From Movie to 3D Space…

You should have a folder full of DV files.

Folder starting with 1,

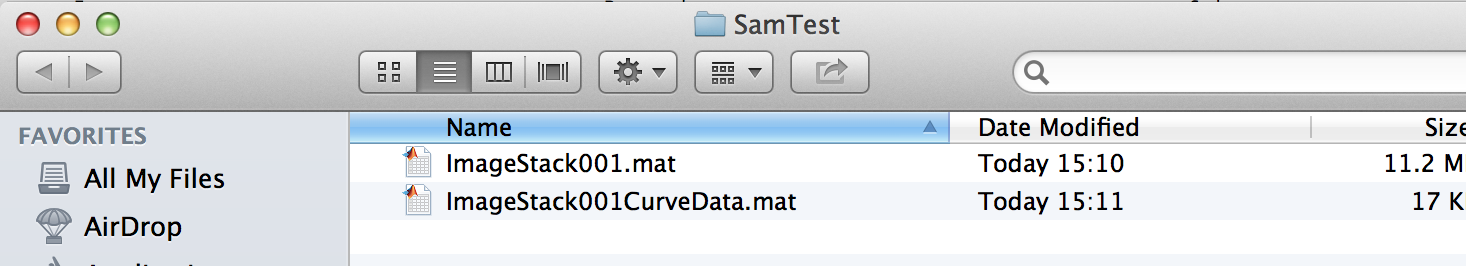
RunMeFirst.

* Index of file (which of the files you want to do
* Start of frame, end of frame
* Path (that can be reg expression)

DV becomes a matlab variables; This one does the segmentation . which can take quite a while. As a rule of thumb, it 1500

* RunMeFirst(1, 11,12, '/Users/iasmam/Desktop/SamJ\_Code\_and\_Data/DemoData/GLA6\_exp3\_220713\_untreated01\_114\_R3D.dv','/Users/iasmam/Desktop/SamTest' )
* Movies should have all the same number of frames (assumptions)

After this step, we have for each file the following files



# Macintosh HD:Users:iasmam:Desktop:Screen Shot 2015-02-06 at 15.28.52.png

This are the two parameters directly in the code, that influence how the segmenation is done.

Rangeband= how much noise it is picking up  
spagtialband= how large a cell is (say 3 is the readious… if a cell is large, say always larger than 10, you might want 10, as the smallest cluster).

We can give a good default (as shown in the image).

The next step involves opening a guid

In the folder 1-ImageSementationFull there is a file called “CSG\_v2.m”

Open in Matlab this file and start it. You should see:



Click on “New Experiment”

And enter the path, where you have been saving things, in all three boxes.



If you are a programmer, the 3 paths would make sense, or if you want to do some advanced output. Otherwise, if you have just been following this tutorial, you have to put 3 times the same path in.

If you were successful, you should see a dialog window, that should look like this: Important: it should have a number!



# Now to open a stack. You have to enter the experiment number and the number of the stack (see below)

# Macintosh HD:Users:iasmam:Desktop:Screen Shot 2015-02-07 at 14.42.13.png

# If everthing was working out. You should see your stack, with the outline of the cells.



This interface permits you to manually correct things. Say I would like to remove the small detected cell here, I would Shft+Click on it, and then save the data. Saving the data, means that I generate 3 new files for the stack. CewllArray001.mat

CellFrameData001.mat

Handledata\_exp1.Stack.mat



# Options in the GUI

* You can delete cell shapes (also froma specific point)
* You can merge cells (even with merge point)
* There are instruction on the movie.

You can toggle off the short lived ones, and you can toggle of the small ones. Be careful as the two annul each other. Which is pretty annoying.

You can only load one stack at the time.  
You save them in your folder.

# Once all your files are done

Say, you have corrected all files, then you are ready for the next step: in Folder 1-ImageSegmantation Full, there is the file MakeBigStructs.m

Open it in Matlab and click on the “Run” Button.



There should be an explorer that opens, and you should enter as the directory, the experminet directory you have been using.



After selecting the folder, the program should create 2 new files in the folder, namely Bigcellarrayandindex.mat and BigCellDataStruct.mat (see below).



Now you are ready for the next processing step. You go to the Folder   
“2-ShapeManifoldEmbedding”

And open the script “RunShapeManifoldEmbedding.m” in Matlab. Click on the “Run” Button in Matlab, and the script should ask you for experiment folder (as before). Select the folder and when the script complets correctly, you should see the following dialog box.



Also, if the program run correctly, you should have the file “CellShapeData.mat” in your experiment folder.

Now you are ready to move to the steps in Folder

# 3 Extended Affinity Propagation

This step will look into your sample and look for similar shape classes. For running the affinity propagation, all necessary files are in the folder   
“3-Extended\_Affinity\_Propagation”

The first preparatory stop consists in running the Affinity progagation. To do so, you open the file “Run\_Affinnity\_Progagation.m” in Matlab and click “Run”. The program should ask for your Experiment Folder where you have saved the files of the previous step. After selecting the folder, the program should start automatically and on the console write the progress. When the program finishes, you should see progress bars as below.



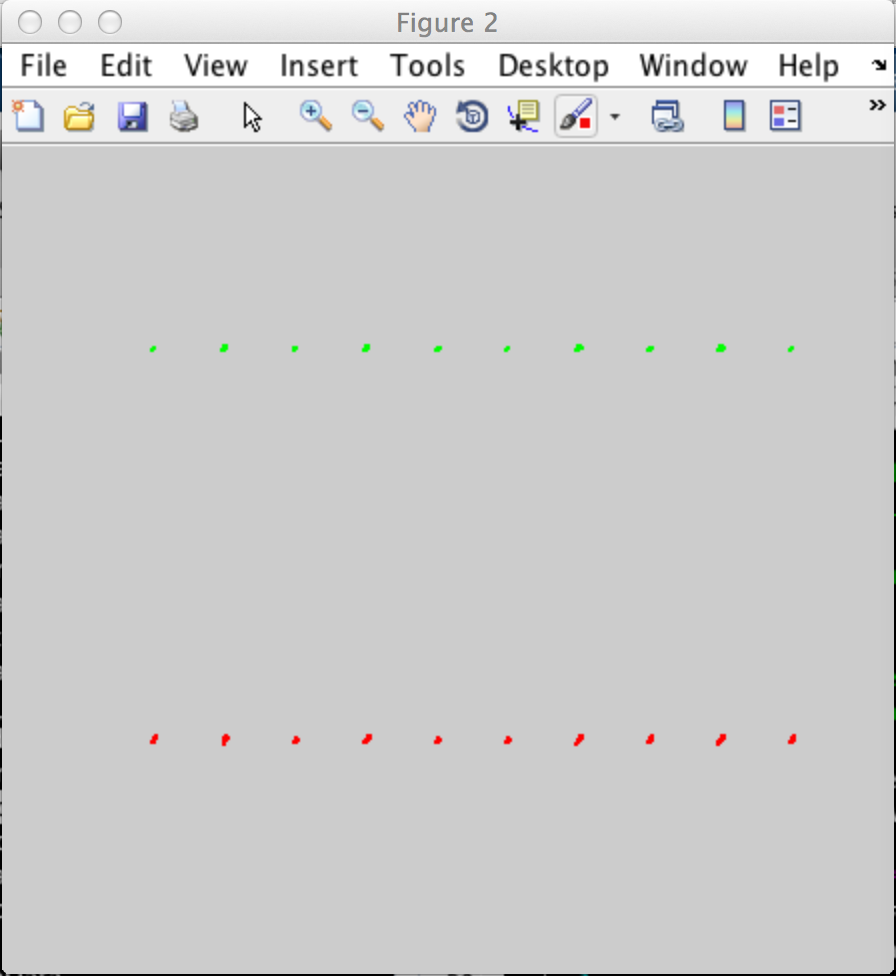
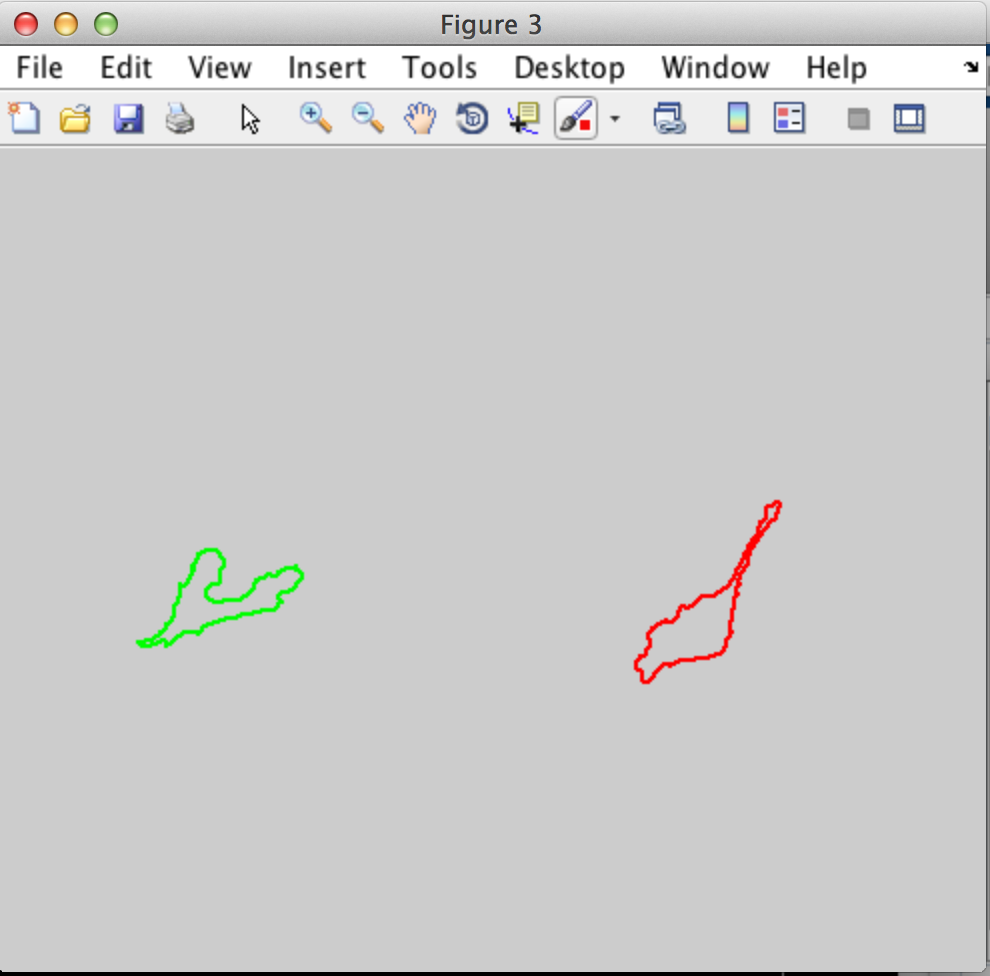
Once the program finishes successful, there should be nine new files in the folder, which should have the following names:



Inspect the

Now we are ready to

Open “unordered\_list.m” and click on Run. Again select your experiment folder, and then the program should run through. You should get 2 figures.



You can save vector images by clicking on File>Save As…

Ordered List.

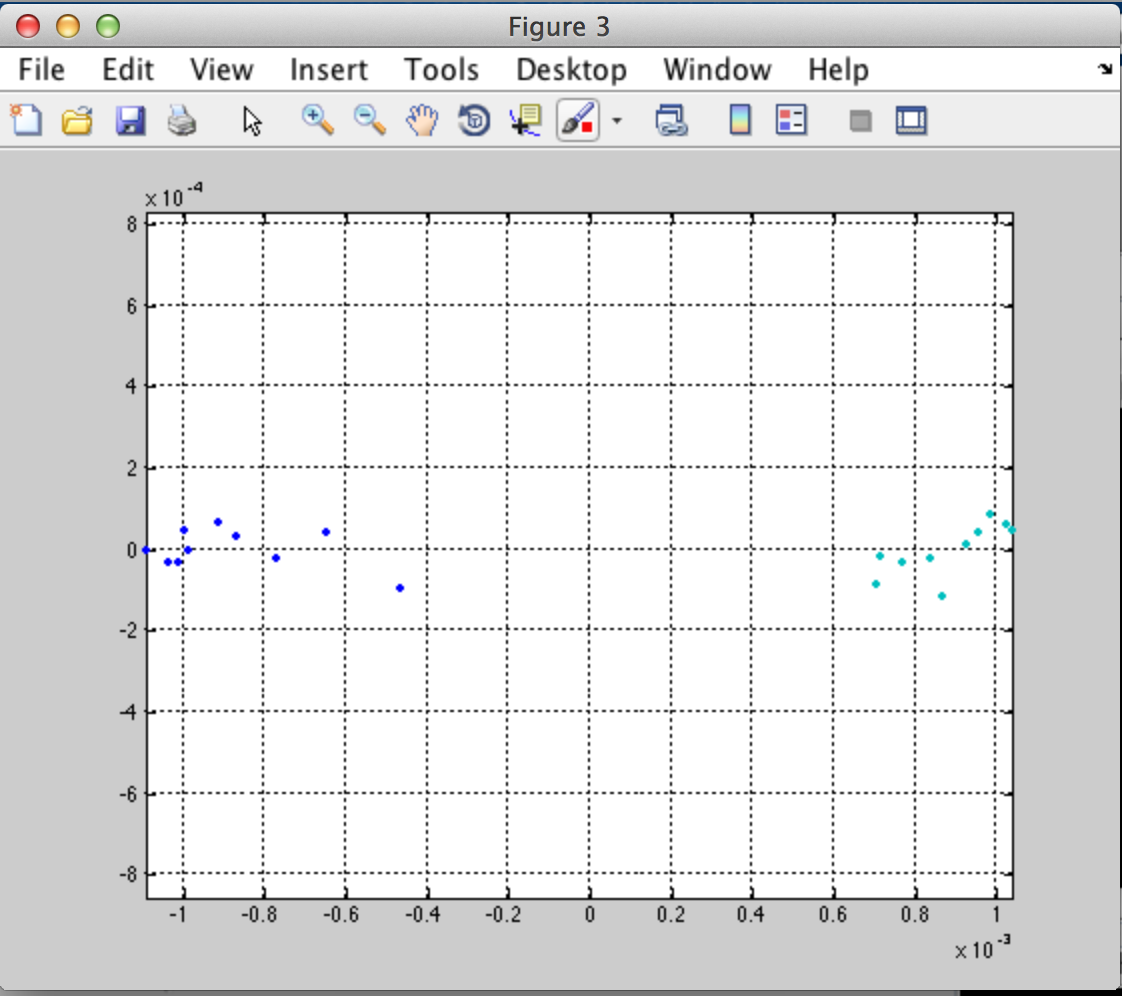
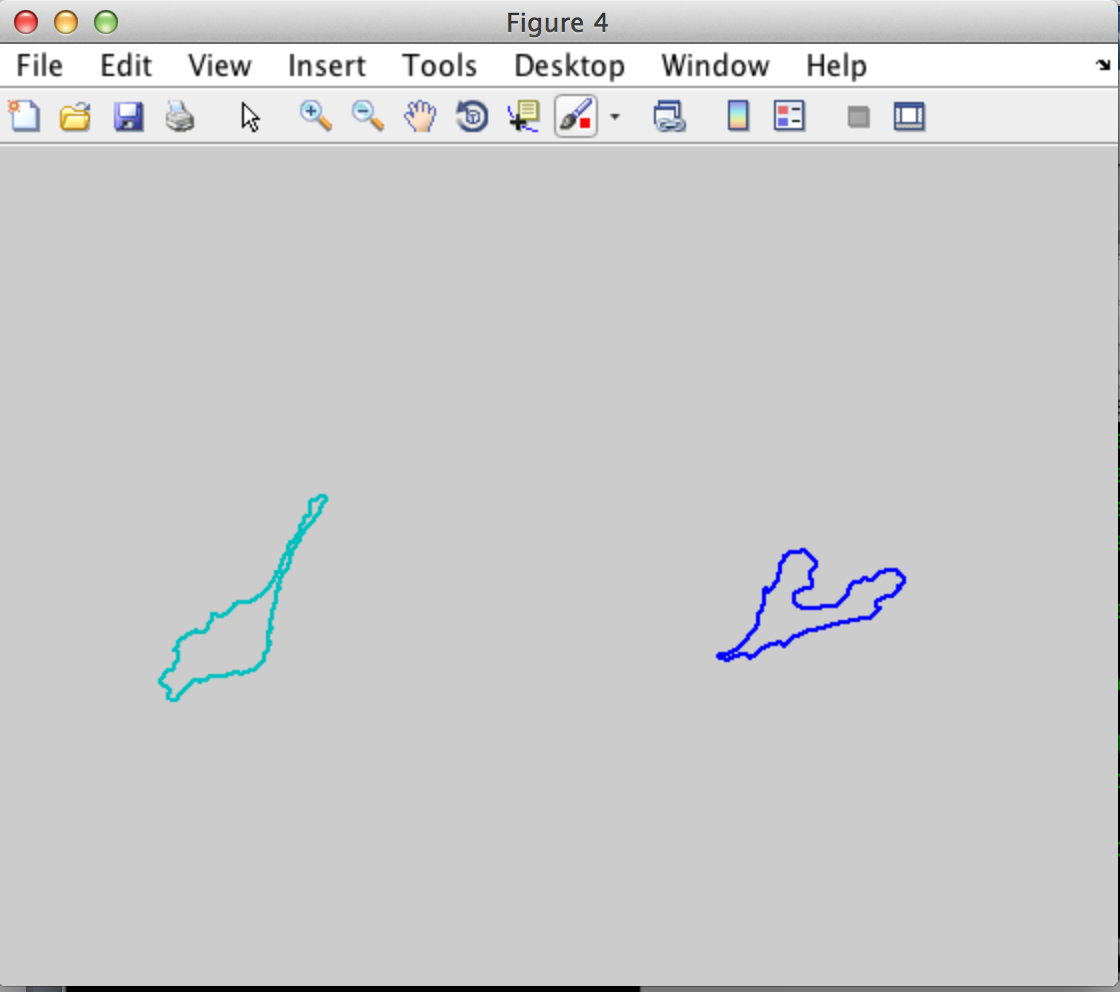
Sometimes, the unordered list might produce too many classes. Than, we first look at the dendogram, by entering zero to the file.

ordered\_list(0,CellShapeData,APe\_output\_foldername)



Now, if you enter a number, you get additional output: namely the following diagrams.

ordered\_list(2,CellShapeData,APe\_output\_foldername)



# 4 Shape Slicer

To run the shape slicer, you need to go to folder “4-Shape\_Averaging”, open the file “Run\_SpaceSlicer”, and click on the “Run” button in Matlab. You should then see the following graphical interface. You click on the button “File” to select the Matlab file “CellShapeData.mat” from your experiment Folder. Then you click on the “Generate Figure” to run the program.



Figure Shape Slicer GUI: Select CellShapeData.mat and configure the slices for the x and y axis.

When the program runs successfully, you should receive four different figures as output. One of those figures should look somewhat similar to the figure below. If you click on the figure under File 🡪 Save As… you can save the figure in vector format and modify with any graphic program.



Figure Sliced Shape Space: blue figures indicate average space in slices along the x-axis, green figures indicate average shape in slices along the y-axis.

However, the program also produces the single parts of the programs individually, e.g. the average figures along the axis as well as only the content, permitting maximum flexibility in graph creation.



Figure : Partial output: left average figure along the x-axis, middle average figure along the y-axis, and shape in the sliced grid.

# 5 Out of sample extension

First, For out of sample extension. I need to train new addition to the sample. For this tutorial, I use the same movie but frame 11 & 12.

* RunMeFirst with this line: RunMeFirst(1, 11,12, '/Users/iasmam/Desktop/SamJ\_Code\_and\_Data/DemoData/GLA6\_exp3\_220713\_untreated01\_114\_R3D.dv','/Users/iasmam/Desktop/SamTest/OOSse' )
* Correct data
* MakeStructre

And now I am ready start.

LP\_OoSE\_train(CellShapeData, '/Users/iasmam/Desktop/SamTest/OOSse' )

CellShape Data comes from CellShapeData.mat from the big training set (not from the small set!)

After this, we have generated in OOse, the file: LP\_trained.mat



The structure insiside LP has a Nx5 matrix. N is the number of frames we analised.

Next step, we run LP\_OoSE\_run;

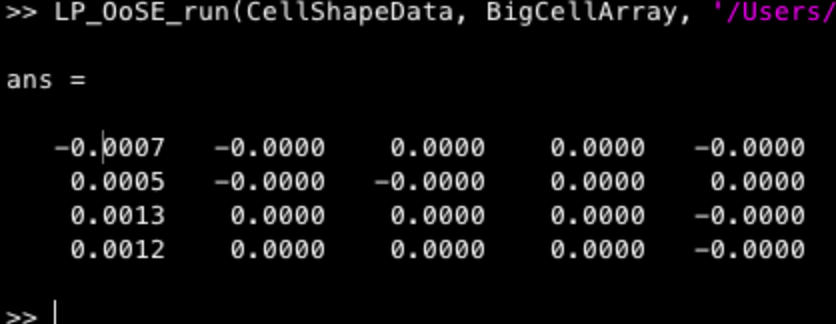
* Where CellShapeData comes from the big set
* BigCellArray comes from the new possible small set.

LP\_OoSE\_run(CellShapeData, BigCellArray, '/Users/iasmam/Desktop/SamTest/OOSse/LP\_trained.mat', '/Users/iasmam/Desktop/SamTest/OOSse/' )

Your console look like this.

And you should have this new files in your folder

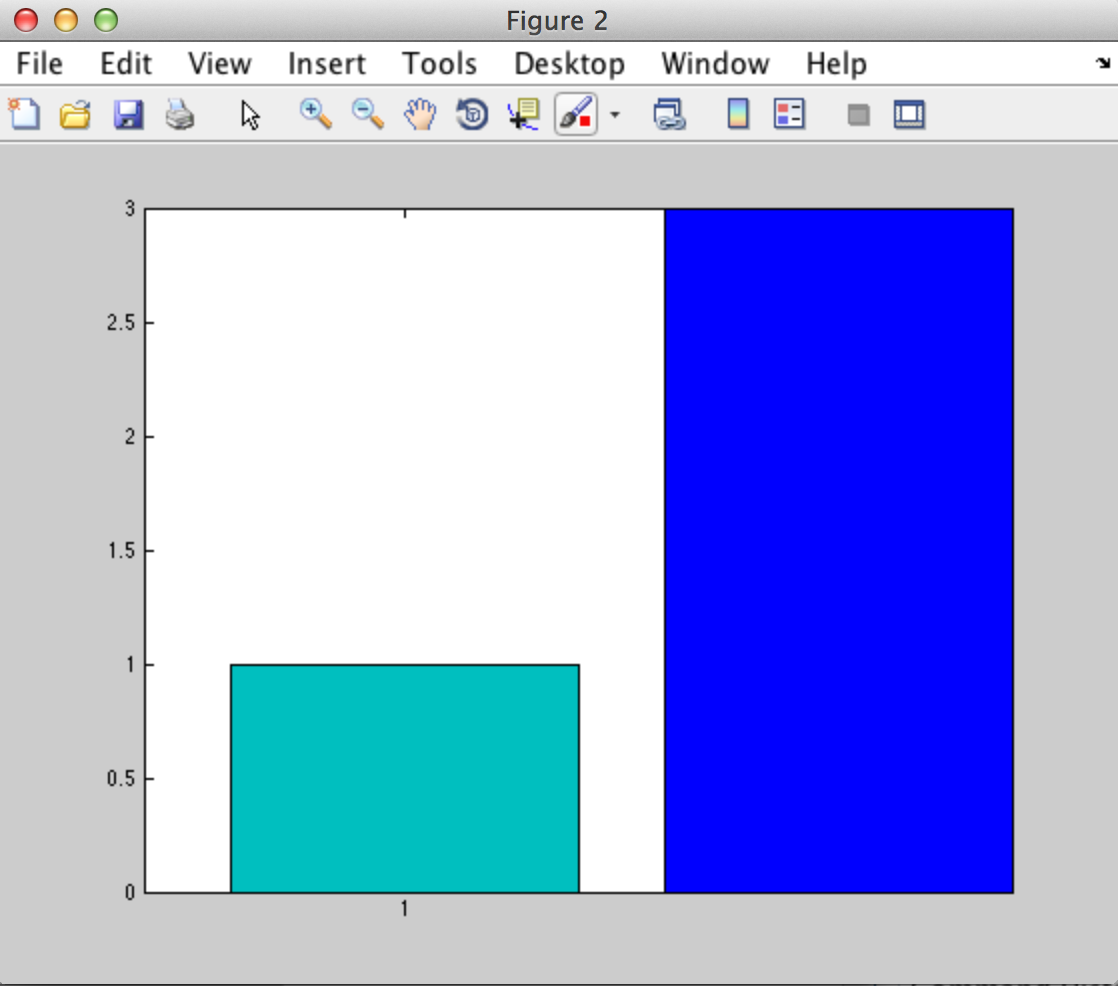
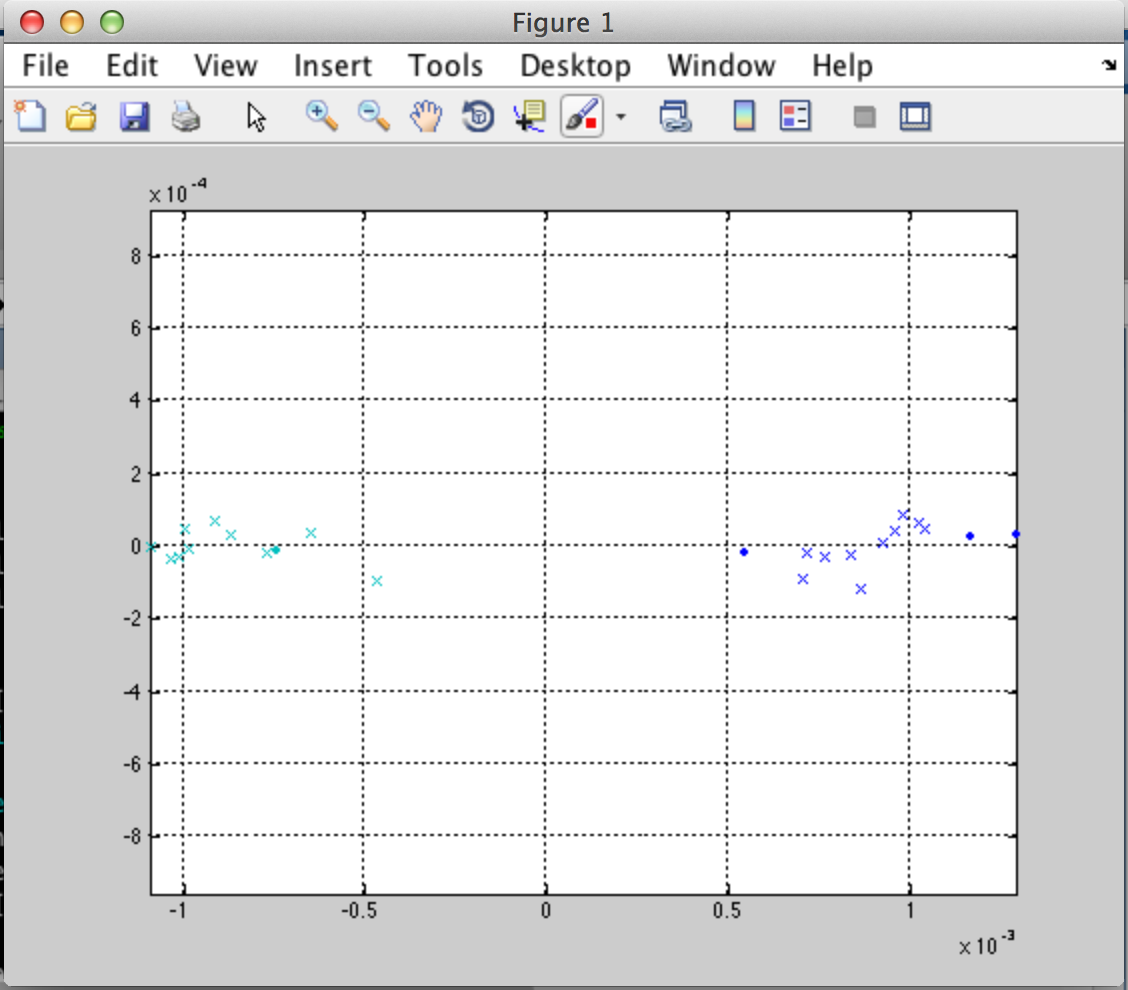
* Dist\_mat.mat
* OoSE\_embedding.mat



Now you are ready to the 3rst step:

To get it to run, enter the following:

OoSE\_bargraphs(2, CellShapeData, '/Users/iasmam/Desktop/SamTest/', '/Users/iasmam/Desktop/SamTest/OOSse')



Now, we see with the crosses the two clusters of shape we have had from the big set. However, the two dots (two green and two blue) are the points that came from the out of sample addition. 3 of the new ones are blues, while one is a greenish one.

# 6 Out of sample extension

Run first MorpProps as for example with

Morph\_Props(BigCellArray, '/Users/iasmam/Desktop/SamTest')

BigCellArray comes from opening “Bigcellarrayandindex.mat” from the big training sample.

This process can be close. After running this process, we have a new file in the folder.

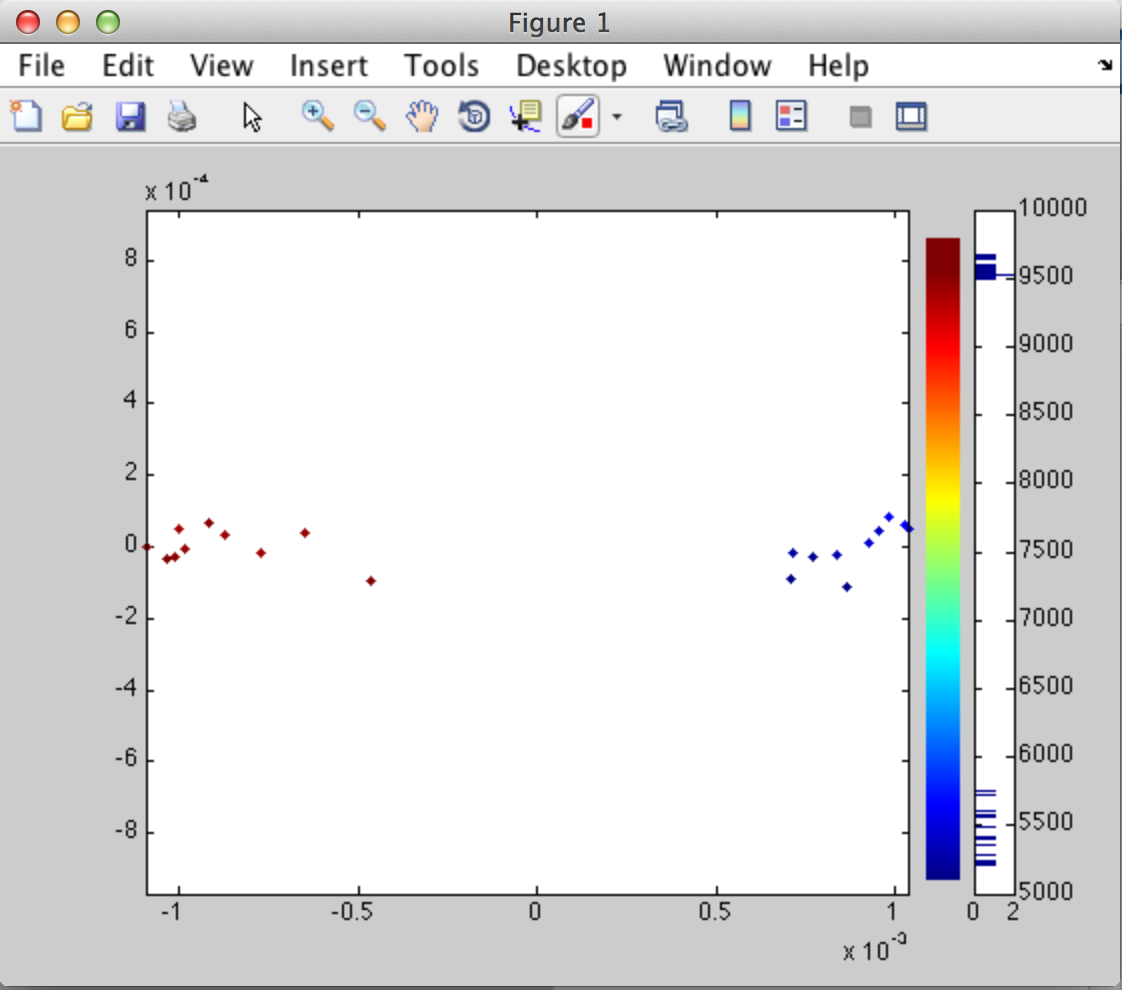


Now we are ready to produce a graph with the following command

Prop\_display(CellShapeData, morphframe, 'Area')

Where

* CellShapeData comes from the big set
* Morphframe from previous operation
* ‘Area’ is one of the possibilities we can select and get



# Step 7 Shape Space Dynamic

To start it, you type for your example this:

ShapeSpaceDynamics(cell\_indices, CellShapeData, '/Users/iasmam/Desktop/SamTest')

* cell\_indices is from the big sample from the file “Bigcellarrayandindex.mat”
* CellShapeData comes from the “CellShapeData.” Also from the big data set.

After this operation we have a new file, DynamicData.mat



in the structure we get we have the following fields:

* Track (is track per cell)
* Speed (is the speed of the cell)
* Average speed is the average speed per tack
* Angles is the angle per cell
* Av\_displacement\_direction : average displacement direction the cell does in its track.

If we have N cells segmented, we should have N structures in this array.

So the first plot we can generate is with this line

Local\_direction\_rose\_plots(DynamicData)

We can influence in how many slices we want to cut the data. “Nbins=[10 6];” in the source code is the important line.

We get than this graph: This shows the local directions in the grids.



The other figure we can create is by typing this:

Exemplar\_direction\_rose\_plots(DynamicData, CellShapeData, 2, '/Users/iasmam/Desktop/SamTest')

* Dynamic Data comes from Folder 7 first step.
* CellShapeData (big sample)
* 2 is the number of cluster, we have determined this in step 3, by looking at the dendogram.
* Foldername, to get the AffinityDatasample files, in detail, they are
  + APclusterOutput.mat
  + Wish\_list.mat
  + Linkagemat.mat

The graph we get is the on in the bottom.

