確認テストの解答

｛其の一｝

【問題】サンプル動画[kin.tif(10M)]は動原体が分かれていく様子を撮影したものである．後に動原体の移動速度の分布を解析することを目的に自動追跡せよ．結果をCSVファイルで保存せよ．

【答え】

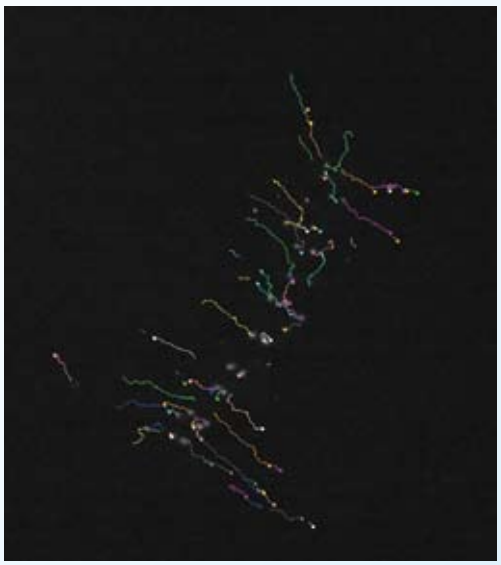
解析の流れは，MTrackJでまず大体の移動速度を測定し，この結果を元にParticleTrackerでトラッキングを行う．

[kin.tif(10M)]をまず開く（サンプル画像のプラグインをインストールしている必要がある）．**[MTrackJ]**を起動し，最初から最後まで追跡できそうなシグナルを選んで**“Add”**ボタンをクリックし，手動追跡を行う．**“Measure”**ボタンを押して結果を表示する．1フレームあたりの移動距離が3〜4ピクセルであることがこれでわかる．MTrackJのウィンドウを閉じて終了し，**[Particle Tracker 2D/3D]**を起動する．

まず粒子検出の設定．**“Radius”**，**“Cutoff”**，**“Per/Abs”**の設定値をいろいろ変えて，動原体シグナルがうまく検出される組み合わせを探す．すっぽりと収まる粒子半径は3ピクセル程度であることがまずわかるだろう．次に，CutoffとPer/Absの値の設定だが，動原体のシグナルが焦点面から時々外れて暗くなると検出が難しいことがわかる．暗いシグナルを含もうとすると，背景のノイズも検出されるからである．そこでノイズを低減するため，**[Gaussian blur...]**で若干のぼかしを加える．**“Sigma”**は1でよい．次に，動原体の移動速度の測定が目的であることから，「すべての動原体を検出するよりも，確実に動原体だけを検出する」ことを粒子検出の方針として，数値設定を行う．この結果

・Radius: 3
・Cutoff: 3
・Per/Abs: 0.3

という設定を筆者は選んだ．また，粒子リンクの設定は，確実に追跡を行いたいことから**“Link Range”**は1とする．**“Displacement”**はMTrackJの結果から5とする．



■図H1

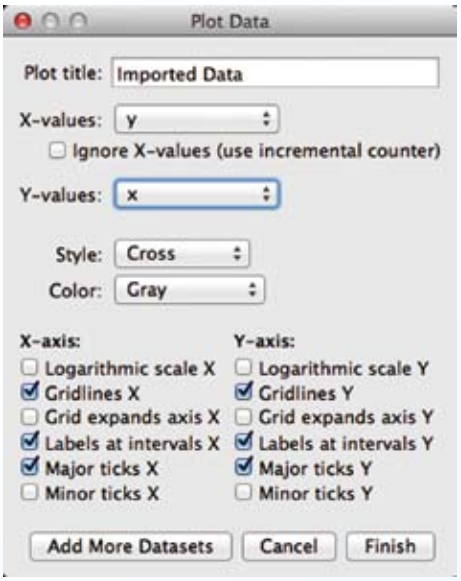
以上の設定で自動追跡を行った結果が図H1である．

“Focus on Area”の機能を使って結果の軌跡をよく見てみると，シグナルを動画の全長にわたって追跡できているものは少なく軌跡が断片化している．とはいえ，移動速度の分布を解析することが目的なので，断片化していても解析は行える．そこで，結果はこれでよし，とする．

軌跡のデータをCSVファイルに保存するには，“All Trajectories to Table”をクリックして結果をResultsに表示する．[File > Save as...]で，ファイル名を“any name.csv”として保存する．拡張子を“.csv”にするのは重要で，こうしないとCSV形式で保存されないので注意すること．

｛其の二｝

【問題】確認テスト｛其の一｝で保存したCSVファイルを[File > Import > Results]によって開き，
①連載第11回で使ったプラグイン集BARの[Plot Results]を使って，動原体の位置の変化をX-Yのグラフにプロットせよ．
②マクロを書いて1フレームあたりの移動距離の平均を計算し，Resultsの表に書き込め．そののちに速度の分布をヒストグラムとして表示せよ．

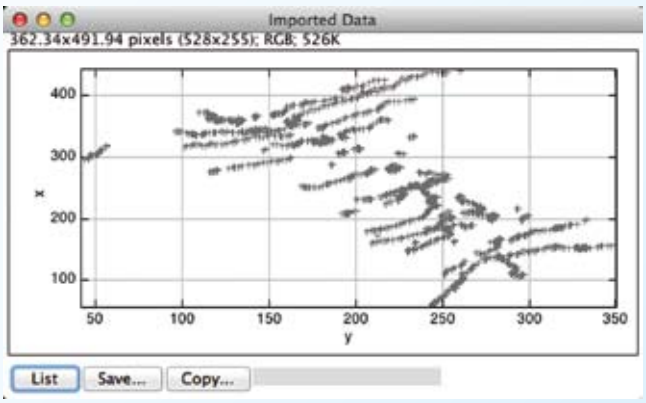


■図H2

【答え①】

図H2にあるように，[Plot Results]を設定する．

●ページ注1で述べたようにXとYを逆にすること．アスペクト比はあいにく変えられないので，図H3にあるように，縦方向につぶれているが，軌跡をグラフ上に表示できる．



■図H3

【答え②】

各時点での速度の計算と，表への書き込み

ある時点での（x1，y1）座標と，その次の時点での（x2，y2）座標を表から取得し，その間の距離を表に書き込めばよい．ただし，表のすべての行でこの計算をすると，異なる軌跡の最後の点と最初の点の間の距離も測定に含まれてしまう．したがってこの場合，計算をしないように条件分枝をする必要がある．以下のようなマクロになる．

```
1   for (i = 0; i < nResults - 1; i++){
2       trackID = getResult("Trajectory", i);
3       nexttrackID = getResult("Trajectory", i + 1);
4       if (trackID == nexttrackID){
5           x1 = getResult("x", i);
6           y1 = getResult("y", i);
7           x2 = getResult("x", i + 1);
8           y2 = getResult("y", i + 1);
9           displacement = sqrt(pow(x2 - x1, 2) + pow(y2 - y1, 2));
10          setResult("Displacement", i, displacement);
11      } else {
12          setResult("Displacement", i, NaN);
13      }
14  }
```

コピーできるようにコードはhttp://goo.gl/dCHWYBに掲載した．

解説する．

・1行目：forループの設定．表の最後の行は次の点がないので計算しない．

・2～3行目：ループ中，現在の行と次の行の軌跡ID番号を取得する．

・4行目：もし現在の行と次の行が同じ軌跡のデータであったら，という条件分枝．真であるならば，5行目から9行目にあるように，座標間の距離を計算する．偽であるならば，12行目を実行．

・5～8行目：現在の行と次の行のXY座標を取得する．

・9行目：座標間の距離，すなわち移動量を計算する．

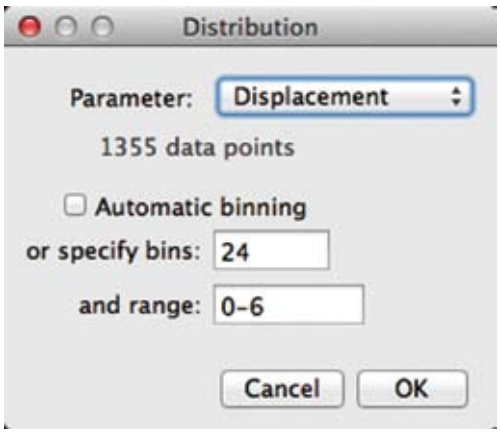
・10行目：移動量を“Displacement”の列に書き込む．

・12行目：軌跡と次の軌跡の間にあたるのでNaN（Not a Number，非数）を書き込む．

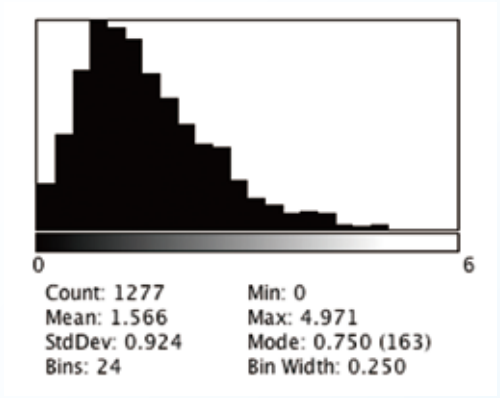
速度分布のプロット

Resultsの表にデータが書き込まれているので，Resultsのウィンドウがアクティブな状態で**[Results > Distribution...]**を実行し，開いた設定ウィンドウで図H4のようにヒストグラムの設定を行う．

OKをクリックすれば，速度分布が表示される（図H5）．



■図H4



■図H5