```
function varargout = ORCID(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @ORCID_OpeningFcn, ...
                   'gui_OutputFcn',  @ORCID_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function ORCID_OpeningFcn(hObject, eventdata, handles, varargin) %#ok<*INUSL>
handles.output = hObject;
channeloptions={'Standard deviation' 'Mean Pixel value' 'Area' 'Random' 'on' 'off'
set(handles.popupredchannel,'string',channeloptions)
set(handles.popupredchannel,'value',4)
set(handles.popupgreenchannel,'string',channeloptions)
set(handles.popupgreenchannel,'value',4)
set(handles.popupbluechannel,'string',channeloptions)
set(handles.popupbluechannel,'value',4)
set(handles.popupchannel,'string',{'Channel 1 (Red)' 'Channel 2 (Green)' 'Channel
set(handles.radiooriginal,'Enable','off')
set(handles.radiofilter,'Enable','off')
set(handles.radiofilterthresh,'Enable','off')
set(handles.radionone,'Enable','off')
set(handles.frameslider,'Enable','off')
set(handles.regionpropsbutton,'Enable','off')
set(handles.applybutton,'Enable','off')
set(handles.ROIbutton,'Enable','off')
set(handles.ZstackClusterButton,'Enable','off')
set(handles.stack3dbutton,'Enable','off')
set(handles.slidertext,'string','Frame: 1')
handles.path=cd;
handles.channel=1;
handles.FILTEREDIMAGE=imread('TESTIMAGE.png');
handles.IMAGEINPUT=imread('TESTIMAGE.png');
handles.ROImask=zeros(size(handles.IMAGEINPUT))+1;
handles.graphtype=handles.radiooriginal;
handles.threshold=0.5;
set(handles.thresholdtext,'string',num2str(handles.threshold))
handles.autoupdate=0;
handles.autoid=0;
handles.currentframe=1;
handles.redgraph=4;
handles.greengraph=4;
handles.bluegraph=4;
handles.coords=[];
handles.rgb=[0 0 0];
handles.mikestopix=0.212;
handles.vmikestopix=0.45;
handles.edgemask=str2double(get(handles.edgeremovetext,'String'));
handles.stackrange(1)=1;
handles.stackrange(2)=1;
handles.INFO=[];
handles.datamode=0; %change this to 1 when you get 3D data
%colormap('gray')
% handles.appendnumber=0;
addpath('subfunctions');
% Update handles structure
guidata(hObject, handles);
function varargout = ORCID_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
function loadbutton_Callback(hObject, eventdata, handles) %#ok<*DEFNU>
[a b c]=uigetfile({'*.*'},'Select a file to load', ...
    'Multiselect','off',handles.path);
if c==1;
    colormap gray;
    handles.fname=a;
    handles.path=b;
    set(handles.radiooriginal,'Enable','off')
    set(handles.radiofilter,'Enable','off')
    set(handles.radiofilterthresh,'Enable','off')
    set(handles.radionone,'Enable','off')
    handles.IMAGEINPUT=imread([handles.path handles.fname]);
    tempvar=size(handles.IMAGEINPUT);
    handles.ROImask=zeros(tempvar(1),tempvar(2))+1;
    imagesc(handles.IMAGEINPUT,'parent',handles.axes1)
    handles=Update_All_Data(handles);
    set(handles.frameslider,'Enable','off')
    set(handles.regionpropsbutton,'Enable','off')
    %
    set(handles.applybutton,'Enable','on')
    set(handles.ROIbutton,'Enable','on')
    set(handles.ZstackClusterButton,'Enable','on')
    set(handles.stack3dbutton,'Enable','on')
    %%FIND the scale automatically
    try
        ftext=[handles.path handles.fname(1:end-4) '.txt'];
        str=fscanf(fopen(ftext),'%s');
        str=str((strfind(str,'XDimension')):end);
        commas=strfind(str,',');
        str=str((commas(2)+1):end);
        str=str((1:strfind(str,'[um/Pixel]')-1));
        disp('Automatically found image scale:')
        disp([str ' microns per pixel'])
        set(handles.mikestopixtext,'String',str)
        handles.edgemask=str2double(get(handles.edgeremovetext,'String'))/handles.
        %now find z scaling
        str=fscanf(fopen(ftext),'%s');
        str=str((strfind(str,'ZDimension')):end);
        commas=strfind(str,',');
        str=str((commas(2)+1):end); %go to the second comma after the string
        str=str((1:strfind(str,'[um/Slice]')-1));%read until you arrrive at the un
        handles.vmikestopix=eval(str);%turn string to a number
        disp('Automatically found slice depth')
        disp([str ' microns per slice'])
    catch error
        disp(error)
    end
    %%End finding scale
    imageinfo=imfinfo([handles.path handles.fname]);
    handles.numframes=length(imageinfo); %find dimensionality of image
    handles.hardmax=2^max([imageinfo.BitDepth]);
    if handles.numframes>1;
        set(handles.frameslider,'Enable','on');
        set(handles.frameslider,'SliderStep',[1/(handles.numframes-1) 10/(handles.
    end;

%     %Auto threshold
%     for i=1:handles.numframes;
%         if i==1; IMIN=imread([handles.path handles.fname]); end;
%         if i>1; IMIN=imread([handles.path handles.fname],'index',i); end;
%         maxval(i)=max(max(IMIN(:,:,handles.channel)));
%         minval(i)=min(min(IMIN(:,:,handles.channel)));
%     end;
%     threshval=min(minval)+(max(maxval)-min(minval))/10;
%     disp(hardmax)
%     threshval=threshval/hardmax;

end;
guidata(hObject, handles);
function frameslider_Callback(hObject, eventdata, handles) %#ok<*INUSD>
set(handles.radiooriginal,'Enable','off')
set(handles.radiofilter,'Enable','off')
set(handles.radiofilterthresh,'Enable','off')
set(handles.radionone,'Enable','off')
set(handles.regionpropsbutton,'Enable','off')

handles.frame=floor(get(hObject,'Value')*(handles.numframes-1))+1;
handles.IMAGEINPUT=imread([handles.path handles.fname],handles.frame);
imagesc(handles.IMAGEINPUT,'parent',handles.axes1)
set(handles.slidertext,'string',['Frame: ' num2str(handles.frame)])
drawnow;
handles=Update_All_Data(handles);
guidata(hObject, handles);
function frameslider_CreateFcn(hObject, eventdata, handles)
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'
    set(hObject,'BackgroundColor',[.9 .9 .9]);
```

```
end
function popupbluechannel_Callback(hObject, eventdata, handles)
handles.bluegraph=get(hObject,'value');
handles=Update_Clusters(handles);
guidata(hObject, handles);
function popupbluechannel_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
    set(hObject,'BackgroundColor','white');
end
function popupgreenchannel_Callback(hObject, eventdata, handles)
handles.greengraph=get(hObject,'value');
handles=Update_Clusters(handles);
guidata(hObject, handles);
function popupgreenchannel_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
    set(hObject,'BackgroundColor','white');
end
function popupredchannel_Callback(hObject, eventdata, handles)
handles.redgraph=get(hObject,'value');
handles=Update_Clusters(handles);
guidata(hObject, handles);
function popupredchannel_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
    set(hObject,'BackgroundColor','white');
end
function upperfiltertext_Callback(hObject, eventdata, handles)
handles.upperR=str2double(get(hObject,'string'));
handles=Update_All_Data(handles);
guidata(hObject, handles);
function upperfiltertext_CreateFcn(hObject, eventdata, handles)
set(hObject,'string','0.96')
handles.upperR=str2double(get(hObject,'string'));
guidata(hObject, handles);
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
    set(hObject,'BackgroundColor','white');
end
function lowerfiltertext_Callback(hObject, eventdata, handles)
handles.lowerR=str2double(get(hObject,'string'));
handles=Update_All_Data(handles);
guidata(hObject, handles);
function lowerfiltertext_CreateFcn(hObject, eventdata, handles)
set(hObject,'string','0.63')
handles.lowerR=str2double(get(hObject,'string'));
guidata(hObject, handles);
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
    set(hObject,'BackgroundColor','white');
end
function thresholdtext_Callback(hObject, eventdata, handles)
handles.threshold=str2double(get(hObject,'string'));
handles=Update_All_Data(handles);
guidata(hObject, handles);
function thresholdtext_CreateFcn(hObject, eventdata, handles)
handles.threshold=str2double(get(hObject,'string'));
guidata(hObject, handles);
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
    set(hObject,'BackgroundColor','white');
end
function popupchannel_Callback(hObject, eventdata, handles)
handles.channel=get(handles.popupchannel,'value');
%Change channel above and update fourier below:
handles=Update_All_Data(handles);

guidata(hObject, handles);
function popupchannel_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
    set(hObject,'BackgroundColor','white');
end
function uipanel4_SelectionChangeFcn(hObject, eventdata, handles)
handles.graphtype=eventdata.NewValue;
handles.graphimage=update_display(handles.graphtype,handles);
imagesc(handles.graphimage,'parent',handles.axes1)
handles=Update_Clusters(handles);
try handles=update_cluster_plot(handles); end;
guidata(hObject, handles);
function applybutton_Callback(hObject, eventdata, handles)
handles=fourier_update(handles);
handles.graphimage=update_display(handles.graphtype,handles);
imagesc(handles.graphimage,'parent',handles.axes1)
handles=Update_Clusters(handles);
guidata(hObject, handles);
function checkboxauto_Callback(hObject, eventdata, handles)
handles.autoupdate=get(hObject,'Value');
guidata(hObject, handles);
function regionpropsbutton_Callback(hObject, eventdata, handles)
BEF=handles.autoid;
handles.autoid=1;%oops, this is just to make the following function work.
handles=Update_Clusters(handles);
handles.autoid=BEF;%reset to whatever it was before.
guidata(hObject, handles);
function checkboxautoID_Callback(hObject, eventdata, handles)
handles.autoid=get(hObject,'value');
guidata(hObject, handles);
function edgemaskbox_Callback(hObject, eventdata, handles)
handles.edgemask=get(hObject,'value');
if handles.edgemask==1;
    handles.edgemask=str2double(get(handles.edgeremovetext,'String'))/handles.mike
else
    handles.edgemask=-1;
end;
handles=Update_All_Data(handles);
guidata(hObject, handles);
function uitable1_CellSelectionCallback(hObject, eventdata, handles)
if handles.datamode==0;
    set(hObject,'ColumnName',{'Ecc';'Ori';'Value';'Area';'Perim.';'ID' })
    try %#ok<*TRYNC>
        %the following block sometimes messes up when organizing the list
        cell=eventdata.Indices;
        data=get(hObject,'Data');
```

```
                inds=data(cell(:,1),end);
                handles.cellinds=inds;
                %coords=cat(1,handles.STATS(inds).PixelList);
                %plot(coords(:,1),max(get(handles.axes1,'Ylim'))-coords(:,2),'.','color',[
            %     hold on
            %     axis equal
            %     set(handles.axes2,'Xlim',get(handles.axes1,'Xlim'))
            %     set(handles.axes2,'Ylim',get(handles.axes1,'Ylim'))
            %     hold off

                imagesc(handles.graphimage,'parent',handles.axes2)
                hold on
                hulls=[];
                for ii=1:length(inds);
                    %rectangle('position',handles.STATS(inds(ii)).BoundingBox,'edgecolor',
                    hull=handles.STATS(inds(ii)).ConvexHull;
                    col=[handles.RGB(inds(ii),1) handles.RGB(inds(ii),2) handles.RGB(inds(
                    plot(hull(:,1),hull(:,2),'color',col,'parent',handles.axes2)
                    hulls=cat(1,hulls,hull);
                end;
                axis equal
                border=(max(hulls(:,1))-min(hulls(:,1)));
                set(handles.axes2,'Xlim',[min(hulls(:,1))-border max(hulls(:,1))+border])
                set(handles.axes2,'Ylim',[min(hulls(:,2))-border max(hulls(:,2))+border])
                hold off
            end;
        end;
        if handles.datamode==1;
            set(hObject,'ColumnName',{'Volume';'Surface';'ID';'Brightness';' ';' ' })
            %the following block sometimes messes up when organizing the list
            ax=handles.axes2; %#ok<*NASGU>
            %maxcoord=max(cat(1,cat(1,handles.MegaCluster.xx),cat(1,handles.MegaCluster.yy
            %mincoord=min(cat(1,cat(1,handles.MegaCluster.xx),cat(1,handles.MegaCluster.yy
            %zmin=min(cat(1,MegaCluster.zz))*handles.mikestopix;
            %zmax=max(cat(1,MegaCluster.zz))*handles.mikestopix;
            %zrange=(zmax-zmin);
            %axis([mincoord maxcoord mincoord maxcoord zmin-zrange*2.5 zmin+zrange*2.5 0 1
            %teh above line sets the axis to span all image space, and then scale z
            Rind=0; %This index is used in the data refinement
            %plot the data!
            inds=1;
            cell=[eventdata.Indices(:,1)]';
            data=get(hObject,'Data');
            inds=[data(cell,3)]';
            disp(inds)
            for i=inds;
                col=handles.MegaCluster(i).color;
                col3=handles.MegaCluster(i).color2;
                obj=handles.MegaCluster(i).obj;
                %col=[random('uniform',0,1) random('uniform',0,1) random('uniform',0,1)];
                %col=[MegaCluster(i).Volume/max([MegaCluster.Volume]) 1 random('uniform',0
        %         xnoise=0*random('uniform',0,1,size(handles.MegaCluster(i).xx));
        %         ynoise=0*random('uniform',0,1,size(handles.MegaCluster(i).yy));
        %         znoise=0*random('uniform',0,1,size(handles.MegaCluster(i).zz));
                xx=handles.MegaCluster(i).xx;
                yy=handles.MegaCluster(i).yy;
                zz=handles.MegaCluster(i).zz;
                uz=unique(zz);
                for j=1:length(uz)
                    col2=col*(j/length(unique(zz)));
                    col4=col3*(j/length(unique(zz)));
                    %col3=spcol;
                    %col2=col;
                    x=xx(vertcat(handles.MegaCluster(i).zz)==deal(uz(j)));
                    y=yy([handles.MegaCluster(i).zz]==deal(uz(j)));
                    z=zz([handles.MegaCluster(i).zz]==deal(uz(j)));
                    xnoise=random('normal',0,0.5,size(x));
                    ynoise=random('normal',0,0.5,size(y));
                    znoise=random('normal',0,0.5,size(z));
                    hand(i,j)=plot3(...
                        handles.mikestopix*(x+xnoise),...
                        handles.mikestopix*(y+ynoise),...
                        handles.vmikestopix*(z+znoise),...
                        obj,'color',col4,'MarkerSize',10,'MarkerFaceColor',col2,'parent',a
                    drawnow;
                    hold(ax,'on')
                end;
                axis(ax,'tight','equal');
                grid(ax,'on')
                camproj(ax,'perspective')
            end;
            hold(ax,'off')
        end;
        guidata(hObject, handles);
        function stack3dbutton_Callback(hObject, eventdata, handles)
        F_CT3D([handles.path handles.fname],...
            handles.lowerR,handles.upperR,handles.threshold,handles.ROImask,...
            1,handles.edgemask,...
            handles.stackrange(1),handles.stackrange(2),handles.text15)
        function minframesedit_Callback(hObject, eventdata, handles)
        handles.stackrange(1)=str2double(get(hObject,'String'));
        guidata(hObject, handles);
        function minframesedit_CreateFcn(hObject, eventdata, handles)
        % hObject    handle to minframesedit (see GCBO)
        % eventdata  reserved - to be defined in a future version of MATLAB
        % handles    empty - handles not created until after all CreateFcns called

        % Hint: edit controls usually have a white background on Windows.
        %       See ISPC and COMPUTER.
        if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
            set(hObject,'BackgroundColor','white');
        end
        function maxframesedit_Callback(hObject, eventdata, handles)
        handles.stackrange(2)=str2double(get(hObject,'String'));
        guidata(hObject, handles);
        function maxframesedit_CreateFcn(hObject, eventdata, handles)
        % hObject    handle to maxframesedit (see GCBO)
        % eventdata  reserved - to be defined in a future version of MATLAB
        % handles    empty - handles not created until after all CreateFcns called

        % Hint: edit controls usually have a white background on Windows.
        %       See ISPC and COMPUTER.
        if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
            set(hObject,'BackgroundColor','white');
        end
        function uipanel8_SelectionChangeFcn(hObject, eventdata, handles)
        if handles.datamode==0;
            switch eventdata.NewValue
                case handles.ORG1
                    set(handles.uitable1,'Data',sortrows(handles.INFO,-1));
                case handles.ORG2
                    set(handles.uitable1,'Data',sortrows(handles.INFO,2));
                case handles.ORG3
                    set(handles.uitable1,'Data',sortrows(handles.INFO,-3));
                case handles.ORG4
                    set(handles.uitable1,'Data',sortrows(handles.INFO,-4));
                case handles.ORG5
                    set(handles.uitable1,'Data',sortrows(handles.INFO,-5));
                case handles.ORG6
                    set(handles.uitable1,'Data',sortrows(handles.INFO,6));
            end;
        else
            NINFO=[handles.MegaCluster.Volume;handles.MegaCluster.surface_area;handles.Meg
            switch eventdata.NewValue
                case handles.ORG1
                    set(handles.uitable1,'Data',sortrows(NINFO,-1));
                case handles.ORG2
                    set(handles.uitable1,'Data',sortrows(NINFO,-2));
                case handles.ORG3
                    set(handles.uitable1,'Data',sortrows(NINFO,3));
                case handles.ORG4
                    set(handles.uitable1,'Data',sortrows(NINFO,-4));
            end;
        end;
        function exportdatabutton_Callback(hObject, eventdata, handles)
        fname=['ORCID Data output ' handles.fname(1:end-4) ' ' datestr(today) '.xls'];
        fpath=handles.path;
        try
            [fname fpath]=uiputfile([fpath '\' fname]);
            temp=get(handles.uitable1,'Data');
            Ea=mean(temp(:,1));
            Es=std(temp(:,1));
            Oa=mean(temp(:,2));
            Os=std(temp(:,2));
            Va=mean(temp(:,3));
            Vs=std(temp(:,3));
            Aa=mean(temp(:,4));
            As=std(temp(:,4));
            Pa=mean(temp(:,5));
            Ps=std(temp(:,5));
            I=max(temp(:,6));
            A=cat(1,{handles.fname(1:end-4) [] [] [] [] []},...
                {'Averages' [] [] [] [] []},...
                {'Eccentricity' 'Orientation' 'Value' 'Area' 'Perimeter' 'total clusters'}
                {Ea Oa Va Aa Pa I},...
                {'Std Deviations' [] [] [] [] []},...
                {Es Os Vs As Ps []},...
                {'frame' 'Upper R' 'Lower R' 'Thresh' 'edge Rem. pix' 'um/pix'},...
                {num2str(floor(get(handles.frameslider,'Value')*(handles.numframes-1))+1).
                handles.upperR handles.lowerR handles.threshold round(handles.edgemask) ha
                {'Eccentricity' 'Orientation' 'Value' 'Area' 'Perimeter' 'ID'},...
                num2cell(temp(temp(:,4)>5,:)));
            [FD GH]=xlswrite([fpath fname],A); %#ok<*ASGLU>
            disp(GH.message)
        catch error
            msgbox('Warning! The file was not saved. Maybe you dont have Administrator pri
            disp(error);
        end;
        function ROIbutton_Callback(hObject, eventdata, handles)
        BB=questdlg('Draw the ROI and right click -> create mask to finish','Region of Int
        %uiwait(BB);
        tempvar=size(handles.IMAGEINPUT);
        if strcmp(BB,'Clear previous ROI')==1;
            handles.ROImask=zeros(tempvar(1),tempvar(2))+1;
            guidata(hObject,handles);
            return;
        end;
        if strcmp(BB,'Cancel')==1; guidata(hObject,handles); return; end;
        AA=figure;
        copyobj(handles.axes1,AA);
        handles.ROImask=roipoly();
        close(AA);
        guidata(hObject,handles);
        BEF=handles.autoid;
        handles.autoid=1;%oops, this is just to make the following function work.
        handles=Update_All_Data(handles);
        handles.autoid=BEF;
        guidata(hObject,handles)
        function edgeremovetext_Callback(hObject, eventdata, handles)
        handles.edgemask=str2double(get(handles.edgeremovetext,'String'))/handles.mikestop
        handles=Update_All_Data(handles);
        guidata(hObject,handles)
        function edgeremovetext_CreateFcn(hObject, eventdata, handles)
        if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
            set(hObject,'BackgroundColor','white');
        end
        function figure1_ResizeFcn(hObject, eventdata, handles)
        figpos=get(gcf,'Position');
        pos1=get(handles.uipanel1,'Position');
        pos2=get(handles.uipanel2,'Position');
        pos3=get(handles.uipanel3,'Position');
        pos4=get(handles.uipanel4,'Position');
        pos5=get(handles.uipanel5,'Position');
        pos7=get(handles.uipanel7,'Position');
        pos1(3)=figpos(3)-pos2(3)-pos7(3);
        if pos1(3)<1; pos1(3)=5; end;
```

```
pos1(2)=pos5(2)+pos5(4);
pos1(4)=figpos(4)-pos1(2);
set(handles.uipanel1,'Position',pos1)
set(handles.axes1,'Position',[4 4 pos1(3)-4 pos1(4)-4])

pos2(1)=pos1(1)+pos1(3);
pos2(2)=pos1(2)+pos1(4)-pos2(4);
set(handles.uipanel2,'Position',pos2)

pos4(1)=pos1(1)+pos1(3);
pos4(2)=pos2(2)-pos4(4);
pos4(3)=pos2(3);
set(handles.uipanel4,'Position',pos4)

pos3(1)=pos1(1)+pos1(3);
pos3(2)=pos4(2)-pos3(4);
pos3(3)=pos4(3);
set(handles.uipanel3,'Position',pos3)

pos7(1)=pos1(1)+pos1(3)+pos2(3);
pos7(2)=pos5(2)+pos5(4);
pos7(4)=figpos(4)-pos5(4);
if pos7(4)-29.58-3>3;
    set(handles.uipanel7,'Position',pos7)
    set(handles.axes2,'Position',[5 pos7(4)-25.58 pos7(3)-5 25.58])
    set(handles.uipanel8,'Position',[1 pos7(4)-32.58 pos7(3)-5 5])
    set(handles.uitable1,'Position',[1 3 pos7(3)-5 pos7(4)-29.58-3]);
end;


pos5(3)=pos2(1);
set(handles.uipanel5,'Position',pos5)
set(handles.frameslider,'Position',[pos5(1)+2,pos5(2)+1 pos5(3)-4 pos5(4)-2])
set(handles.slidertext,'Position',[pos5(1)+2,pos5(2)+0.1 pos5(3)-4 pos5(4)-3])
set(handles.uipanel10,'Position',[pos2(1) pos5(2) pos7(3)+pos7(1) pos5(4)])
function mikestopixtext_Callback(hObject, eventdata, handles)
handles.mikestopix=str2double(get(hObject,'string'));
guidata(hObject,handles);
function mikestopixtext_CreateFcn(hObject, eventdata, handles)
% hObject    handle to mikestopixtext (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgrou
    set(hObject,'BackgroundColor','white');
end
function ZstackClusterButton_Callback(hObject, eventdata, handles)
framelimit=floor(handles.numframes/2);
requests={'Number of Species','Frame range for species 1','Frame range for species
defs={'2',[ '[1 ' num2str(framelimit) ']' ],...
    ['[' num2str(framelimit+1) ' ' num2str(handles.numframes) ']'],'45' };
answer=inputdlg(requests,'Z stack Correlation',1,defs);
```

```
handles.volumelimit=eval(char(answer(4)));
eval(['handles.stackrange=' char(answer(2)) ';']);
eval(['handles.stackrange2=' char(answer(3)) ';']);
handles.numspecies=str2double(answer(1));
disp(handles.numspecies)
%handles.speciescolor(1)=[1 0 0];
%handles.speciescolor(2)=[0 1 0];
%consider doing the following for each species (1 or 2)
%also set a parameter in handles that will establish the coloring for each
%species.
%return a field that indicates the correct graph, and append the data
%correctly! Possibly make an sp1MegaCluster and an sp2MegaCluster.
handles.MegaCluster=Zspace_Megacluster(handles);
handles.datamode=1;
guidata(hObject,handles)
% reqframe=floor(get(handles.frameslider,'Value')*(handles.numframes-1))+1;
% %Creq(reqframe).list=cat(1,handles.STATS(handles.cellinds).PixelList);
% Creq(reqframe).list=[];
% Zspan_F_CT3D([handles.path handles.fname],...
%     handles.lowerR,handles.upperR,handles.threshold,handles.ROImask,...
%     1,handles.edgemask,...
%     handles.stackrange(1),handles.stackrange(2),handles.text15,...
%     reqframe,Creq)
```

Published with MATLAB® 7.13