

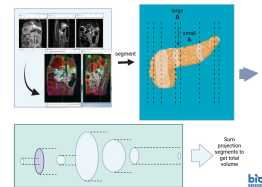
Jun 16, 2025

Version 2

Merge multiple 2D MRI datasets as multichannel stacks via ImageJ & Analyze Pancreas Size V.2

DOI

<https://dx.doi.org/10.17504/protocols.io.j8nlkrpmdv5r/v2>



Elena Carlson¹, Hoku West-Foyle¹

¹Fred Hutchinson Cancer Center



Elena Carlson

Fred Hutchinson Cancer Center

Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



DOI: <https://dx.doi.org/10.17504/protocols.io.j8nlkrpmdv5r/v2>

Protocol Citation: Elena Carlson, Hoku West-Foyle 2025. Merge multiple 2D MRI datasets as multichannel stacks via ImageJ & Analyze Pancreas Size. **protocols.io** <https://dx.doi.org/10.17504/protocols.io.j8nlkrpmdv5r/v2> Version created by **Elena Carlson**

License: This is an open access protocol distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working

We use this protocol and it's working

Created: May 31, 2025

Last Modified: June 16, 2025

Protocol Integer ID: 219245

Keywords: imageJ, fiji, mri, 3d visualization, preclinical imaging, translational imaging, mri metadata, MR Solutions, Bruker, 7T , coronal, sagittal , microsoft excel, excel, excel macro, imagej macro, axial, pancreas, pancreatic cancer, abdomen, abdominal imaging, in vivo imaging, multiple 2d mri datasets as multichannel stack, multiple 2d mri dataset, many mri dataset, analyze pancreas size many mri dataset, orthogonal mri dataset, data comparability across different mr platform, full 3d datasets due to pulse sequence limitation, weighted scan, vivo imaging, segment the pancrea, imaging, merged tiff image, full 3d dataset, certain mr pulse sequence, merged dataset, extended scan duration, segmenting region, imagej roi manager, using imagej roi manager, particular mr system, solid understanding of anatomy, analyze pancreas size, segmenting regions of interest, using anatomical fiducial marker, scan time, associated tumors from the stack, volume of the rois, boundary detection, anatomy, mr software, anatomical fiducial marker, dataset, da

Funders Acknowledgements:

National Institutes of Health

Grant ID: P30CA015704 (RRID:SCR_022616)

National Institutes of Health

Grant ID: P30CA015704 (RRID:SCR_022609)

National Institutes of Health

Grant ID: S10OD26919

Abstract

Many MRI datasets are acquired in orthogonal orientations, providing complementary perspectives that are especially useful when segmenting regions of interest. Switching between these views can enhance boundary detection and improve data analysis.

However, certain MR pulse sequences—such as T2-weighted scans—are challenging to acquire as full 3D datasets due to pulse sequence limitations or prohibitively long scan times. This is particularly problematic in in vivo imaging, where extended scan durations are not advisable, especially in preclinical studies involving anesthetized subjects with significant disease burden.

In this protocol, we outline how we:

- 1) merge multiple, orthogonal MRI datasets acquired sequentially from the same subject using anatomical fiducial markers. These merged datasets can then be imported into the analysis software of your choice for further processing.
- 2) segment the pancreas and associated tumors from the stack of merged TIFF images using ImageJ ROI Manager, save and convert these to a mask, and we calculate the volume of the rois using an Excel macro.

This approach is especially useful when machine coordinates or MR software cannot overlay datasets, or when such metadata is unavailable. It does, however, require a solid understanding of anatomy and physiology.

Our stack merging method serves as an alternative to using licensed software such as VivoQuant, and is particularly valuable in cases where coordinate metadata is inconsistently maintained during data export.

We aim to provide a flexible, system-agnostic solution that supports data comparability across different MR platforms and software environments. Some code customization may be necessary depending on how metadata is structured from a particular MR system.

Attachments

					
MRS Animal handling ...	imagej-huion.cfg	template-protocolsio...	imageJ scripts.zip	Pancreas segmentatio...	excelMacro-final.ost...
1.7MB	8.5MB	19KB	4KB	1MB	2KB
					
start-to-finish-merg...					
185.1MB					

Image Attribution

Created in BioRender. Carlson, E. (2025) <https://BioRender.com/undefined>



Guidelines

Naming conventions:

- cor= coronal otherwise known as the dorsal plane
- ax= axial, otherwise known as the transverse plane
- sag = sagittal

Materials

Required:

- MRI data (we acquired our data using a 7T MR Solutions magnet model 7017)
- ImageJ version 1.54p with the following plugins installed:
 - **TransformJ**
 - **ImageScience** library suite of plugins (in the plugin folder, not the jars folder! e.g. C:\ \ImageJ\plugins\jars)
 - Microsoft Excel and excel automation feature
 - Template excel file to follow along
 - Advanced rodent anatomical knowledge (see reference material recommendations below)

Optional but recommended:

- Huion/KAMVAS 13 tablet and associated drivers **Download Drivers And User Manual - Huion**
- "Comparative Anatomy of the Mouse and Rat: A Color Atlas and Text" from **AALAS**
- imaios annotated CT mouse data **Anatomy of the laboratory mouse: normal anatomy | vet-Anatomy**

Safety warnings

- ! MR Scans usually acquired in sequence, not simultaneous acquisition like you may be able to do for some microscopy image applications.

Breathing and gating efficiency can cause some differences in positioning and registration.

Only works for 3 datasets, any more won't work with the code.

Ethics statement

All animal experiments were done in accordance with protocols approved by the Institutional Animal Care and Use Committees of Fred Hutchinson Cancer Center (protocol no. 50898).

Acquire MR Data

- 1 We acquired data on a preclinical 7 Tesla MR scanner (MR Solutions, DRYMAG7.0T model 7017).

Animals were positioned following the guidelines in the attached "Animal handling guidelines" in the attached supporting document.

- 2 Export MRI data as DICOM files according to following naming structure:

Files must follow naming structure for Step 3.1+ below to work

T2w_sag

T2w_ax

T2w_cor

T1w_sag

T1w_cor

Run ImageJ macro "MRI_scale_overlay"

- 3 Drag and drop the .ijm file containing the following script into the imageJ toolbar:

```
3.1 //A run("Close All");  
//selectWindow("name");  
//run(close);
```

input=getDirectory("Please choose directory with T1w and/or T2w images. Must be titled appropriately.");

```
ref_voxel=newArray(0.1,0.1,0.1);
```

```
ref_canvas=newArray(260,350,260);
```

```
if(File.exists(input+"T1w_cor/")){  
    File.openSequence(input+"T1w_cor/");  
    scale_to_ref(ref_voxel,ref_canvas);  
}
```

```
if(File.exists(input+"T2w_cor/")){  
    File.openSequence(input+"T2w_cor/");  
    scale_to_ref(ref_voxel,ref_canvas);  
}
```

```
if(File.exists(input+"T2w_sag/")){
```



```
File.openSequence(input+"T2w_sag/");
scale_to_ref(ref_voxel,ref_canvas);
}
if(File.exists(input+"T1w_ax/")){
File.openSequence(input+"T1w_ax/");
scale_to_ref(ref_voxel,ref_canvas);
}

if(File.exists(input+"T2w_ax/")){
File.openSequence(input+"T2w_ax/");
scale_to_ref(ref_voxel,ref_canvas);
}

//T1w 0020,0032 Image Position (Patient): 10.962077\7.420000\19.931641
//T2w 0020,0032 Image Position (Patient): 10.539853\7.510000\19.931406
//z: 4 slices "deeper" in T1W
//Pixel dimensions: T1W has more X-pixels but fewer Z-slices. Same Y-pixels
//Sizing: T1W has physically more of the mouse in Y (appear to line up at top though)
//they line up at TOP
//T2W CLAIMS to be 0.110x, 0.117Y (insignicant)
//T1W CLAIMS to 0.1058X,0.1367Y.
//DELETE first 4 slices of T2W--BOTTOM seems to be ref. coordinate
//Ref: TOP of Y, BOTTOM of Z

function scale_to_ref(ref_voxel,ref_canvas){
name=getTitle();
// mouseName=0010,0010 Patient's Name: DE1689
meta = getImageInfo();
meta_array=split(meta,"\n");

//for loop which applies for each image set, SCANS metadata
for(i=0;i
//orientation of the image, i.e. ax, cor or sag, but there may be some slight
angles/adjustments so not always perf 90 degrees offset
//first 3 columns give you row values
//second 3 columns

//axial: "0020,0037 Image Orientation (Patient):
-0.95\ -0.32\ -0.00\ 0.32\ -0.95\ 0.00 " ; //
https://dicomiseasy.blogspot.com/2013/06/getting-oriented-using-image-plane.html
//sagittal: "0020,0037 Image Orientation (Patient):
0.30\ -0.95\ -0.00\ 0.00\ 0.00\ -1.00"
```

```
//coronal"0020,0037 Image Orientation (Patient):  
-0.95\ -0.30\ -0.00\ 0.00\ -0.00\ -1.00"  
  
    if(startsWith(meta_array[i],"0020,0037 Image Orientation (Patient)")){  
        data=split(meta_array[i],":\\");  
        print("---here is the data output-----");  
//        print(data[1]);  
  
        rowX=parseInt(data[1]);  
        colX=parseInt(data[4]);  
        rowY=parseInt(data[2]);  
        colY=parseInt(data[5]);  
        rowZ=parseInt(data[3]);  
        colZ=parseInt(data[6]);  
  
    }  
  
    //patient position - THIS MIGHT BE THE MACHINE COORDINATES? : //axial  
example:    0020,0032 Image Position (Patient): 12.99\15.29\8.42; //coronal example:  
0020,0032 Image Position (Patient): 14.80\ -4.74\13.69 ; saggital example: 0020,0032  
Image Position (Patient): -3.29\15.04\20.92  
  
    if(startsWith(meta_array[i],"0020,0032 Image Position (Patient)")){  
        data=split(meta_array[i],":\\");  
        x_position=parseInt(data[1]);  
        y_position=parseInt(data[2]);  
        z_position=parseInt(data[3]);  
    }  
  
    if(startsWith(meta_array[i],"0018,0088 Spacing Between Slices")){ //Slice  
Thickness???  
        data=split(meta_array[i],":");  
        z_scaling=parseFloat(data[1]);  
    }  
  
    if(    startsWith(meta_array[i],"0028,0030 Pixel Spacing")){  
        data=split(meta_array[i],":\\");  
        y_scaling=parseFloat(data[1]);  
        x_scaling=parseFloat(data[2]);  
    }  
    if(startsWith(meta_array[i],"0018,1310 Acquisition Matrix")){  
        print(name,meta_array[i]);  
    }  
}
```

```
        if(startsWith(meta_array[i],"0010,0010 Patient's Name")){
            print(name,meta_array[i]);
            //PatientID=meta_array[i]
        }
        if(startsWith(meta_array[i],"0008,0012 Instance Creation Date")){
            print(name,meta_array[i]);
        }
    }
    setVoxelSize(x_scaling,y_scaling,z_scaling,"mm");

    run("TransformJ Scale", "x-factor="+x_scaling/ref_voxel[0]+" y-
factor="+y_scaling/ref_voxel[1]+" z-factor="+z_scaling/ref_voxel[2]+" interpolation=
[Quintic B-Spline] preserve");
    close(name);

    //axial matrix = "-1,-0,-0,0,-1,0"
    //sag matrix = "0,-1,-0,0,0,-1"
    //cor matrix = "-1,-0,-0,0,-0,-1"negative zeros are NOT maintained by the
    parseInt function above;

    //sagittal rotation
    if(matches(rowX+","+rowY+","+rowZ+","+colX+","+colY+","+colZ, "0,-1,0,0,0,-1")){ //
        pre=getTitle();
        run("TransformJ Turn", "z-angle=0 y-angle=90 x-angle=0"); //y 90 for sag
        close(pre);
        rename(pre);
        //add little regostration marker in corner

    makeText("sag", 30, 20);
    setColor(255, 165, 0); // Orange
    run("Add Selection...");
    run("Overlay Options...", "stroke=none fill=orange draw");
    //run("Flatten", "stack");

    }

    // axial
    if(matches(rowX+","+rowY+","+rowZ+","+colX+","+colY+","+colZ, "-1,0,0,0,-1,0")){
        pre=getTitle();
```



```
run("TransformJ Turn", "z-angle=0 y-angle=0 x-angle=270"); // z 0; y90 looks
sag now... , x =90? right
close(pre);
rename(pre);

makeText("ax", 30, 20);
setColor(255, 165, 0); // Orange
run("Add Selection...");
run("Overlay Options...", "stroke=none fill=orange draw");
//run("Flatten", "stack");

}

//cor rotate or just add label
if(matches(rowX+", "+rowY+", "+rowZ+", "+colX+", "+colY+", "+colZ, "-1,-0,-0,0,-0,-1")){
    //      //pre=getTitle();
    //      //run("TransformJ Turn", "z-angle=0 y-angle=0 x-angle=0"); // no rotation
as it is here - if you need to, adjust code in this line
    //      //close(pre);
    //      //rename(pre);
    //

makeText("cor", 30, 20);
setColor(255, 165, 0); // Orange
run("Add Selection...");
run("Overlay Options...", "stroke=none fill=orange draw");
//run("Flatten", "stack");

}

run("Canvas Size...", "width="+ref_canvas[0]+" height="+ref_canvas[1]+"
position=Top-Center zero");
setSlice(round(nSlices/2));
resetMinAndMax;
setMetadata("Info", meta);

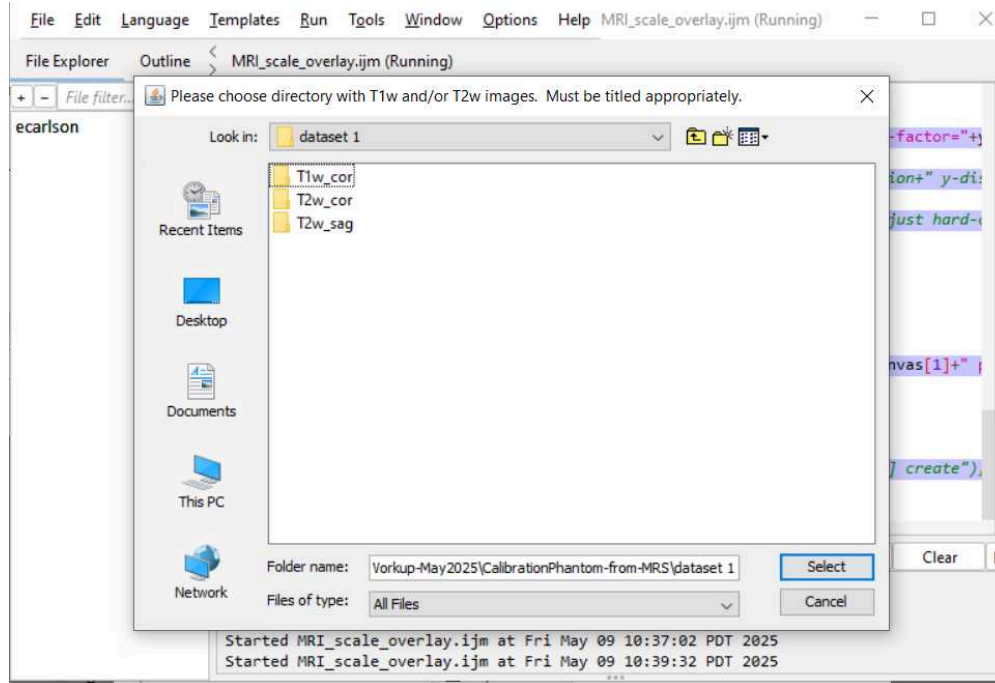
//add little marker in corner

makeText("R, H", 215, 20);
setColor(255, 165, 0); // Orange
run("Add Selection...");
```

```
run("Overlay Options...", "stroke=none fill=orange draw");

//flatten so the label follows any further stack manipulations
//waitForUser("keep registration label?", "flatten?");
run("Flatten", "stack");
}
```

- 4 Run & Select parent folder containing a list of exported dicom files with naming structure as described in step 2 above



Example of how to select parent folder, containing list of dicom exports with naming structure as shown

4.1 WARNING: BEFORE SELECTING PARENT FOLDER YOU SHOULD

Note which study ID and timepoint you have used as this will not be conserved in the metadata after processing unless you save the log files.

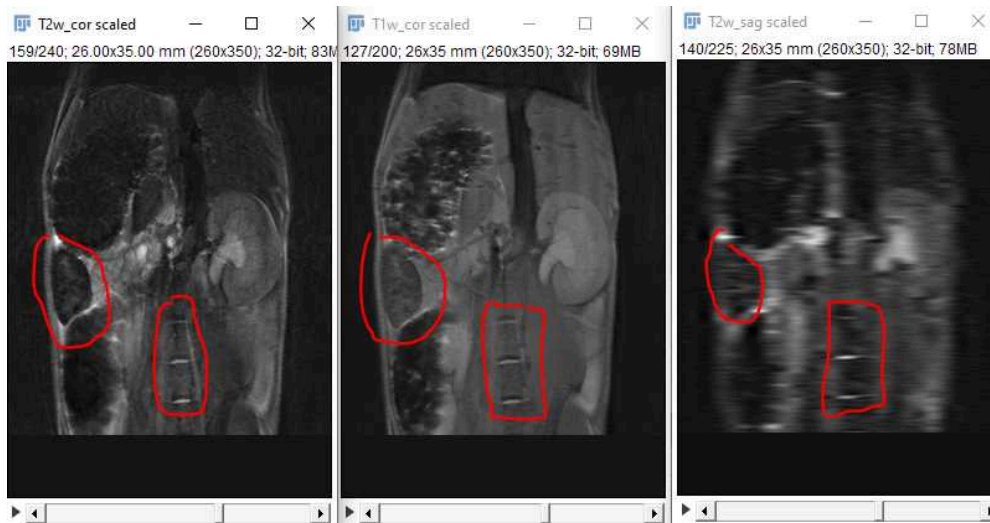
(The Log will output the "Patient name" and Date acquired)

You can use the attached template Excel File to keep track.

Fiducial mark registration to co-register slices between orthogonal, processed, rotated stacks

- 5 Locate an anatomical reference obvious between both stacks (e.g. a particular vertebrae and spleen, or gallbladder and stomach) or some other anatomical fiducial marker.

note: Perfect co-registration may not be possible. We will account for this later.



Calculate delta

6 Calculate Delta for the T2 stacks

Note the slice location in each stack. Put into excel template (columns F, G, and/or H) to calculate "Delta" (column I in Template Excel File).

7 IF DELTA IS POSITIVE

Duplicate "Ctrl+Shift+D" the **T2w_sag scaled** stack
Set Range as: **Delta** - End of Stack

7.1 Verify that fiducial marker for new stack is at same slice location between the T2w_sag scaled-1 stack and the T2w_cor scaled stacks

8 IF DELTA IS NEGATIVE

Duplicate "Ctrl+Shift+D" the **T2w_cor scaled** stack
Check box and set **Range: [Absolute Value(Delta)] - [End of Stack]**

Verify that Fiducial Marker for T2w_cor scaled-1 is at same slice location between the T2w_cor scaled-1 stack and the T2w_sag scaled stacks.

Equalize Num Slices between T2 stacks



9 Do this step if all 3 stacks have a different number of slices

- This step is *not necessary* if *two of the three* stacks are equal in length at this point however to skip this step, the sagittal scan stack must be equal in length to one of the other coronal stacks!

Calculate n

$n = \text{total Num Slices[Bigger Stack]} - [\text{TotalNumSlices (Smaller Stack)}]$

Duplicate the T2w stack that has MORE total slices

Set **Range:** 1 - [(End of Stack)-n]

Check box

- 9.1 Verify stacks have same number of slices
Verify fiducial markers are registered once again

- 9.2 To prevent any confusion, delete T2w_cor scaled and T2w_sag scaled here.

KEEP "T2w_cor scaled-1" and "T2w_sag scaled -1" or possibly the "-2" stacks

Third Stack Sync Up

- 10 *If the fiducial marks are aligned between the 3 stacks, and all 3 stacks are the same length, you can skip to step 12 below.*

Calculate Delta-Stack3 and LastSlice (columns Q and O respectively in template Excel sheet)

Delta-Stack3 = Fiducial Mark Location for the two aligned stacks *as determined above in step 9*

if Delta-Stack3 is NOT Zero:

LastSlice= LargestStack + Difference(Fiducial Marker Location of the two processed stacks and Fiducial mark location in T1 channel) - Difference in the Stacks Length

if Delta-Stack3 = 0:

LastSlice= LargestStack + Difference(Fiducial Marker Location of the two processed stacks and Fiducial mark location in T1 channel) - Difference in the Stacks Length -1

- 11 Image > Stacks > tools > **Slice Keeper** on (*each of*) the larger stack(s)
First Slice: Delta-Stack3
Last Slice: LastSlice
Increment: 1



- 11.1 Verify that all three stacks have same fiducial mark locations AND are the same stack length!

Merge 3 Stacks & correct for registration offsets

- 12 **Run this .ijm macro script - process used is described step by step below in steps 12.1-13**



```
//B-Hyperstack-3D drift correction - unhyperstack

run("Merge Channels...")
//waitForUser("merge your channels","OK to proceed");

selectImage("Composite");

run("Re-order Hyperstack ...", "channels=[Frames (t)] slices=[Slices (z)] frames=[Channels (c)]");
run("Enhance Contrast", "saturated=0.35");

selectImage("Composite");

if(nSlices/3==200){
    run("Correct 3D drift", "channel=1 only=0 lowest=1 highest=200
max_shift_x=10.000000000 max_shift_y=10.000000000 max_shift_z=10.000000000");
} else {
    run("Correct 3D drift");
}

//if large xy offset, run line 13 below instead of line 10
//run("Correct 3D drift");

selectImage("registered time points");
run("Re-order Hyperstack ...", "channels=[Frames (t)] slices=[Slices (z)] frames=[Channels (c)]");
//for (c = 1; c <= channels; c++) {
//    Stack.setChannel(c);
//    // Apply LUT from the array (wrap around if more channels than LUTs)
//    run("Grays");
//resetMinAndMax;
//}
```

```
saveLocationB = getDirectory("Where do you want to save the data?");
```

```
run("Tiff...",saveLocationB);
```

Alternative to using Step B imageJ macro

12.1 If you do not use the macro, here are the steps it performs:

Merge Channels (Image > Color > Merge Channels)

Scroll through and verify approximate (+/- 10 slices) registration overlay between colors

(you can further verify with Ctrl + Shift + H to get Sagittal scan at better resolution)

12.2 Reorder hyperstack

Set:

Channels (c) → **Frames (t)**

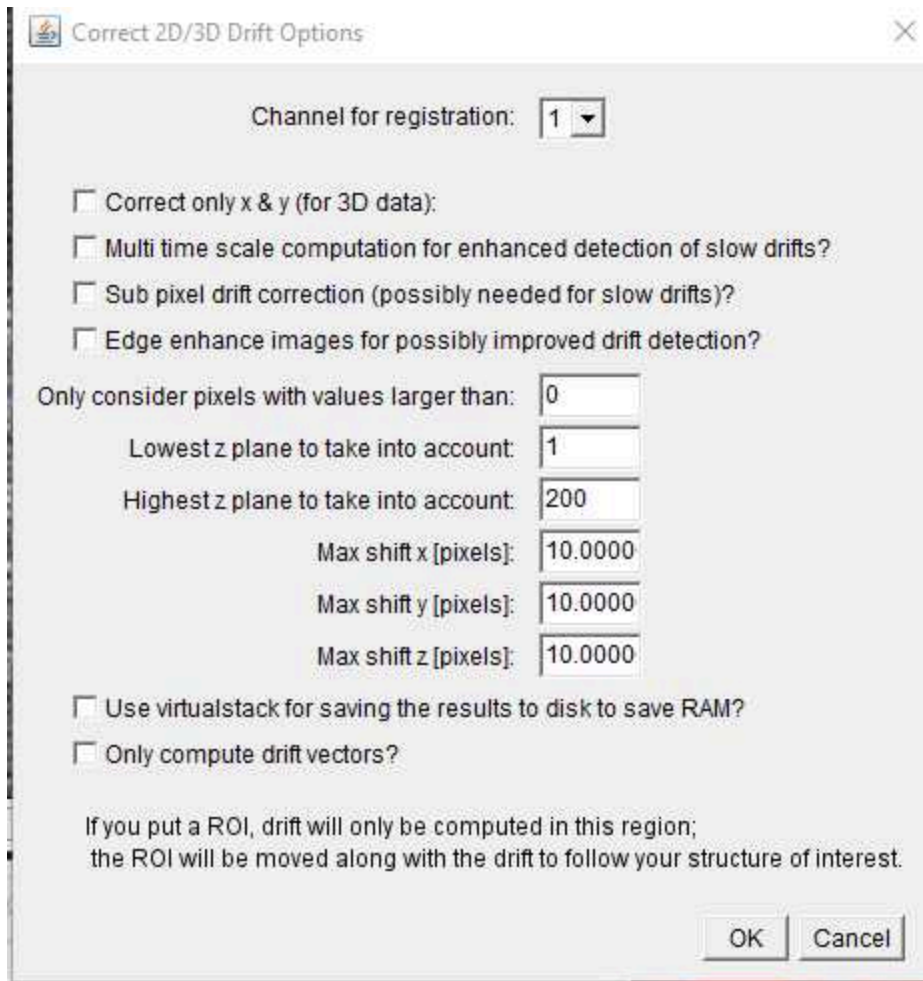
Slices (z) → Slices (z)

Frames (t) → **Channels (c)**

12.3 CORRECT 3D DRIFT

Plugins > Registration > Correct 3D Drift

leave default settings



Default Correct 3D Drift Plugin Settings

For big xy drift between sagittal and coronal stacks, may need to adjust to 20 or 30 instead of the default values of 10. When this is the case, I make a note of it for reference in further analysis steps.

12.4 Reorder hyperstack

Set

Channels (c) → **Frames (t)**

Slices (z) → Slices (z)

Frames (t) → **Channels (c)**

Verify that the stack is a Color stack with channels, **not** TIMEPOINTS

- 13 **Save** As .tiff , and if not proceeding to part 2, you can load this data into another software for further post processing.

Pancreas measurement and analysis

- 14 In steps 3-13 we saved the TIFFS with preference to the coronal stacks.
To preference the sagittal view, this macro was developed so that measurements could be primarily done in the sagittal plane.

Macro to rotate and open ROI Manager tools

- 15 Open the merged TIFF stack in ImageJ.
- 16 Run the ImageJ macro: C-Process-Merged-TIFFs.ijm

- 16.1 this script:
- Converts channels to grayscale,
 - Opens Channel 2.
 - Outputs width, height, depth in the Log (this is not saved as a default, however)
 - Rotates the stack with preference to the sagittal view
 - Asks user to set a save destination for the rotated stack, and save the stack

The .ijm script contents are listed below in substep 16.2:

16.2 **//C-process merged tiffs.ijm script**

```
getDimensions(width, height, channels, slices, frames);
print("Number of channels: " + channels);
print("Number of slices: " + slices);
print("width: " + width);
print("height: " + height);
run("TransformJ Rotate", "z-angle=0 y-angle=90 x-angle=0 interpolation=Linear
background=0.0 adjust");
```

```
getDimensions(width, height, channels, slices, frames);
print("Number of channels TransformJ: " + channels);
print("Number of slices TransformJ: " + slices);
print("TransformJ width: " + width);
print("TransformJ height: " + height);
```

```
// Loop through each channel
```



```
for (c = 1; c <= channels; c++) {  
    Stack.setChannel(c);  
    // Apply LUT from the array (wrap around if more channels than LUTs)  
    run("Grays");  
    resetMinAndMax;  
}  
  
saveAs(".tiff")  
run("ROI Manager...");  
  
run("Channels Tool...");  
  
Stack.setActiveChannels("010");  
Stack.setChannel(2);  
Stack.setFrame(100);  
  
setTool("polygon");  
roiManager("Show All with labels");  
roiManager("Show All");
```

Anatomical tracing

- 17 Scroll through the stack to identify the boundaries of the pancreas, and any other relevant organs (especially stomach, spleen, gallbladder, and the kidneys).



Be very mindful of the intestines. If something looks like a "rolled up sock" when you scroll around, it is likely part of the intestines, not the pancreas.

Note that there are two inguinal lymph nodes, and mesenteric lymph nodes which can be mistaken as tumors as well.

On the mouse's right side: be mindful of the greater omentum, jejunum, & duodenal ampulla.

Trace the sections of the Pancreas in sagittal stack

- 17.1 Use FIJI's orthogonal views function to verify anatomical structures.
Be mindful that there may be some co-registration errors between channels.
- 17.2 Adjust image contrast as needed (*Ctrl + Shift + C*)



18 Use ROI Manager and the polygon tool to begin tracing.

18.1 You do NOT need to trace every slice but you do need to trace every part of the pancreas and tumors when you DO select a slice

19 Verification and cross checking your ROIs



19.1 Use Sort to scroll through ROIs of the pancreas in order. This is in **ROI Manager > More > Sort**

19.2 Re-verify the sections via sagittal and coronal views to verify consistency and anatomical accuracy, while tabbing through ROIs in ROI manager.

Ensure ROIs are well-sampled where there are large changes in structure/shape as you scroll across the stack.

19.3 Delete duplicate or incorrect ROIs, or update existing ROIs after cross checking across the different orientations via ImageJ's orthogonal views feature.

Excel Macro for volume measurements

20 Run the D-ROIman.ijm ImageJ macro. This will prompt an automatic save of a "results" .csv file

.ijm script contents listed below in substep 20.1:

20.1 `//D-RoiManager
// Prompt user for save location
saveLocationD = getDirectory("Where do you want to save the Log, ROI zips, roi mask,
and roi measurement results?");

// Select all ROIs
roiManager("Deselect");
roiManager("Sort"); // ensures will load for excel macro properly

// Save ROIs in zip file
roiManager("Save", saveLocationD + "rois.zip");

// Set measurement options`



```
run("Set Measurements...", "area stack display add redirect=None decimal=3"); // if you
change these the excel macro won't work as is
//run("Set Measurements...", "area stack mean standard modal min perimeter shape
integrated median kurtosis display add redirect=None decimal=3");

// Measure selected ROIs with above measurement choices
roiManager("Measure");

//saveAs("Measure", saveLocation);
run("Text...", saveLocationD);

//incorporating E
// duplicate the stack and make a new 8-bit "channel" in binary which fills in all the ROIs
from the ROI manager.
roiManager("Deselect");
run("Duplicate...", "duplicate channels=1");
run("8-bit");
run("Select All");
run("Clear", "stack");
setBackground(0, 0, 0);
setForegroundColor(255, 255, 255);
setBackground(0, 0, 0);

rois=roiManager("count");
for(i=0;i<roiManager("count");i++)roiManager("Select", i);
run("Fill", "slice");
}

saveAs(".tiff", saveLocationD+"mask-of-rois");

//close all windows
waitForUser("OK to close all windows EXCEPT results window?"); //if log file, won't close
yet till very end

close("*")

selectWindow("ROI Manager");
run("Close");
selectWindow("B&C");
run("Close");
selectWindow("Channels");
run("Close");
//in case you want to cross compare excel output to results window before closing it
```

```
waitForUser("OK to close results window?");

selectWindow("Results");
run("Close");

//if log window exists, this will run, otherwise it will error out

selectWindow("Log")
run("Text...", saveLocationD+"Log")
selectWindow("Log")
run("Close");
```

21 Open the .csv file in Excel and Save it as an excel workbook

22 Run the excel macro "excelMacro-final.osts"

22.1 Open the Excel workbook and **ensure cell A1** is selected before running the macro, which is operates on a relative basis and will error out of A1 is not selected.

Go to automations tab in excel and run the excel macro script, whose contents are described below in substep 22.2

22.2

```
function main(workbook: ExcelScript.Workbook) {
  let selectedCell = workbook.getActiveCell();
  let selectedSheet = workbook.getActiveWorksheet();
  // Insert at 1 row(s) on selectedSheet offset by 0 row(s) relative to selectedCell, move
  existing cells down
  selectedCell.getEntireRow().insert(ExcelScript.InsertShiftDirection.down);
  // Insert at 1 row(s) on selectedSheet offset by 0 row(s) relative to selectedCell, move
  existing cells down
  selectedCell.getEntireRow().insert(ExcelScript.InsertShiftDirection.down);
  // Set 3 by 1 range on selectedSheet offset by 2 row(s) and 3 column(s) relative to
  selectedCell
  selectedCell.getOffsetRange(2, 3).getAbsoluteResizedRange(3,
1).setFormulasLocal(["num slice in each segment"],["1"],["=IF(E5-E4>1,E5-E4,1)"]);
  // Paste to cell on selectedSheet offset by 3 row(s) and 3 column(s) relative to
  selectedCell from cell on selectedSheet offset by 4 row(s) and 3 column(s) relative to
  selectedCell
  selectedCell.getOffsetRange(3, 3).copyFrom(selectedCell.getOffsetRange(4, 3),
ExcelScript.RangeCopyType.all, false, false);
  // Set cell on selectedSheet offset by 3 row(s) and 3 column(s) relative to selectedCell
```

```
selectedCell.getOffsetRange(3, 3).setFormulaLocal("=IF(E4-E3>1,E4-E3,1)");
// Auto fill range
selectedCell.getOffsetRange(4, 3).autoFill("D5:D308",
ExcelScript.AutoFillType.fillDefault);
// Set 2 by 1 range on selectedSheet offset by 2 row(s) and 5 column(s) relative to
selectedCell
selectedCell.getOffsetRange(2, 5).getAbsoluteResizedRange(2,
1).setFormulasLocal(["Volume per segment (mm^3)",["=D4*C4*R1C5"]]);
// Set 1 by 3 range on selectedSheet offset by 0 row(s) and 2 column(s) relative to
selectedCell
selectedCell.getOffsetRange(0, 2).getAbsoluteResizedRange(1, 3).setValues(["Voxel
size (mm^3) =", "", 0.1]);
// Set cell on selectedSheet offset by 3 row(s) and 3 column(s) relative to selectedCell
selectedCell.getOffsetRange(3, 3).setValue("1");
// Auto fill range
selectedCell.getOffsetRange(3, 5).autoFill("F4:F308",
ExcelScript.AutoFillType.fillDefault);
// Set 3 by 1 range on selectedSheet offset by 7 row(s) and 7 column(s) relative to
selectedCell
selectedCell.getOffsetRange(7, 7).getAbsoluteResizedRange(3,
1).setFormulasLocal(["total Volume (mm^3)",[""],["=SUM(F:F)"]]);
// Set cell on selectedSheet offset by 4 row(s) and 3 column(s) relative to selectedCell
selectedCell.getOffsetRange(4, 3).setFormulaLocal("=IF(E5-E4>1,E5-E4,1)");
// Set cell on selectedSheet offset by 8 row(s) and 3 column(s) relative to selectedCell
selectedCell.getOffsetRange(8, 3).setFormulaLocal("=IF(E9-E8>1,E9-E8,1)");
// Set cell on selectedSheet offset by 23 row(s) and 5 column(s) relative to selectedCell
selectedCell.getOffsetRange(23, 5).setFormulaLocal("=D24*C24*R1C5");
// Set width of column(s) at 1 column(s) on selectedSheet offset by 7 column(s) relative
to selectedCell to 15.73
selectedCell.getOffsetRange(0,
7).getEntireColumn().getFormat().setColumnWidth(15.73);
// Select cell on selectedSheet offset by 15 row(s) and 7 column(s) relative to
selectedCell
selectedCell.getOffsetRange(15, 7).select();
}
```

22.3 The macro will multiply area by number of slices with voxel depth (e.g., 0.1) to get volume per segment.

22.4 After the script runs, verify that the slice numbers are in ascending order.

22.5 verify that there are no zero values in slice counts. Change the zeros to 1 for consecutive slices (e.g. you had multiple distinct rois on the same slice, just make sure the first in the series indicates the difference from the previous).

you can use conditional formatting in Excel to visually verify data integrity.

22.6 Compare with ImageJ measurements window

23 Sum all segment volumes to get total pancreas/tumor volume.

24 Save and back up all data.

Protocol references

- 1) *Comparative Anatomy of the Mouse and Rat: A Color Atlas and Text* by Gheorghe M Constantinescu, DVM, PhD, mult. Drhc. **AALAS** <https://www.aalas.org/products/2368>
- 2) **Anatomy of the laboratory mouse: normal anatomy | vet-Anatomy** <https://www.imaios.com/en/vet-anatomy/mouse/mouse-whole-body>
- 3) "Getting Oriented using the Image Plane Module" web article from **DICOM is Easy**: Software Programming for Medical Applications - <https://dicomiseasy.blogspot.com/2013/06/getting-oriented-using-image-plane.html>



Acknowledgements

This research was supported by the Preclinical Imaging, Comparative Medicine, and Cellular Imaging Shared Resources (**RRID:SCR_022616** , **RRID:SCR_022609**, RRID:SCR_022610) of the Fred Hutch/University of Washington/Seattle Children's Cancer Consortium (P30 CA015704) and the 3T/7T MRI Shared Instrumentation Grant NIH S10OD26919.

We thank Dr. Alex Hicks-Nelson, DVM, MS for constructive conversations about anatomy and physiology. Thank you to Kiersten Maldonado from MD Anderson Cancer Center for helpful advice about pancreas identification and data analysis pipelines for similar datasets acquired at their institution.

We also acknowledge the work of Brianna Wrightson and Robert Espinoza in acquiring the MR datasets, and the scientific support team at MR Solutions for their assistance in developing pulse sequences- primarily Marco Zampini, PhD.

Graphical abstract created in BioRender. Carlson, E. (2025) <https://BioRender.com/9b1zhou>