Nuclear Mapping of Nano-Drug Delivery Systems in Dynamic Cellular Environments

Ashwinkumar A. Bhirde[1], Ankur Kapoor[2] Gang Liu,[3] Ramiro Iglesias-Bartolome[4], Albert J Jin[5], Guofeng Zhang[5], Ruijun Xing[1], Seulki Lee[1], Richard D. Leapman[5], J Silvio Gutkind[4] and Xiaoyuan Chen[1] *

[1]Laboratory of Molecular Imaging and Nanomedicine, National Institute of Biomedical Imaging and Bioengineering, National Institutes of Health, Bethesda, MD 20892, USA, [2]Department of Radiology and Imaging Sciences, Clinical Center, National Institutes of Health, Bethesda, Maryland 20892. [3]Center for Molecular imaging and translational medicine, School of Public Health, Xiamen University, Xiamen 361005 China. [4]Oral and Pharyngeal Cancer Branch, National Institute of Dental and Craniofacial Research, National Institutes of Health, Bethesda, Maryland 20892, [5]Laboratory of Cellular Imaging and Macromolecular Biophysics, National Institute of Biomedical Imaging and Bioengineering, National Institutes of Health, Bethesda, Maryland 20982, USA

KEYWORDS: iron oxide nanoparticles. doxorubicin. drug resistance. computational. nuclear mapping. live cell imaging. cancer.

* Correspondence: Xiaoyuan Chen, shawn.chen@nih.gov

Supplementary Figures and Videos

Supplementary Figure S1 Fourier transform infrared spectroscopy (FTIR) characterization of Alkyl-PEI2k-SPIONPs-Dox (Nanodox). FTIR was used to characterize the DOX loaded magnetic nanocrystals. a. free DOX, b. PEI coated SPIONPs and c. Alkyl-PEI2k-SPIONPs-Dox. Spectra showed the presence of DOX onto the SPIONPs.

Supplementary Figure S2 Fluorescence spectrometer characterization of Alkyl-PEI2k-SPIONPs -Dox (Nanodox). Fluorescence measurements were carryout at 480 nm wavelength excitation for the florescent drug and dye loaded NPs. a. Nanodox-FITC b. Nano-FITC c. Nanodox d. DOX e. FITC f. Nano (Nano = SPIONPs). Florescence of DOX and FITC were detected in the conjugate.

Supplementary Figure S3 Drug Loading Efficiency Histogram plot shows the drug loading efficiency onto the SPIONPs at various polymer (PEI) to drug DOX ratio.

Supplementary Figure S4 Confocal live cellular image snapshots of DOX treated cancer cells a-b, DOX sensitive a, free DOX b, Nanodox and washed with PBS 3 times 3 hrs post treatment and resuspended in cell media. Nanodox and free drug on its own showed similar uptake profile and readily accumulated in the cell nuclei.

Supplementary Figure S5 Confocal live cellular image snapshots of DOX treated cancer resistant cells DOX resistant cells were treated similarly with **a**, free DOX and **b**, Nanodox and

washed with PBS 3 times 3 hrs post treatment and resuspended in cell media. Nanodox showed more than 90% uptake in the resistant cells compared to free drug DOX alone.

Supplementary Figure S6. Time dependent DOX mapping in a dynamic cellular environment in DOX sensitive and resistant cancer cells. a, Live cell snapshots of drug sensitive OVCAR8 and drug resistant OVCAR8/ADR cells treated with free DOX or Nanodox at 0 - 72 hrs. Nanodox shows the highest drug accumulation in the cell nuclei compared to free drug in both drug sensitive and resistant cells.

Supplementary Videos

Supplementary Video S1. Live microscopy movie of drug sensitive cells treated with DOX. Doxorubicin sensitive cells were treated with free DOX and observed under confocal fluorescence microscope live for an hour. Since DOX is florescent on its own, it is possible to track the movement of the drug directly in a live cellular environment. Because of smaller size of DOX it is not possible to image single DOX molecules using the current optical imaging instrumentation. We could image the DOX only after 30 mins post treatment. DOX got accumulated into the nucleus of the drug sensitive cancer cells. The uptake of DOX into the cells was exponential.

Supplementary Video S2 . Live microscopy movie of drug sensitive cells treated with Nanodox. Doxorubicin sensitive cells were treated with DOX loaded SPIONPs and observed under confocal fluorescence microscope live for an hour. Imaging was done at an excitation of ~480 and collected at ~570 nm emission to track the movement of the drug directly in a live cellular environment. Since the drug was loaded onto the nanoparticles it was possible to image the DOX molecules using the current optical imaging instrumentation. We could image the DOX

as early as 5 mins post treatment, because of high cell membrane penetration of SPIONPs. DOX accumulated into the nucleus of the drug sensitive cancer cells in an exponential rate.

Supplementary Video S3. Computational mapping movie of drug sensitive cells treated with DOX. Movie shows the dynamic movement of drug in a single dox sensitive cell for a total duration of 60 mins. The upper left panel shows the bright field mode of cell capture. The upper right panel shows the bright field plus florescent channel for dox, circled to locate the nuclear region. The lower left panel shows florescent view of the single whole cell and the lower right panel shows the florescent view of the nuclear region alone.

Supplementary Video S4. Computational mapping movie of drug sensitive cells treated with Nanodox. Movie shows the dynamic movement of drug in a single dox sensitive cell for a total duration of 60 mins. The upper left panel shows the bright field mode of cell capture. The upper right panel shows the bright field plus florescent channel for dox, circled to locate the nuclear region. The lower left panel shows florescent view of the single whole cell and the lower right panel shows the florescent view of the nuclear region alone.

Supplementary Video S5. Live microscopy movie of drug resistant cells treated with DOX. Doxorubicin resistant cells were treated with free DOX and observed under confocal fluorescence microscope live for an hour. Compared to imaging DOX in sensitive cells, it was not possible to track the movement of the drug directly in a live cellular environment at early time point. DOX was visible only after 45 mins post treatment only in couple of cells. Fewer (less than 5% of the cells showed presence of DOX) cells showed DOX uptake while only 0ne or two cells showed nuclear uptake as against more than 95% nuclear uptake in sensitive cells.

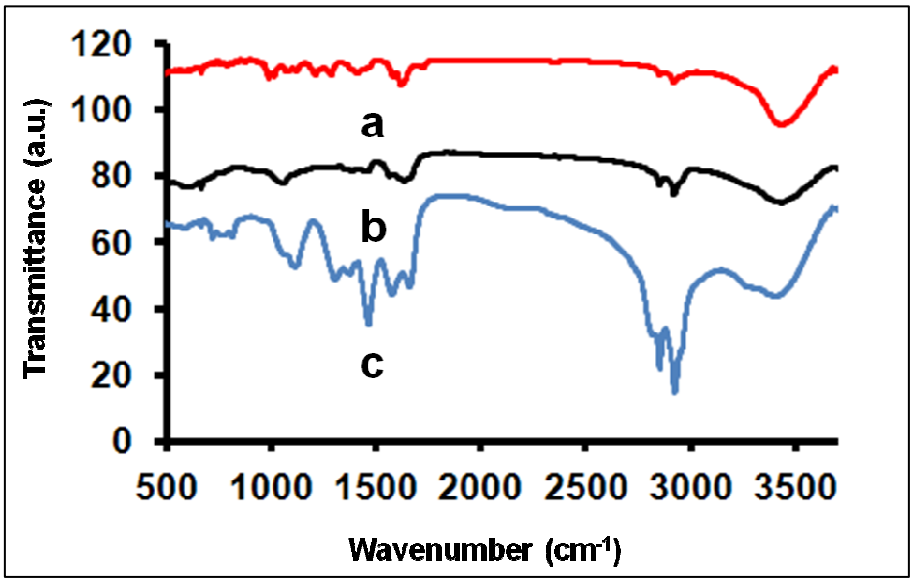
Supplementary Video S6. Live microscopy movie of drug resistant cells treated with Nanodox. Doxorubicin resistant cells were treated with DOX loaded SPIONPs and observed under confocal fluorescence microscope live for an hour. Compared to free drug Nanodox was visible at very early time point in the cancer cells. However there was sharp contrast in nuclear uptake compared to Nanodox in sensitive cells. First 30 mins did not show any promising uptake in the nuclear region. However in 60 mins there was nuclear uptake in couple of cells. Compared to drug on its own Nanodox was able to penetrate most of the resistant cells as evident from the live movie.

Supplementary Video S7. Computational mapping movie of drug resistant cells treated with DOX. Movie shows the dynamic movement of single dox resistant cell treated with free drug for a total duration of 120 mins. The upper left panel shows the bright field mode of cell capture. The upper right panel shows the bright field plus florescent channel for dox, circled to locate the nuclear region. The lower left panel shows florescent view of the single whole cell and the lower right panel shows the fluorescent view of the nuclear region alone.

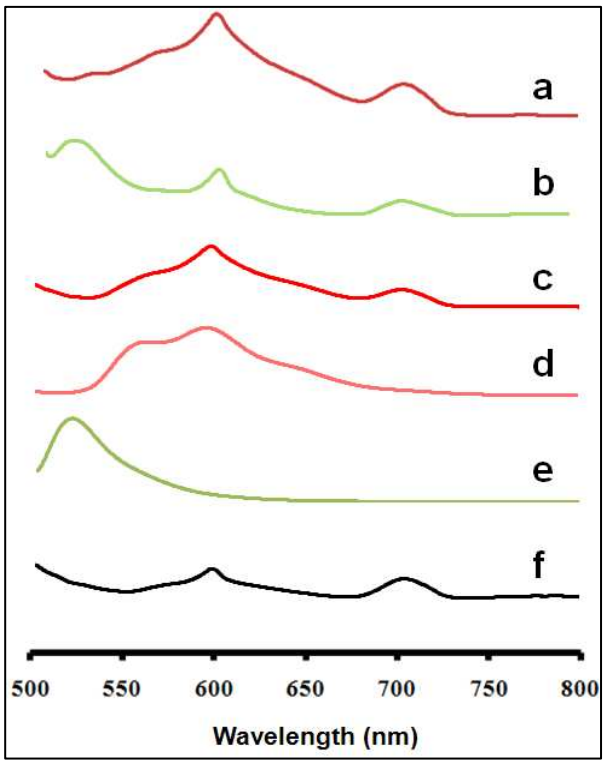
Supplementary Video S8. Computational mapping movie of drug resistant cells treated with Nanodox. Movie shows the dynamic movement of single dox resistant cell treated with Nanodox for a total duration of 120 mins. The upper left panel shows the bright field mode of cell capture. The upper right panel shows the bright field plus florescent channel for dox, circled to locate the nuclear region. The lower left panel shows florescent view of the single whole cell and the lower right panel shows the florescent view of the nuclear region alone.

Supplementary Video S9. Live microscopy movie of Nano-dox-FITC. DOX loaded SPIONPs-PEI labeled with FITC were treated to cancer cells and observed in live dynamic cellular environment for 60 mins post treatment.

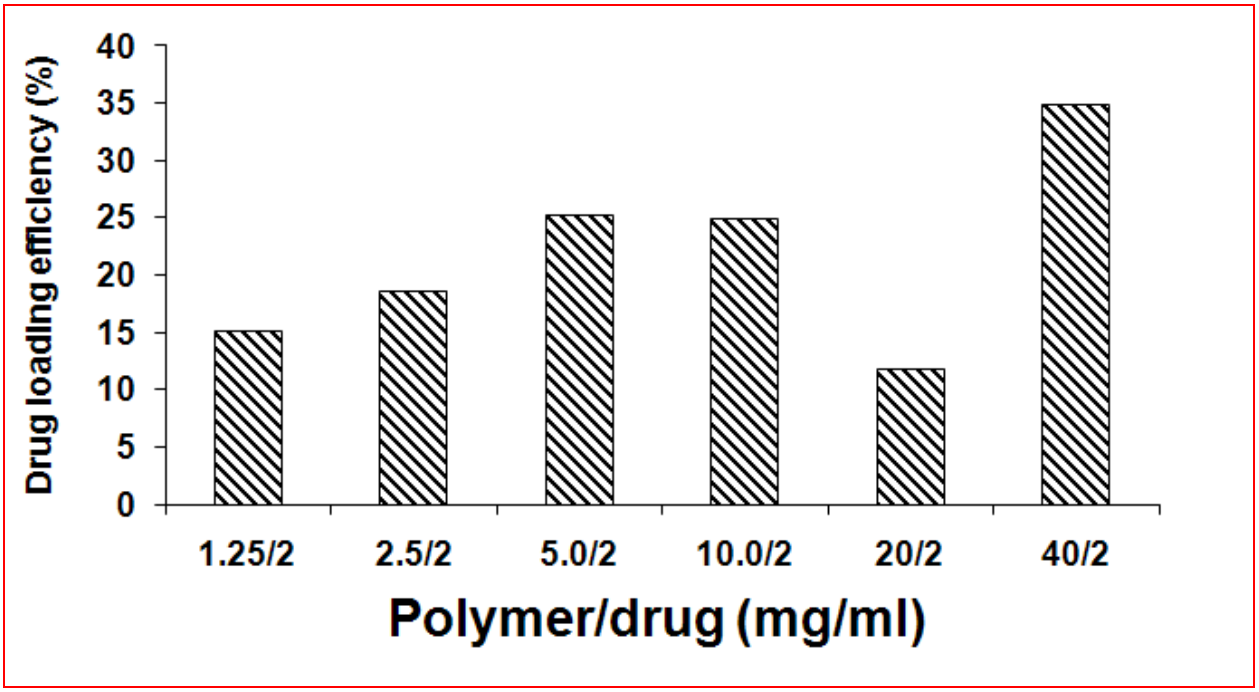
Supplementary Video S10. Live microscopy movie of Nano-FITC. SPIONPs-PEI labeled with FITC were treated to cancer cells and observed live for 60 mins.



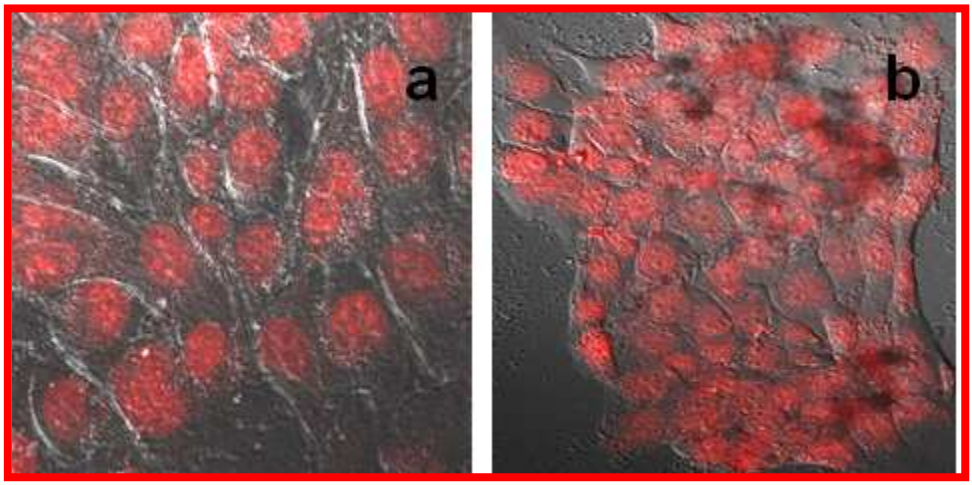
Supplementary Figure S1 Fourier transform infrared spectroscopy (FTIR) characterization of Alkyl-PEI2k-SPIONPs-DOX (Nanodox).



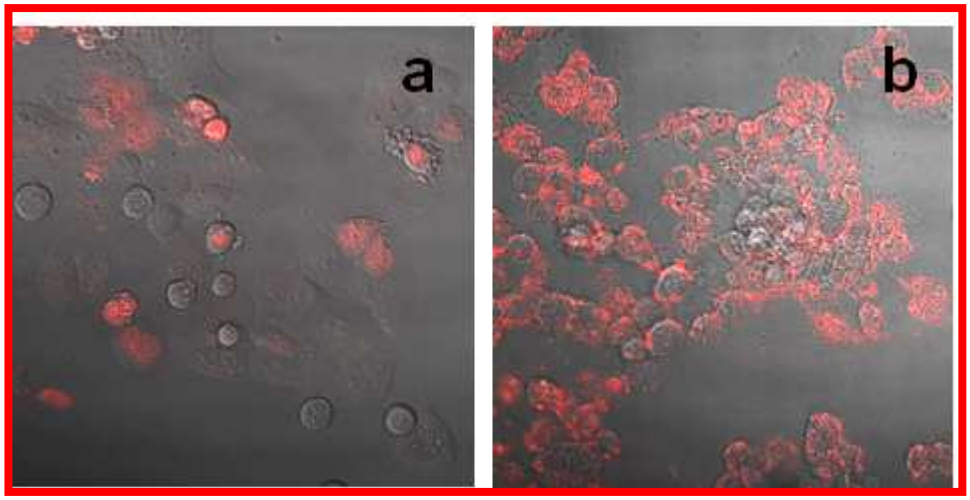
Supplementary Figure S2 Fluorescence spectrometer characterization of Alkyl-PEI2k-SPIONPs-DOX (Nanodox).



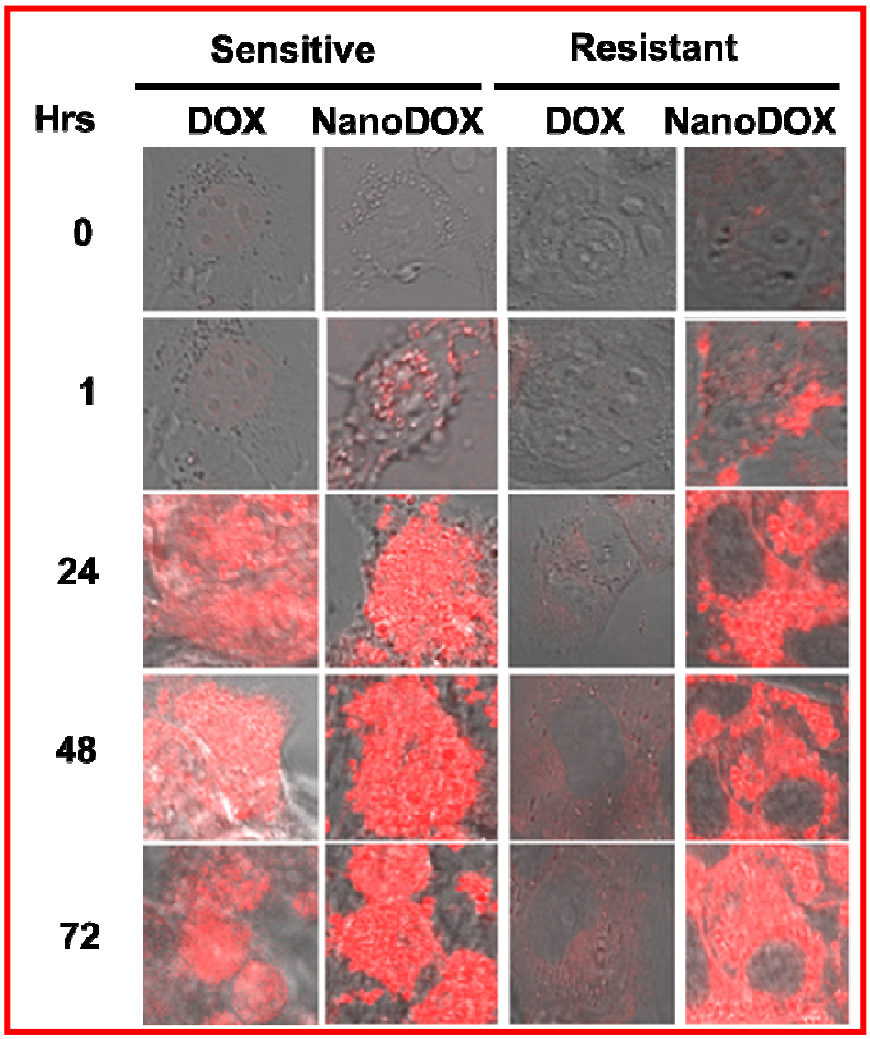
Supplementary Figure S3 Drug DOX Loading Efficiency of Alkyl-PEI2k-SPIONPs.



Supplementary Figure S4 Confocal live cellular image snapshots of DOX treated cancer cells.



Supplementary Figure S5 Confocal live cellular image snapshots of DOX treated cancer resistant cells.



Supplementary Figure S6. Time dependent DOX mapping in a dynamic cellular environment in DOX sensitive and resistant cancer cells.

Supplementary Note:

Algorithm used for tracking and masking cancer cells in dynamic cellular environment. Coding was done using Matlab.

```
%TRACK   Track nucleus in live microscopy images.
%   TRACK(Image,Template,BoundingBox,WindowSize) tracks a nucleus in a
%   series of live microscopy images. Image 3D input matrix, the third
%   dimension (of size nz) represents the time dimension. Template is
%   the inital user input that is used for template matching. At each
%   time point the template is updated. BoundingBox is the enclosing
%   rectangle for the inital ellipse. WindowSize is the maximum allowed
%   displacement of the ellipse center. If template matching return a
%   displacement larger than WindowSize, the tracking is interrupted to
%   allow the user to verify and correct, if necessary, the current
%   tracking result.
%
%   [bbox, bboxfiltered] =  track(I,T,BB,sw,fw) filters the tracked center
%   using a median filter of size fw. If no input is supplied a default
%   value of 6 is used.
%
%   [bbox, bboxfiltered, mask, otherchannelmasked]
%                           =  track(I,T,BB,sw,fw,otherchannel)
%   computes the mask for other channel images based on the location of the
%   tracked ellipse in the corresponding live microscopy images
%
%   [bbox, bboxfiltered, mask, otherchannel1masked, otherchannel2masked]
```

```
%                    =  track(I,T,BB,sw,fw,otherchannel1,otherchanne2)
%   computes the mask for other channel images based on the location of the
%   tracked ellipse in the corresponding live microscopy images
%
function [TrackedBoundingBox, TrackedBoundingBoxFiltered, varargout]...
   =track(Image,Template,BoundingBox,WindowSize,varargin)

yind=floor(BoundingBox(2)):floor(BoundingBox(2))+floor(BoundingBox(4));
xind=floor(BoundingBox(1)):floor(BoundingBox(1))+floor(BoundingBox(3));
% Display the first image and input template
subplot(1,2,1);imshow(Template);
subplot(1,2,2);imshow(Image(:,:,1));

[nc nr nz] = size(Image);
xpeak=zeros(nz,1);
ypeak=zeros(nz,1);
offset=zeros(nz,2);
TrackedBoundingBox = zeros(nz,4);

xs=xind(1);
ys=yind(1);
TrackedBoundingBox(1,:) = BoundingBox;
for i=1:nz
   % Template matching calculates the sum of squared difference (SSD) and
   % normalized cross correlation (NCC) between template and current image
   [ImageSSD,ImageNCC]=template_matching(Template,Image(:,:,i));
   % For our nucleus tracking application we only use NCC
```

```
% Find location of peak value of NCC
[y,x]=find(ImageNCC==max(ImageNCC(:)));
% Display NCC image and the peak
subplot(2,2,2);imagesc(ImageNCC);hold on; plot(x,y, 'r', 'Marker', '+'); hold off
ypeak(i) = y;
xpeak(i) = x;
if (i>1)
   offset(i,:) = [ypeak(i)-ypeak(i-1), xpeak(i)-xpeak(i-1)];
   % Check if displacement in larger than user supplied threshold
   if (abs(offset(i,1))>WindowSize) || (abs(offset(i,2))>WindowSize)
      % Yes, it is. Prompt user to verify and if necessary correct it
      disp(['Offsets larger than search windows.',num2str(offset(i,:))]);
      fig=gcf;
      Pattern=Image(:,:,i);
      fig2=figure;imshow(Pattern);
      ehandle=imellipse(gca,[xs,ys, BoundingBox(3:4)]);
      position = wait(ehandle);
      bw=ehandle.createMask();
      stats = regionprops(bw, 'BoundingBox');
      BoundingBox=stats.BoundingBox;
      yind=floor(BoundingBox(2)):floor(BoundingBox(2))+floor(BoundingBox(4));
      xind=floor(BoundingBox(1)):floor(BoundingBox(1))+floor(BoundingBox(3));
      % Update the template based on user input. If user did not
      % interact with GUI then it remains unchanged.
      Template=Pattern(yind,xind);
      xs=xind(1);
      ys=yind(1);
```

```
            xpeak(i)=xs+round(BoundingBox(3)/2);
            ypeak(i)=ys+round(BoundingBox(4)/2);
            offset(i,:)=[ys-ypeak(i-1),xs-xpeak(i-1)];
            close(fig2);
            figure(fig);
        else
            % Update the template based on current image
            Template=Image(yind+offset(i,1), xind+offset(i,2));
            xs=xs+offset(i,2);
            ys=ys+offset(i,1);
        end
        % Display template and images
        subplot(2,2,1);imagesc(Template);
        subplot(2,2,4);imagesc(Image(:,:,i)-Image(:,:,i-1));
        subplot(2,2,3);imshow(Image(:,:,i));
        title(num2str(i));
        % Draw the ellipse
        rectangle('Position',[xs,ys,
BoundingBox(3:4)],'EdgeColor','g','LineWidth',2,'Curvature',[1,1]);
        title(sprintf('%d, %d, %d, %d, %d',i,ypeak(i),xpeak(i), offset(i,1), offset(i,2)));
        TrackedBoundingBox(i,:)=[xs,ys, BoundingBox(3:4)];
        pause(0.1);
    end
end
if (nargout>1)
    FilterSize = 6;
    if (nargin > 4)
        if (isscalar(varargin{1}))
```

```
            FilterSize=varargin{1};
        else
            error('Filter Size must be a scalar');
        end
    end
    TrackedBoundingBoxFiltered=[medfilt1(TrackedBoundingBox(:,1),FilterSize),...
        medfilt1(TrackedBoundingBox(:,2),FilterSize),...
        TrackedBoundingBox(:,3:4)];
end
generatemaskedoutputs=0;
twochannel=0;
if (nargout>=3)
    if (nargin < 6)
        error('Input channel must be supplied to compute the mask');
    end
    channel1=varargin{1};
    channel1masked=zeros(size(channel1));
    generatemaskedoutputs=1;
elseif (nargout>=4)
    if (nargin < 7)
        error('Input channel must be supplied to compute the mask');
    end
    channel2=varargin{2};
    channel2masked=zeros(size(channel2));
    twochannel=1;
else
    error('Only two extra channels are supported');
```

```
end

if (generatemaskedoutputs)
    imask=zeros(nc, nr, nz);
    for i=1:nz
        imshow(channel1(:,:,i));
        ehandle=imellipse(gca,TrackedBoundingBoxFiltered(i,:));
        mask=ehandle.createMask();
        imask(:,:,i)=mask;
        ch=channel1(:,:,i);
        ch(isnan(ch))=0;
        ch(~mask)=0;
        channel1masked(:,:,i)=ch;
        if (twochannel)
            ch=channel2(:,:,i);
            ch(isnan(ch))=0;
            ch(~mask)=0;
            channel2masked(:,:,i)=ch;
        end
    end
    varargout{1}=imask;
    varargout{2}=channel1masked;
    if (twochannel)
        varargout{3}=channel2masked;
    end
end
```