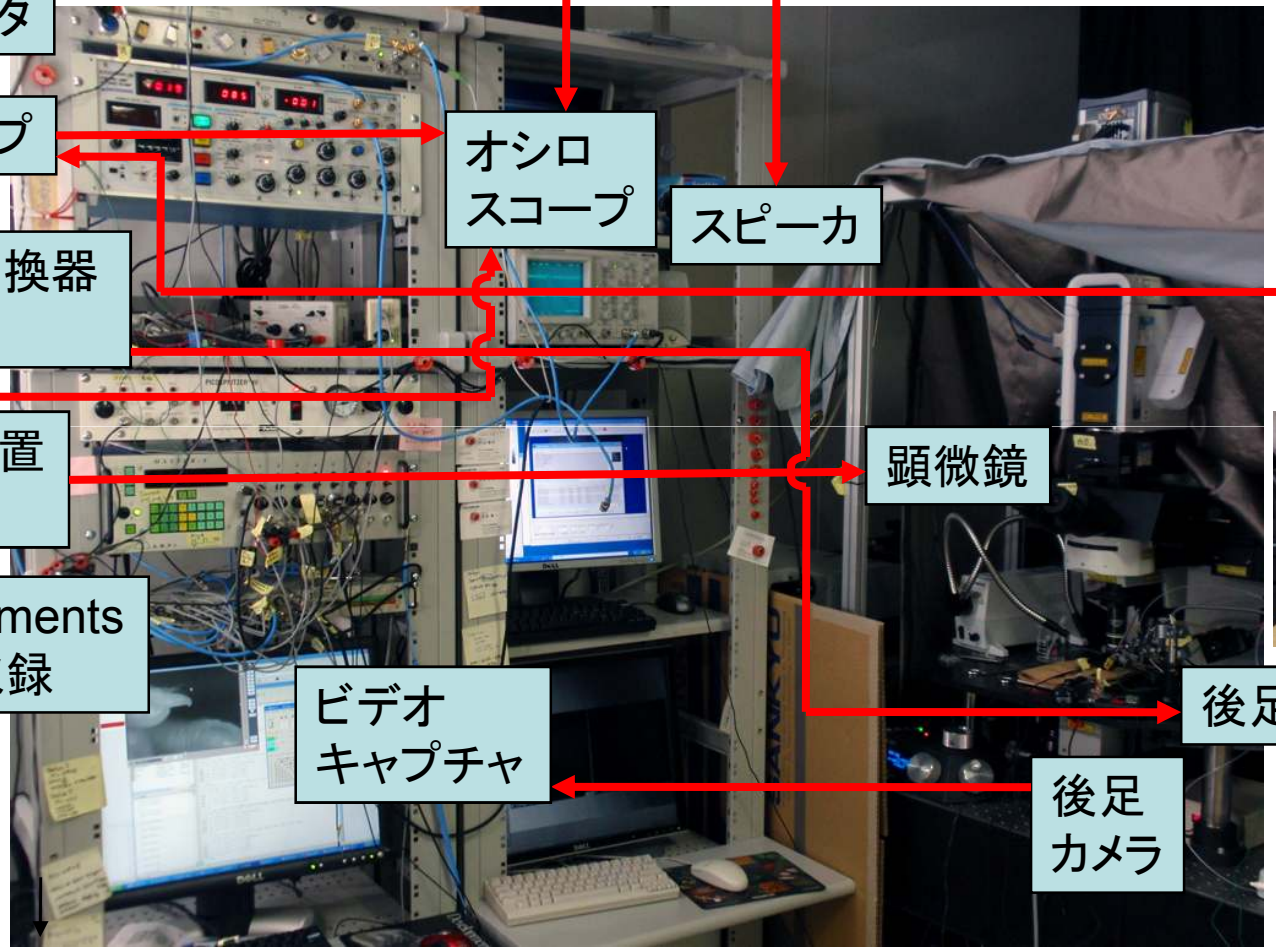
マウス後足に電気ショックを与えながら、
脳細胞Ca2+蛍光経時変化と電気活動を同時記録



顕微鏡は画像取得。電気生理データはNI
データ収録ボード経由でMATLABに入る

実際の画像取得セットアップ

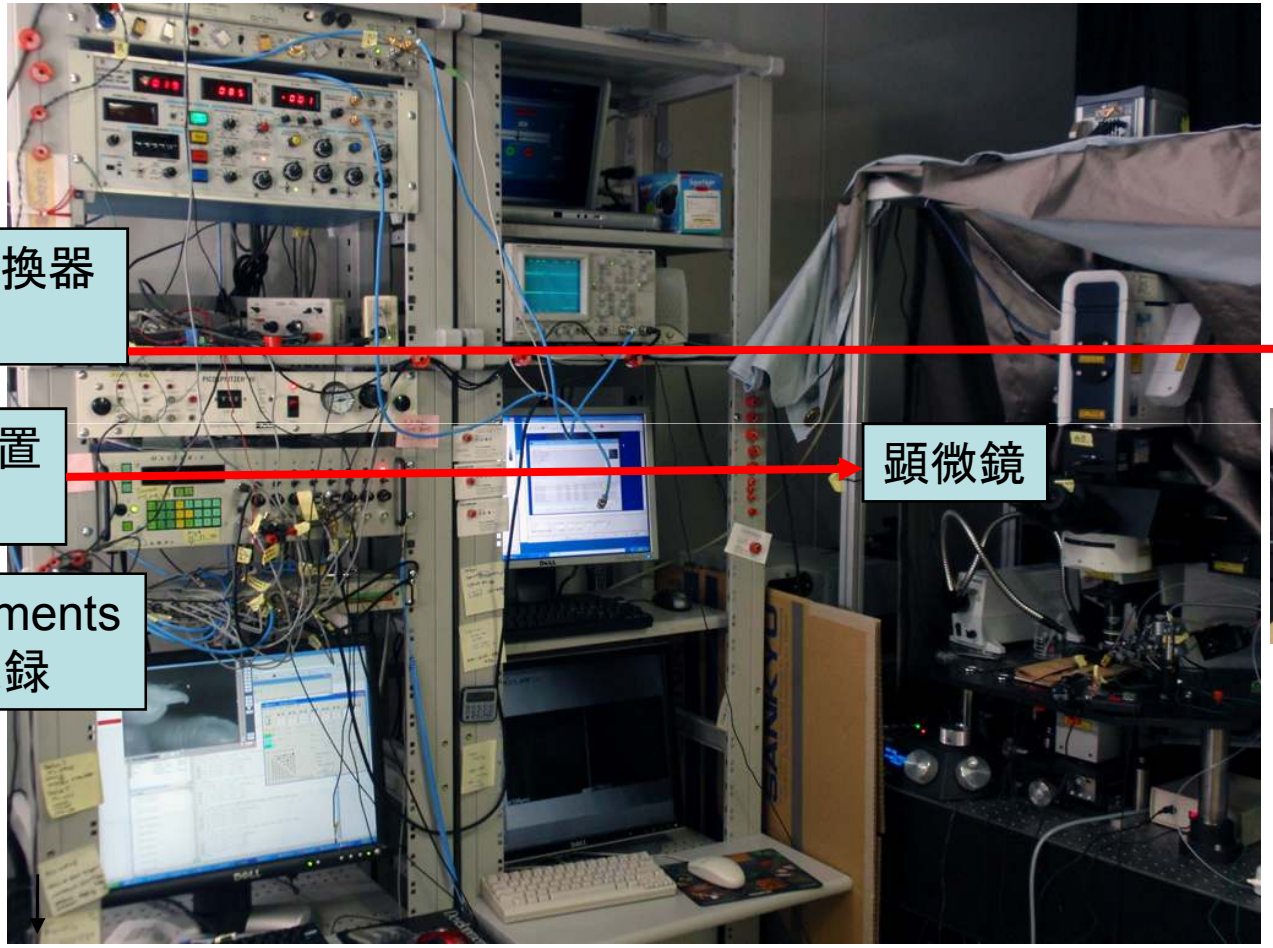
マウス後足に電気ショックを与えながら、
脳細胞Ca2+蛍光経時変化と電気活動を同時記録



後足カメラが後足刺激反射を撮影。
USBビデオキャプチャで保存

実際の画像取得セットアップ

皮質を電気刺激しながら、神経細胞Ca2+蛍光経時変化を記録

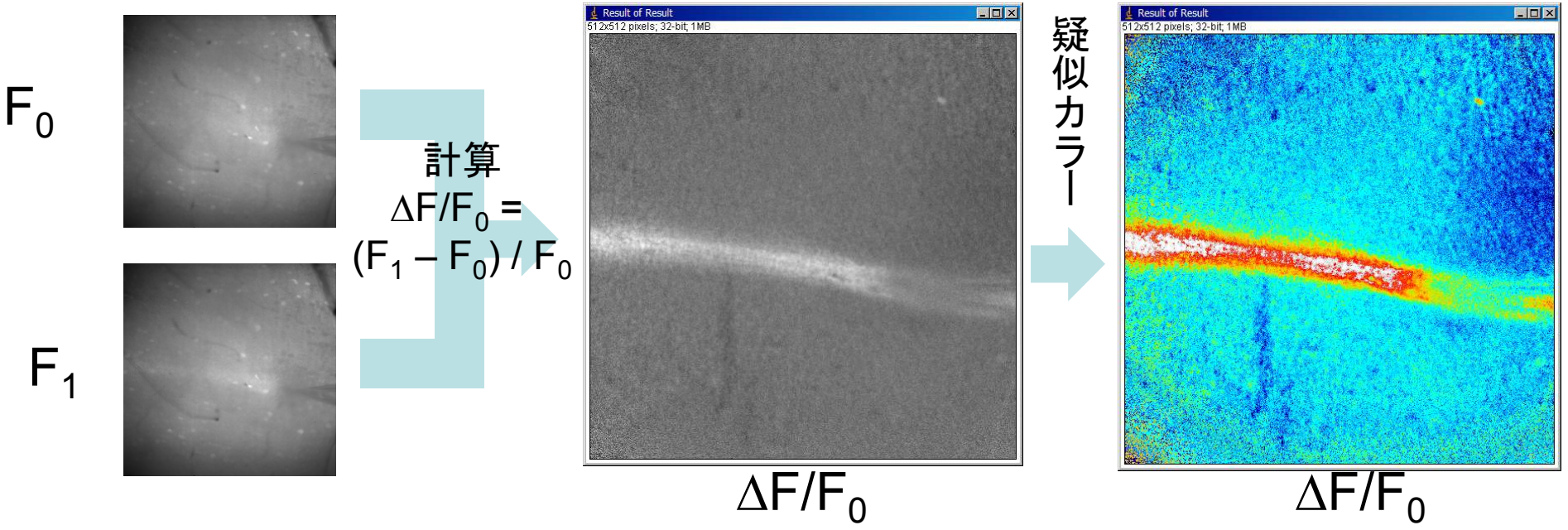


MATLABプログラムが刺激タイミングを作成。脳内ガラス電極から電気刺激。

チュートリアルの課題

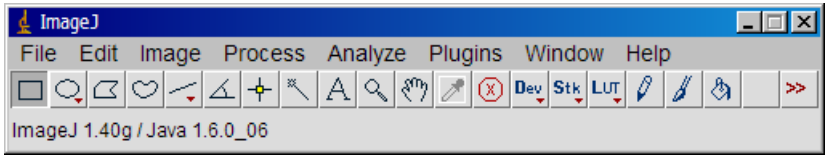
刺激を与えると、細胞の蛍光が光った。
刺激前と刺激後の細胞の画像データから
蛍光変化率($\Delta F/F_0$)を計算して、疑似カラーで表示したい。

$$\Delta F/F_0 = (F_1 - F_0) / F_0$$
変化率 = （後 – 前）/ 前

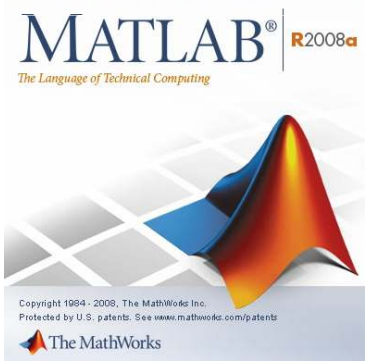


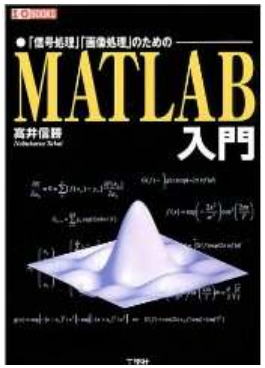
ImageJ と MATLAB

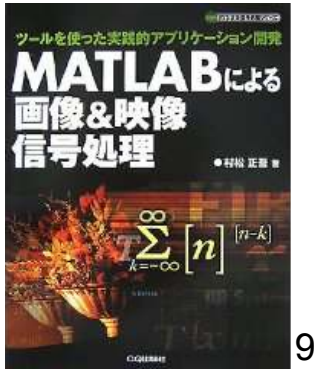
ImageJ (前身はNIH Image)
無料
メニューから機能を選ぶ
誰かが公開した新機能をPlug-inで追加できる
操作はマウス主体で易しいが、複雑な計算や自動処理は苦手。
参考書:医学・ライフサイエンス画像解析テキスト
　　改訂第3版 小島清嗣・岡本洋一 羊土社



MATLAB
有料。毎年契約。大学・研究所単位で契約しているかも。
プログラムを書く。欲しい機能はだいたい用意されている。
画像解析にはオプションのImage Processing Toolbox必須。
他にもSignal Processing Toolbox、Statistic Toolboxは欲しい。
OctaveやScilabは無料MATLAB互換ソフトだが、機能は限定・保証なし？
プログラムを書くのは難しいが、計算・自動処理・fig作成は得意。
参考書（初級）: MATLAB入門「信号処理」「画像処理」のための
　　高井信勝 I/O Books
参考書（上級）: MATLABによる画像＆映像信号処理
　　村松正吾 CQ出版社

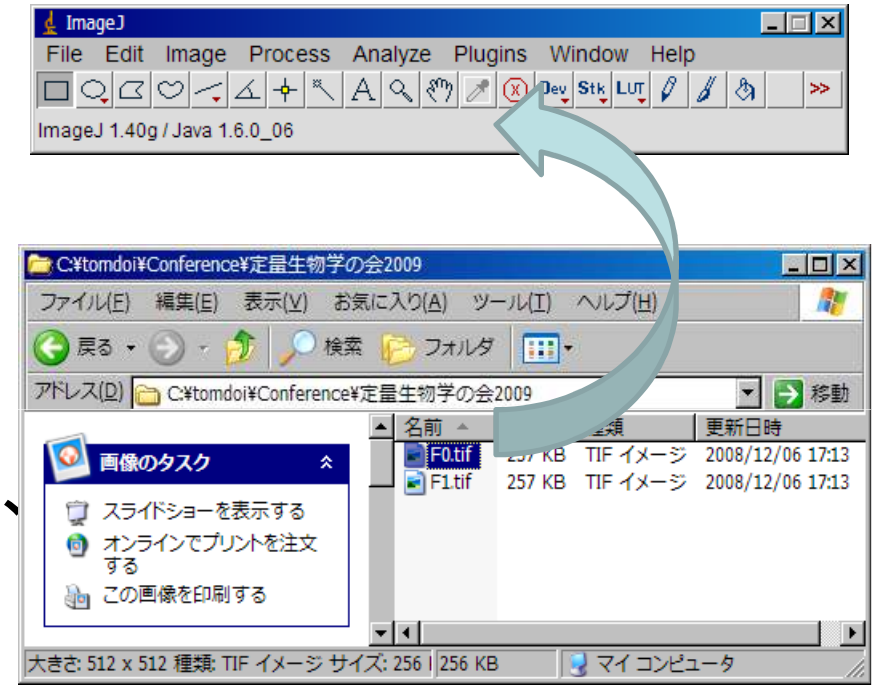






ImageJ で画像ファイルの読込

ファイルをImageJメニューバーに
ドラッグ &ドロップ
読込成功すると画像が勝手に
表示される
生物系顕微鏡のファイル形式を網羅し
ているので確実に読込できるはず

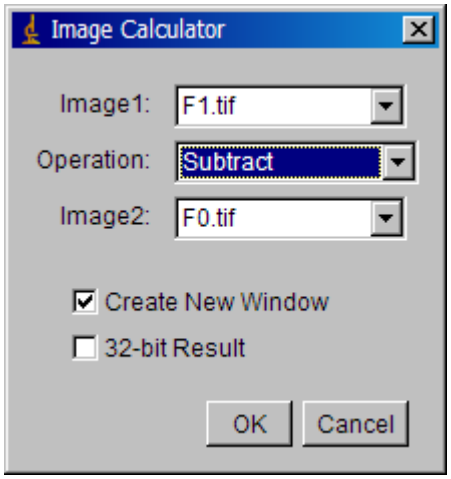


XYZやXYTなどの複数画像ファイルは、
File – Import – Image Sequence…で
一気に取込。
スタックで複数画像を1ファイルに
まとめると操作・ファイル管理が楽。

画像が読めない場合は、顕微鏡ソフトで画像形式を替えて保存し直す。

刺激後 ― 刺激前 の引き算(uint8)

Process – Image Calculatorで、
Image1: F1.tif
Operation: Subtract
Image2: F0.tif

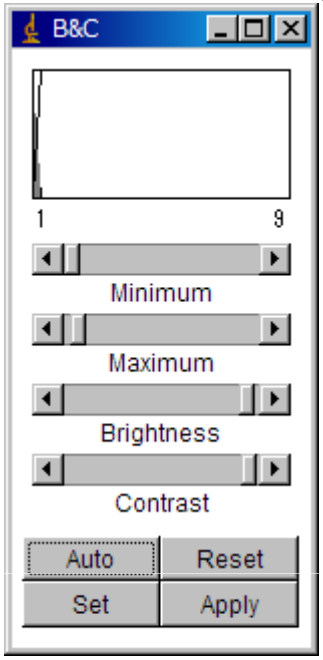


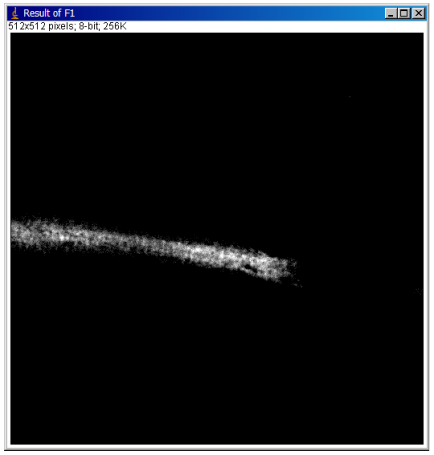
$\Delta F = F_1 - F_0$の値は小さいので、
計算結果の画像は暗い。

計算結果は元画像と同じuint8型で、
uint8型は負値を持たないので、
F1 ≥ F0 のとき、F1 - F0
F1 < F0 のとき、　0
となってしまう。
ちなみに、Operation: Differenceだと、
|F1-F0|のように差分の絶対値になる。

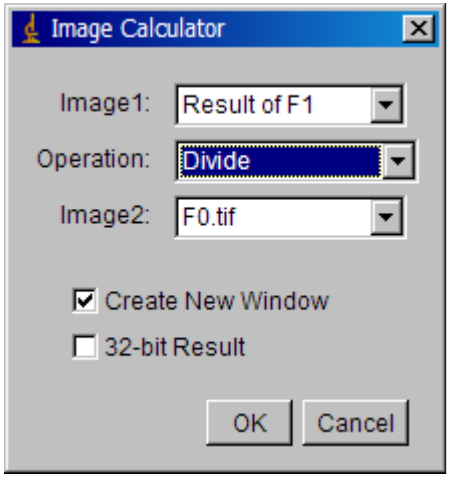
蛍光変化率($\Delta F/F_0$)の計算(uint8)

Image – Adjust – Brightness/Contrast…
とりあえず、Auto – Applyを試してみる
気に入らなければEdit – Undoで元に戻す

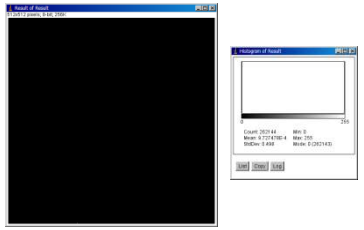




Process – Image Calculatorで、
Image1: Result of F1
Operation: Divide
Image2: F0.tif



画像が真っ暗に！ データも0しか入ってない！
どの点も蛍光変化率($\Delta F/F_0$) < 100% = 1だったので、
小数点以下の値が0に切り捨て扱いに。失敗。

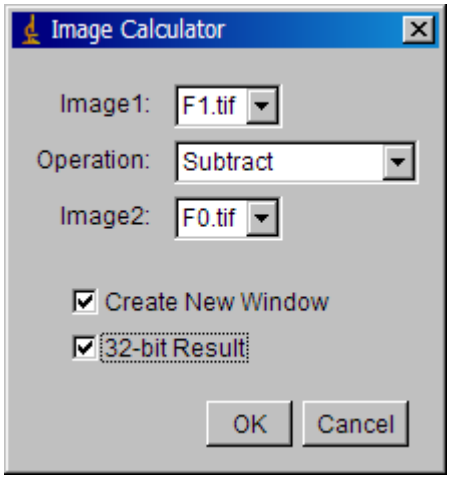


uint8 = unsigned integer 8-bit
符号なし　整数　8ビット
(負値なし)(小数点切捨)(2^8=256階調)

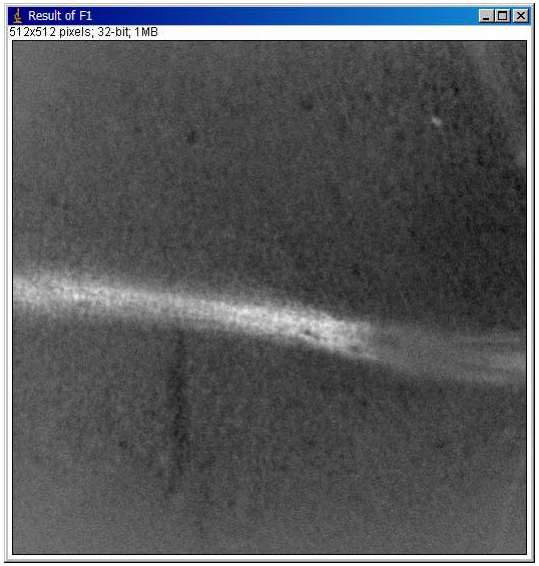
刺激後 ― 刺激前 の引き算(32-bit)

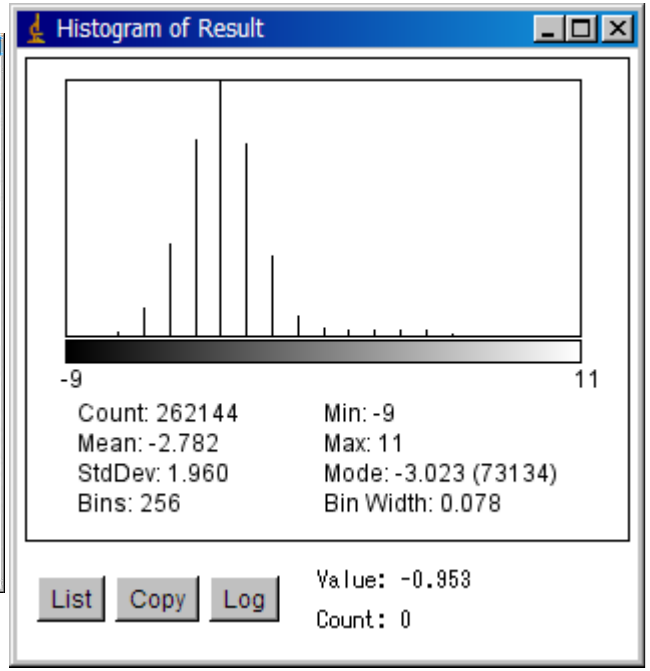
Process – Image Calculatorで、
Image1: F1.tif
Operation: Subtract
Image2: F0.tif

32-bit Resultにチェックをつける



Analyze – Histogramで
ちゃんと負値もある

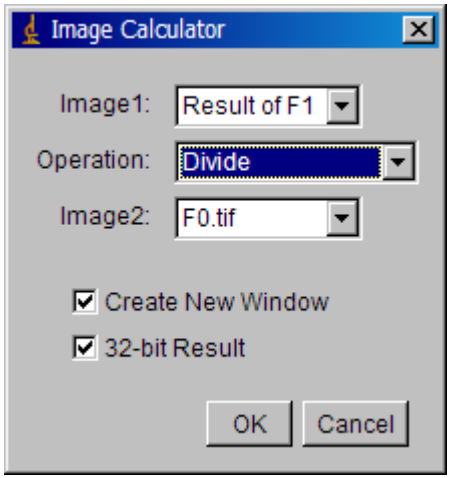




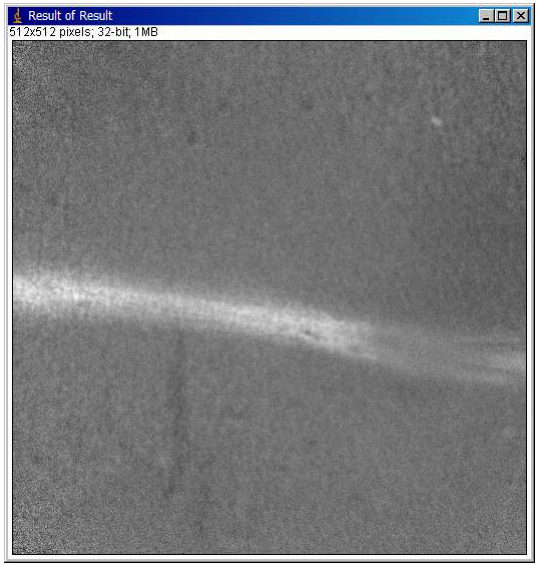
蛍光変化率($\Delta F/F_0$)の計算(32-bit)

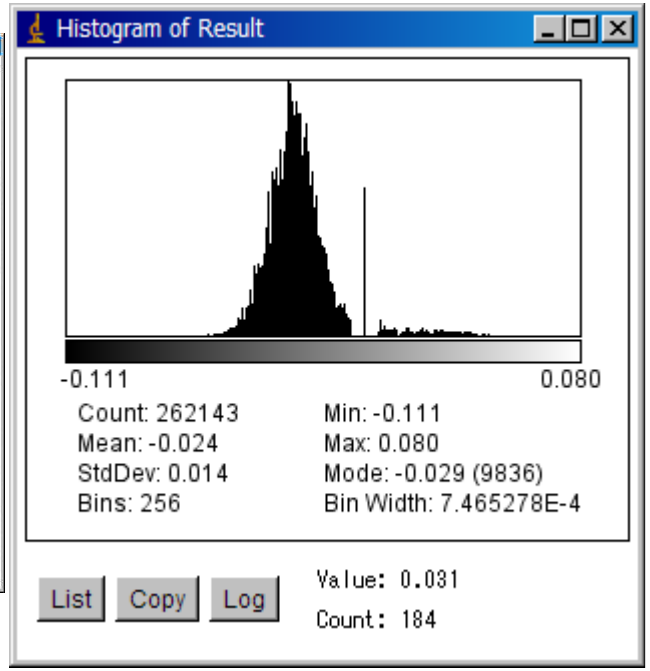
Process – Image Calculatorで、
Image1: Result of F1
Operation: Divide
Image2: F0.tif

32-bit Resultにチェックをつける



Analyze – Histogramで
ちゃんと小数点以下もある





コントラスト補正と疑似カラー(32-bit)

Image – Adjust – Brightness/Contrast…
とりあえず、Auto – Applyを試してみる
気に入らなければEdit – Undoで元に戻す

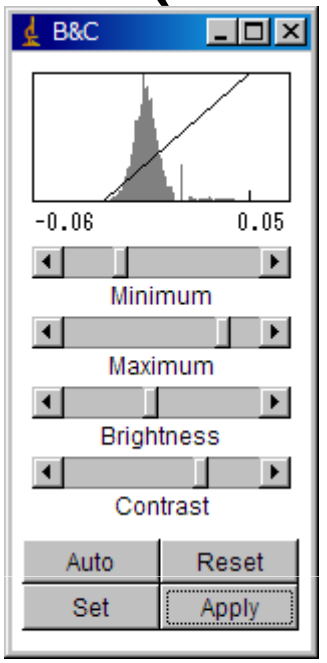
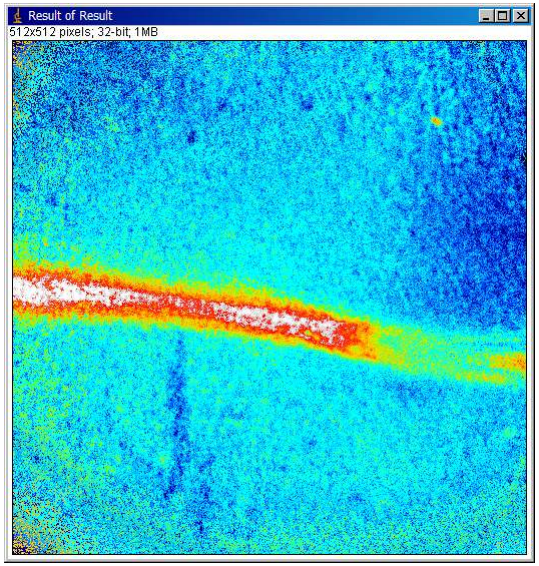
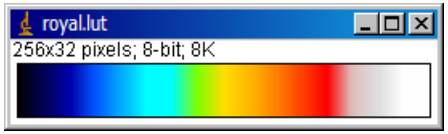


Image – Lookup Tables – royalで
黒青水緑黄赤白の疑似カラー表示。
このまま保存すると白黒で保存されるが、
Image – Type – RGB Colorに変えると
疑似カラーの色で保存される。



何も画像を読み込んでいない状態で
Image – Lookup Tables – royalで
黒青水緑黄赤白のカラーバー表示。



MATLAB画像ファイルの読込

imread関数でファイルからMATLABへ画像を読み込む。

```
F0 = imread('D:¥Ca2+Imaging¥20090110¥F0.tif');
F1 = imread('D:¥Ca2+Imaging¥20090110¥F1.tif');
```

読み込んだ画像を表示させるには、imshowやimagescを使う。

```
imshow(F0)   %スケーリングなし
imagesc(F0) %スケーリングあり
colormap(gray) %疑似カラーをデフォルトのjetからgreyに変更
```

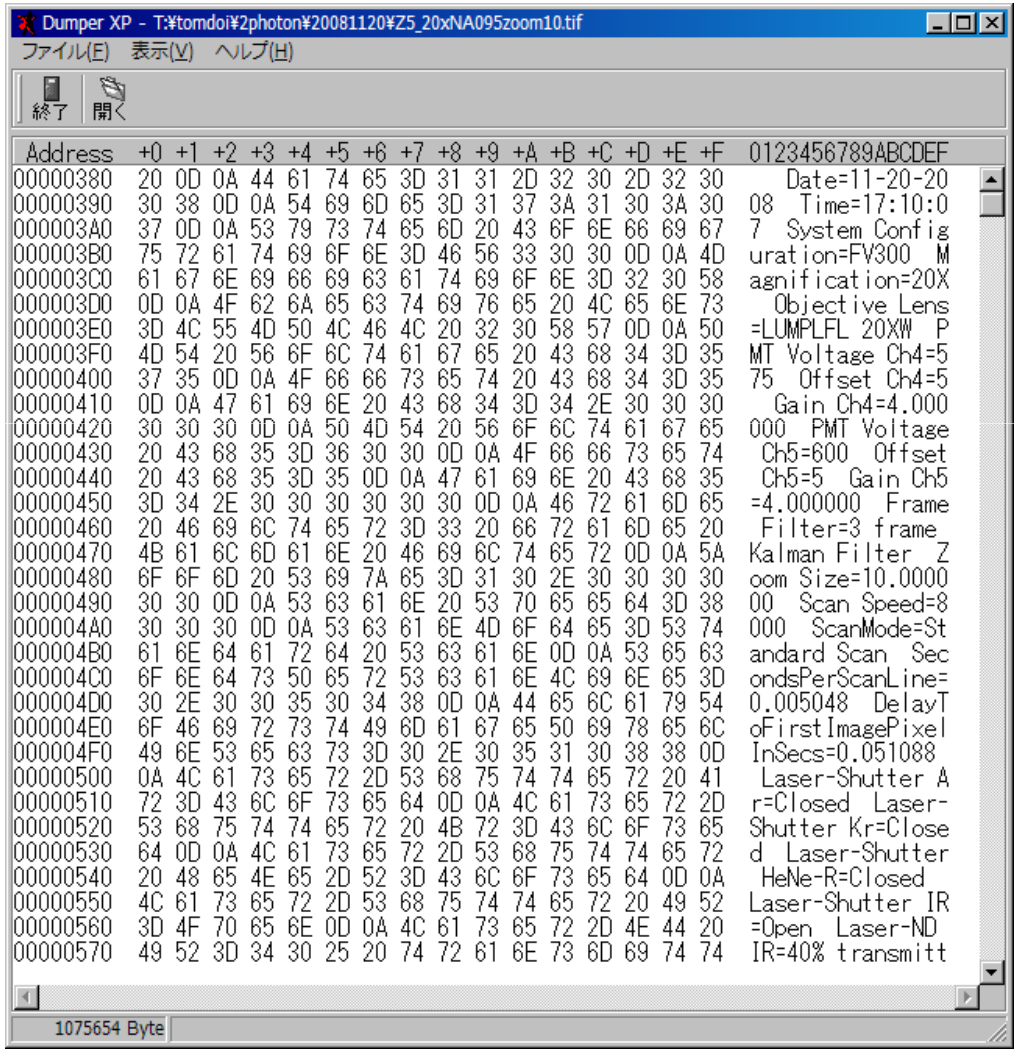
読めない場合は、ImageJで保存し直すか、別形式に保存。
Tiffのスタック読込はtiffread3関数、動画読込はaviread関数を使う。8-bitのデータならavireadが便利。

TIFFファイルは便利だが読込難

TIFFファイルには、顕微鏡の設定などの情報を書けるヘッダー部分が存在する。

Olympus FV300の例→
取得日時、対物レンズ倍率、対物レンズ名、使用チャネル、フォトマル電圧、フォトマルゲイン、フォトマルオフセット、中間倍率、スキャン速度、スキャンモード、レーザーシャッター開閉、レーザーNDフィルタ%、、、

このヘッダー部分が読込の邪魔になるので、ImageJで開いてから保存し直すとヘッダー部分が取れる。



蛍光変化率($\Delta F/F_0$)の計算・表示

```
F0double = double(F0); %double型に変換
F1double = double(F1); %double型に変換
DeltaFdouble = F1double – F0double;
DFF0double   = DeltaFdouble ./ F0double;
imagesc(DFF0double); %スケーリングありで画像表示
colormap(jet);       %カラーマップをjet青水緑黄赤茶に変更
colorbar             %カラーバーの表示
```

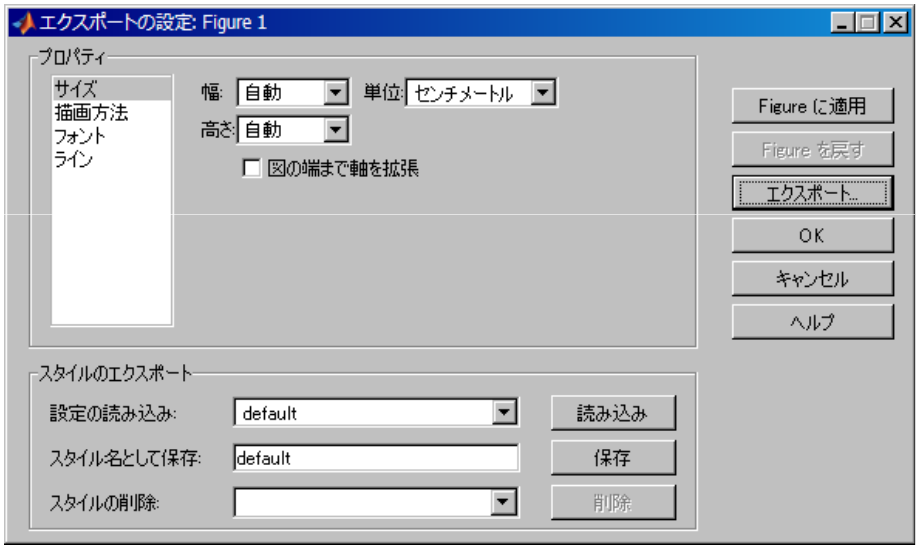
補足（プログラミング知識がある人用）:
MATLABは行列演算が標準なので、行列A –行列Bの計算にforループを使う必要はない。また、MATLABでは、行列A./行列Bと書くと行列の各要素同士の除算をする。行列A/行列Bでは、行列A x(行列Bの逆行列)の意味になってしまう。これはMATLAB特有。

蛍光変化率($\Delta F/F_0$)の保存

```
imwrite(DFF0double,'DFF0.tiff','tiff', ... %次行に続く
        'Compression','none');     %無圧縮でファイル保存
```

いままでのプログラムは.m形式で保存して使い回しできる

Figureウィンドウから、
ファイル(F) – エクスポートの設定…(R) – エクスポート
eps形式で保存すれば、Adobe Illustratorへの扱いがスムーズ



本チュートリアルでは触れないが、MATLABでFigureのタイトルや目盛数値などをプログラムで指定できる。論文のFigを何回も作るときに便利。

画像処理には高性能PCを！

Windows XP 64bit
(32bitでは1アプリケーションでメモリ2GBが上限)

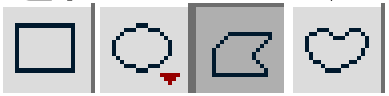
Intel Core2Duo 3.16GHz
(クロック数が計算速度に比例。Quadは効果なし？)

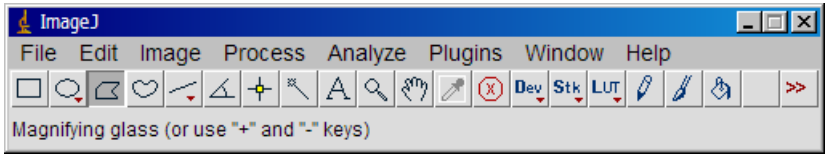
8GB Memory
(読込ファイルの4倍必要、5倍で十分)

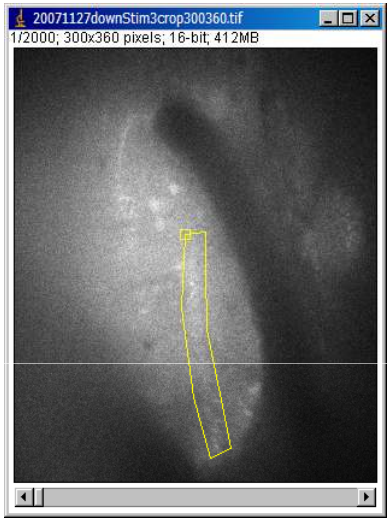
HDD 500GB + 1TB + 1TB + 1TB
(1TB ＜¥10000の時代なのでケチらない)

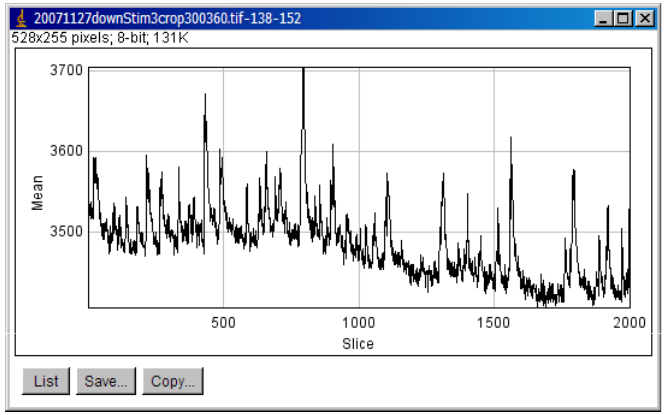
TeraStationPro 4TB 19インチラック型 RAID5 x 2
(データが消えると泣く。バックアップは必ず)

おまけ1：ROIを囲って経時変化

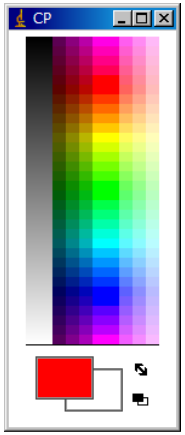
XYT画像で、ROIを囲むには、
ツールバーから、 
のどれかで囲って、
Image – Stacks – Plot Z-axis Profile
経時変化のグラフ数値はグラフウィンドウのSaveから保存。

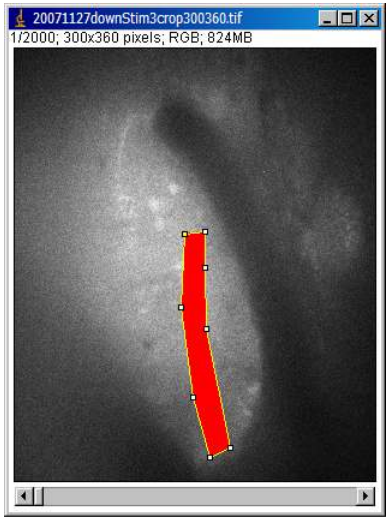






ただし、ROIの位置や形状は保存されないので、Image – Color – Color Pickerで好きな色を選んで、Edit – FillでROIを塗りつぶして画像を保存する。





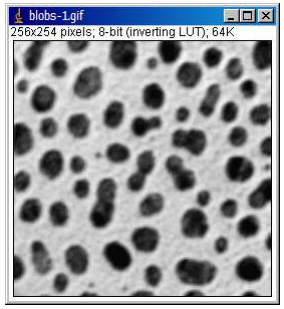
おまけ2：ImageJで細胞を数える

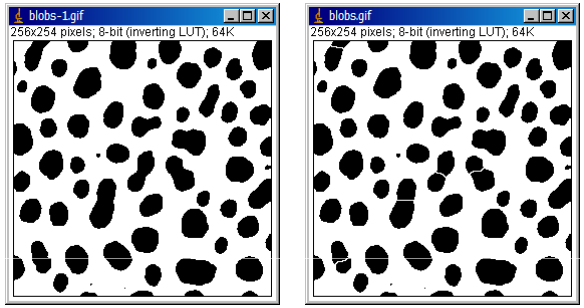
丸い細胞を数えるとき手軽に使える。

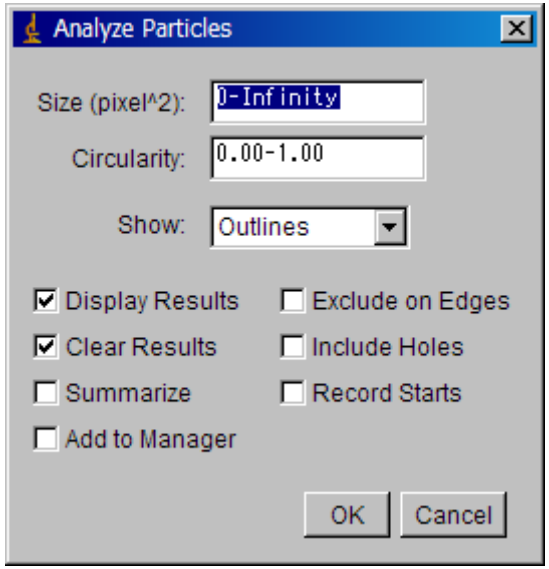
Process – Binary – Make Binaryで画像を二値化。結果に満足できなければ事前に画像フィルタを工夫。

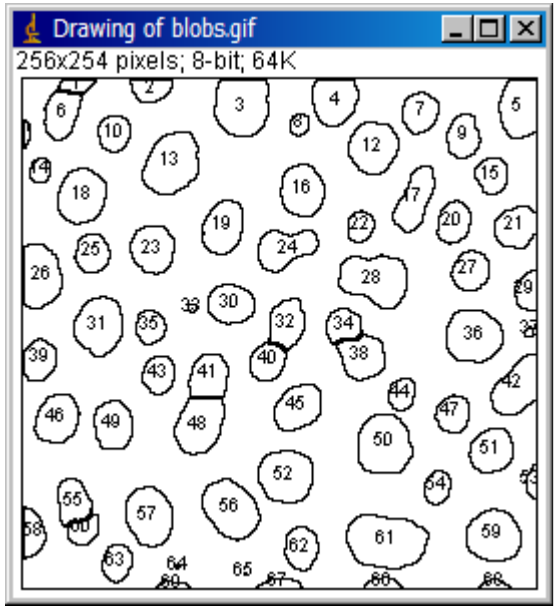
Process – Binary – Watershedで密着した複数の細胞を分離して認識

Analyze – Analyze Particles…
で丸を数える。細胞の大きさや丸さを指定して余計なゴミを数えないようにする。

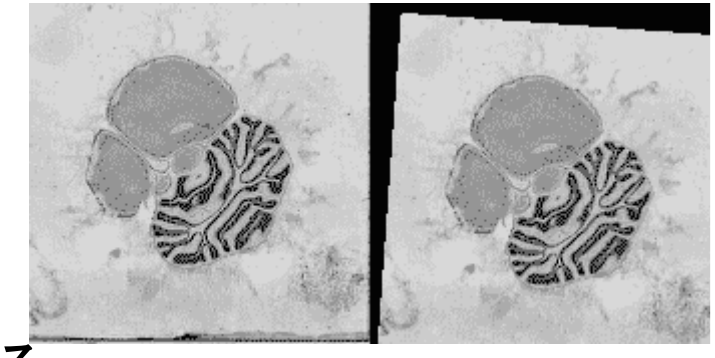




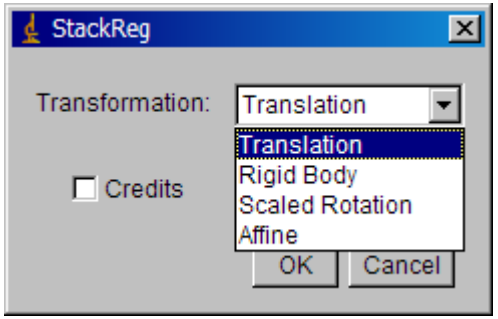




おまけ3：ImageJでズレやブレを補正



StackReg: スライス連続切片を並べ揃える
http://bigwww.epfl.ch/thevenaz/stackreg/
からダウンロードしてPluginsフォルダに入れる
動作にはTurboRegが必要
XYスタック画像を読み込んで、Plugin – StackReg
Translation（平行移動）
Rigid Body（平行移動と回転）
Scaled Rotation（平行移動と回転と拡大）
Affine（平行移動と回転と縦横不等拡大）



TurboReg: ブレ（揺れ）のある連続画像を補正する
http://bigwww.epfl.ch/thevenaz/turboreg/ からダウンロード
XYスタック画像と基準を読み込んで、Plugin – TurboReg

おまけ4：フィルタ処理で細胞自動認識