+/- Message

```
## Loading required package: knitr
```

+/- R Code

+/- R Code

+/- R Code

Quantitative Big Imaging

author: Kevin Mader date: 12 March 2015 width: 1440 height: 900 css: ../common/template.css transition: rotate

ETHZ: 227-0966-00L

Advanced Segmentation and Labeling

Course Outline

+/- R Code

- 19th February - Introduction and Workflows
- 26th February - Image Enhancement (A. Kaestner)
- 5th March - Basic Segmentation, Discrete Binary Structures
- 12th March - Advanced Segmentation
- 19th March - Applying Graphical Models and Machine Learning (A. Lucchi)
- 26th March - Analyzing Single Objects
- 2nd April - Analyzing Complex Objects
- 16th April - Spatial Distribution
- 23rd April - Statistics and Reproducibility
- 30th April - Dynamic Experiments (K. Mader and A. Patera)
- 7th May - Scaling Up / Big Data
- 21th May - Guest Lecture, Applications in Material Science
- 28th May - Project Presentations

Literature / Useful References

- Jean Claude, Morphometry with R
  - Online (http://link.springer.com/book/10.1007%2F978-0-387-77789-4) through ETHZ
  - Buy it (http://www.amazon.com/Morphometrics-R-Use-Julien-Claude/dp/038777789X)
- John C. Russ, “The Image Processing Handbook”,(Boca Raton, CRC Press)
  - Available online (http://dx.doi.org/10.1201/9780203881095) within domain ethz.ch (or proxy.ethz.ch / public VPN)

Advanced Segmentation

- Markov Random Fields for Image Processing Lecture (https://www.youtube.com/watch?v=vRN_j2j-CC4)
- Markov Random Fields Chapter (http://www.cise.ufl.edu/%7Eanand/pdf/bookchap.pdf)
- Fuzzy set theory (http://www.academia.edu/4978200/Applications_of_Fuzzy_Set_Theory_and_Fuzzy_Logic_in_Image_Processing)
- Superpixels (http://ivrg.epfl.ch/research/superpixels)

Contouring

- Active Contours / Snakes (http://link.springer.com/article/10.1007%2FBF00133570#page-1)

Lesson Outline

- Motivation
  - Many Samples
  - Difficult Samples
  - Training / Learning
- Thresholding
  - Automated Methods
  - Hysteresis Method
- Feature Vectors
  - K-Means Clustering
  - Fuzzy Models
  - Probabalistic Models
- Working with Segmented Images
  - Contouring
  - Component Labeling

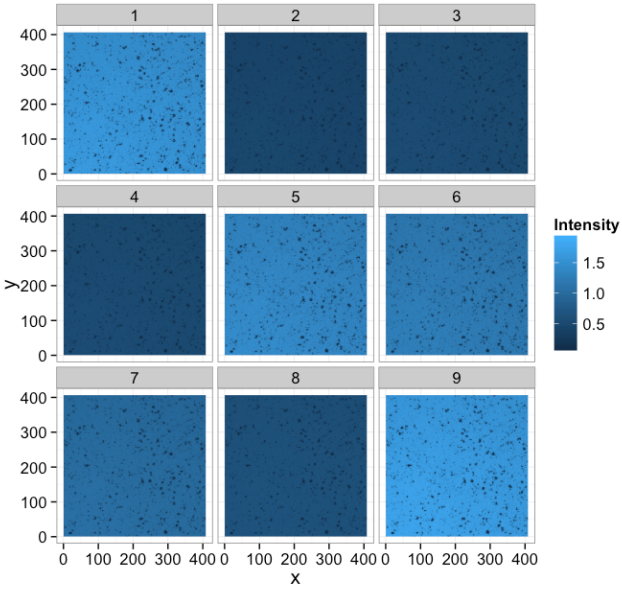
What we covered last time

- Understanding models and interpreting histograms
- Choosing a threshold
  - Examining more complicated, multivariate data sets
- Improving segementation with morphological operations
  - Filling holes
- Partial Volume Effect Caution

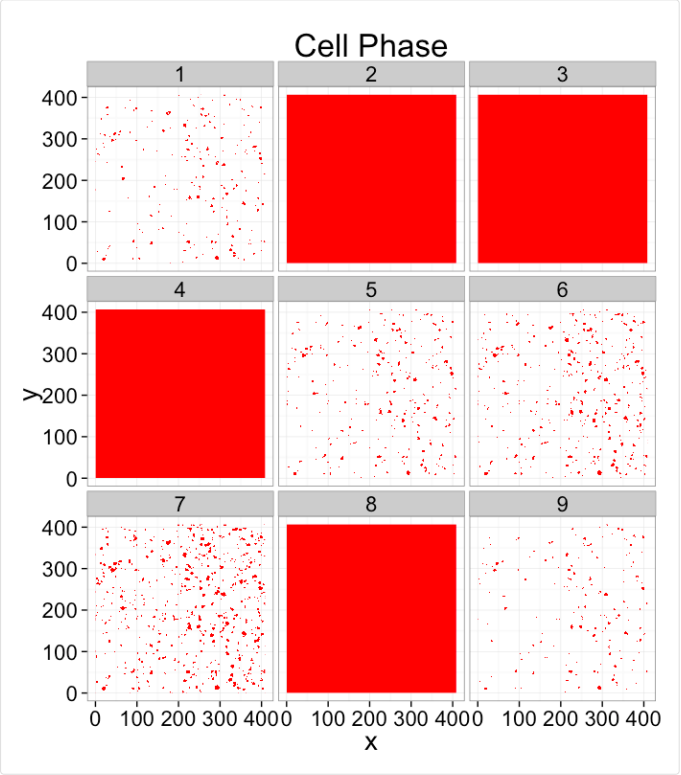
Where segmentation fails: Inconsistent Illumination

With inconsistent or every changing illumination it may not be possible to apply the same threshold to every image.

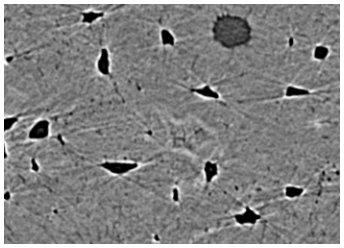
+/- R Code



+/- R Code



Where segmentation fails: Canaliculi



Here is a bone slice

1. Find the larger cellular structures (osteocyte lacunae)
2. Find the small channels which connect them together

The first task

is easy using a threshold and size criteria (we know how big the cells should be)

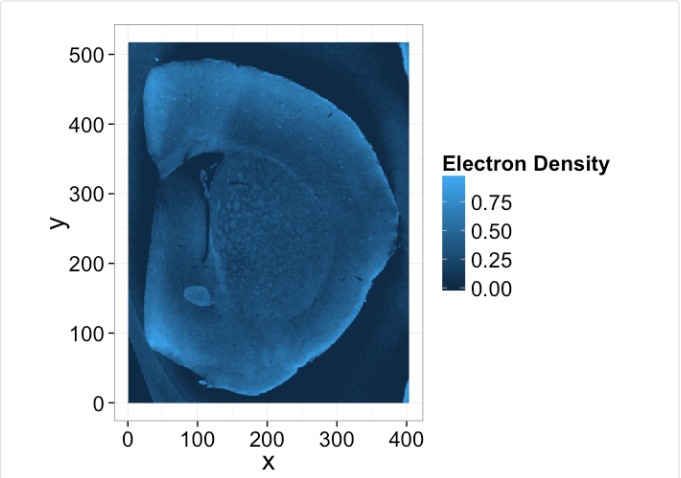
The second

is much more difficult because the small channels having radii on the same order of the pixel size are obscured by partial volume effects and noise.

Where segmentation fails: Brain Cortex

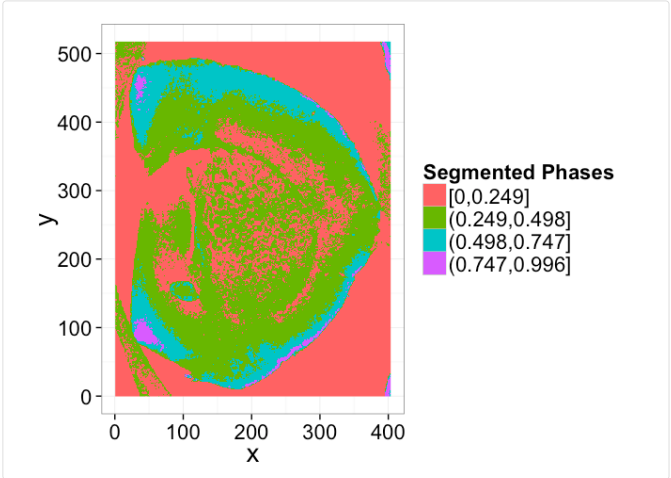
- The cortex is barely visible to the human eye
- Tiny structures hint at where cortex is located

+/- R Code

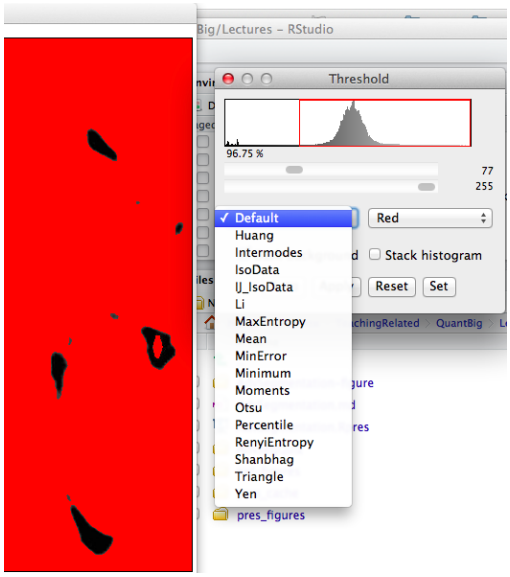


- A simple threshold is insufficient to finding the cortical structures
- Other filtering techniques are unlikely to magicially fix this problem

+/- R Code



Automated Threshold Selection



Given that applying a threshold is such a common and signficant step, there have been many tools developed to automatically (unsupervised) perform it. A particularly important step in setups where images are rarely consistent such as outdoor imaging which has varying lighting (sun, clouds). The methods are based on several basic principles.

Automated Methods

Histogram-based methods

Just like we visually inspect a histogram an algorithm can examine the histogram and find local minimums between two peaks, maximum / minimum entropy and other factors

- Otsu, Isodata, Intermodes, etc

Image based Methods

These look at the statistics of the thresheld image themselves (like entropy) to estimate the threshold

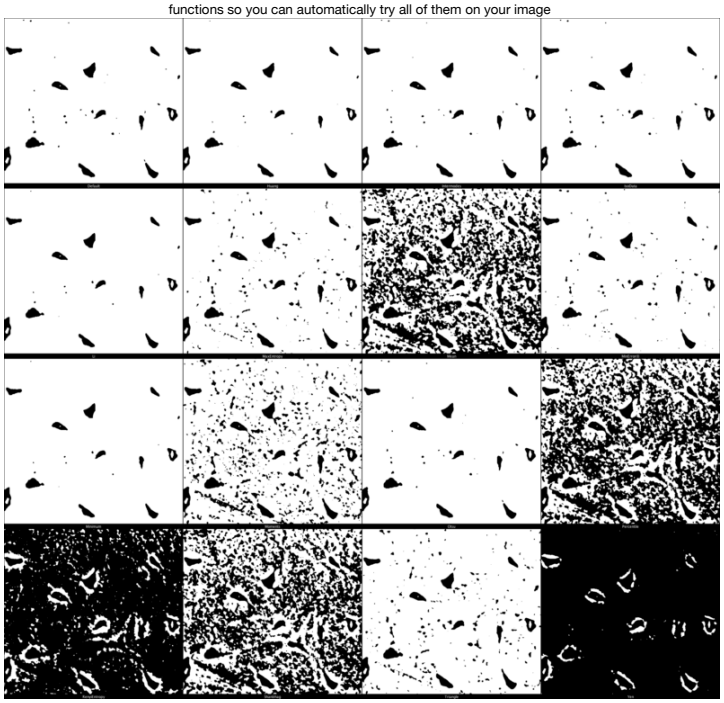
Results-based Methods

These search for a threshold which delivers the desired results in the final objects. For example if you know you have an image of cells and each cell is between 200-10000 pixels the algorithm runs thresholds until the objects are of the desired size

- More specific requirements need to be implemented manually

Fiji -> Adjust -> Auto Threshold

There are many methods and they can be complicated to implement yourself. FIJI offers many of them as built in functions so you can automatically try all of them on your image



Pitfalls

While an incredibly useful tool, there are many potential pitfalls to these automated techniques.

Histogram-based

These methods are very sensitive to the distribution of pixels in your image and may work really well on images with equal amounts of each phase but work horribly on images which have very high amounts of one phase compared to the others

Image-based

These methods are sensitive to noise and a large noise content in the image can change statistics like entropy significantly.

Results-based

These methods are inherently biased by the expectations you have. If you want to find objects between 200 and 1000 pixels you will, they just might not be anything meaningful.

Realistic Approaches for Dealing with these Shortcomings

Imaging science rarely represents the ideal world and will never be 100% perfect. At some point we need to write our master's thesis, defend, or publish a paper. These are approaches for more qualitative assessment we will later cover how to do this a bit more robustly with quantitative approaches

Model-based

One approach is to try and simulate everything (including noise) as well as possible and to apply these techniques to many realizations of the same image and qualitatively keep track of how many of the results accurately identify your phase or not. Hint: >95% seems to convince most biologists

Sample-based

Apply the methods to each sample and keep track of which threshold was used for each one. Go back and apply each threshold to each sample in the image and keep track of how many of them are correct enough to be used for further study.

Worst-case Scenario

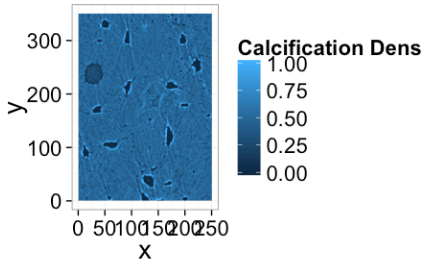
Come up with the worst-case scenario (noise, misalignment, etc) and assess how inacceptable the results are. Then try to estimate the quartiles range (75% - 25% of images).

Hysteresis Thresholding

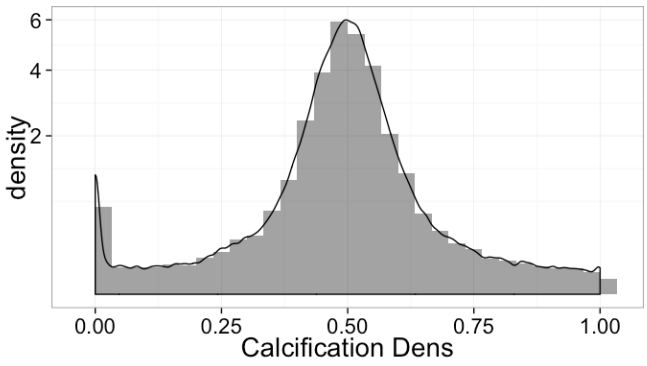
For some images a single threshold does not work

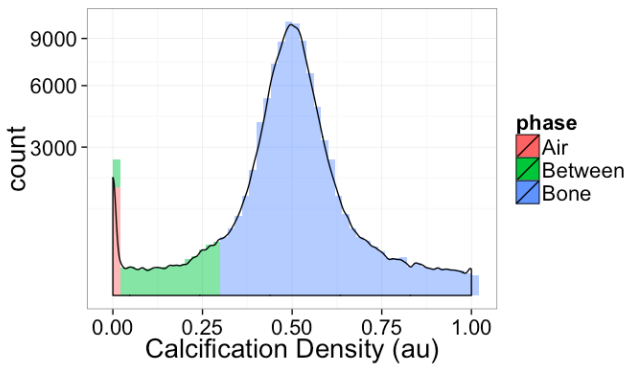
- large structures are very clearly defined
- smaller structures are difficult to differentiated (see partial volume effect (http://bit.ly/1mW7kdP))

ImageJ Source (http://imagejdocu.tudor.lu/doku.php?id=plugin:segmentation:hysteresis_thresholding:start)



+/- R Code

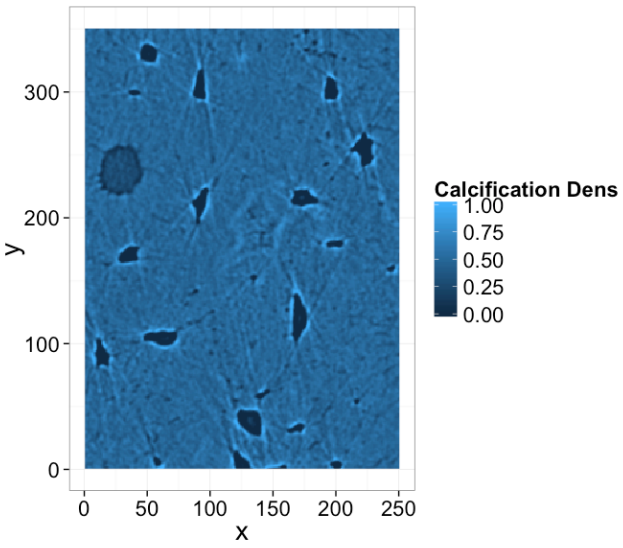


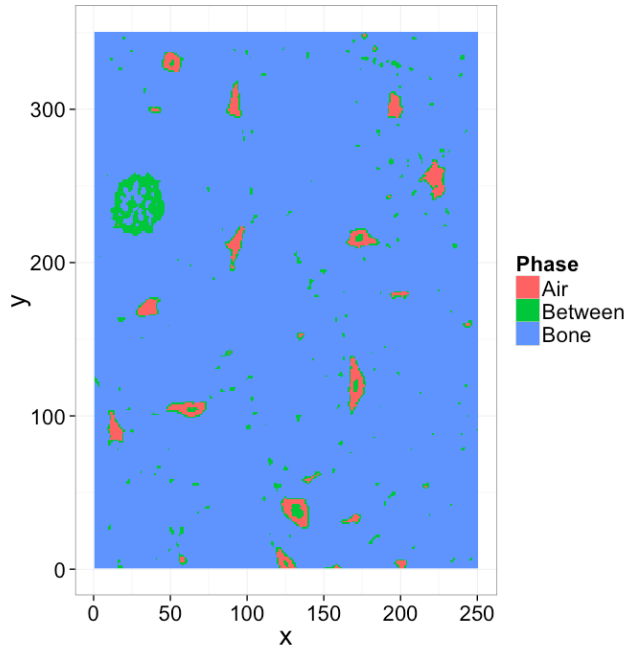


+/- R Code

Hysteresis Thresholding

Comparing the original image with the three phases

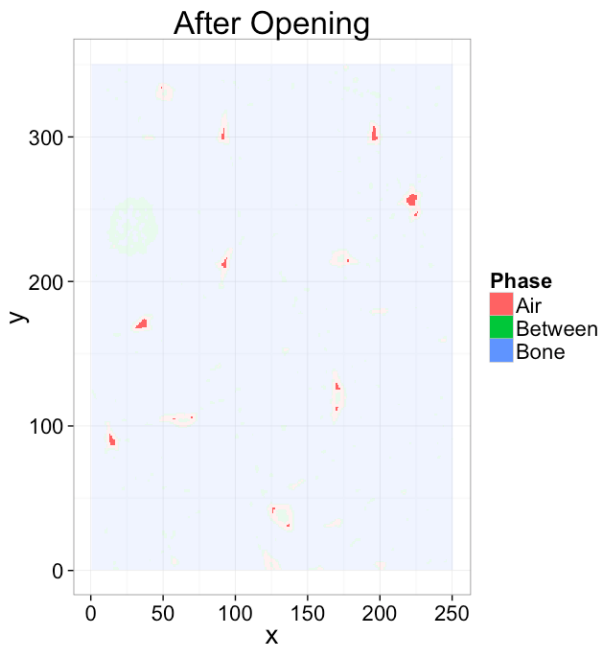




+/- R Code

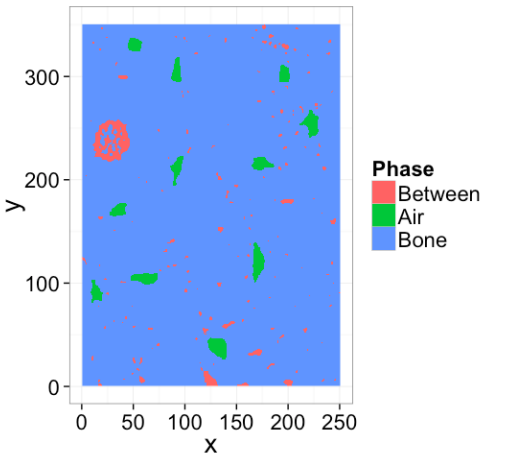
Hysteresis Thresholding: Reducing Pixels

Now we apply two important steps. The first is to remove the objects which are not cells (too small) using an opening operation.



+/- R Code

The second step to keep the *between* pixels which are connected (by looking again at a neighborhood $\mathcal{N}$) to the *air* voxels and ignore the other ones. This goes back to our original supposition that the smaller structures are connected to the larger structures



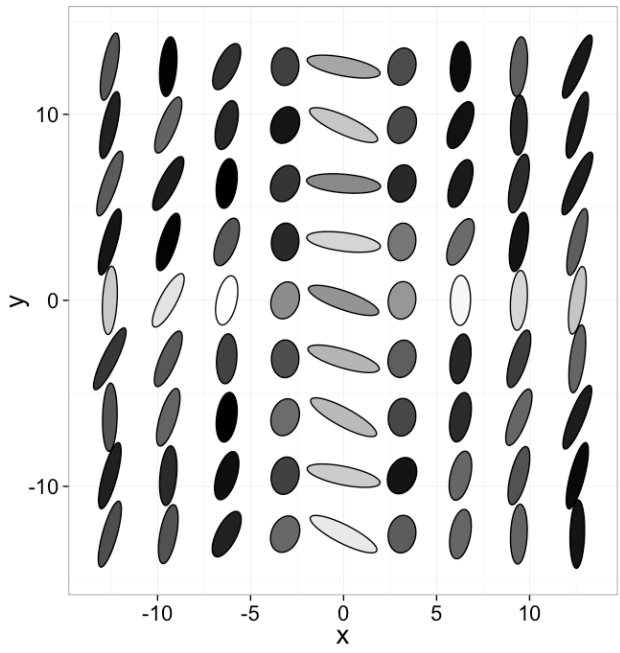
More Complicated Images

As we briefly covered last time, many measurement techniques produce quite rich data.

- Digital cameras produce 3 channels of color for each pixel (rather than just one intensity)
- MRI produces dozens of pieces of information for every voxel which are used when examining different *contrasts* in the system.
- Raman-shift imaging produces an entire spectrum for each pixel

- Coherent diffraction techniques produce 2- (sometimes 3) diffraction patterns for each point.

$$I(x,y) = \hat{f}(x,y)$$



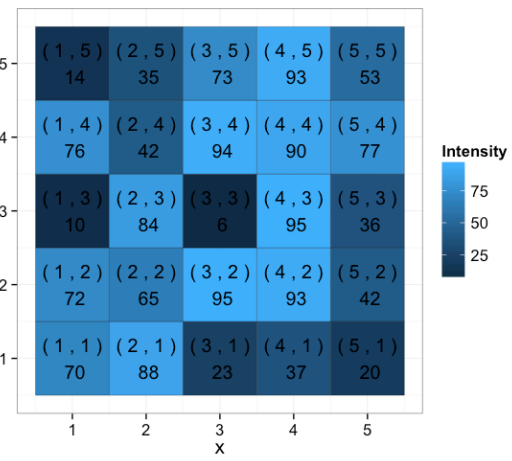
Feature Vectors

A pairing between spatial information (position) and some other kind of information (value).

$$\vec{x} \rightarrow \vec{f}$$

We are used to seeing images in a grid format where the position indicates the row and column in the grid and the intensity (absorption, reflection, tip deflection, etc) is shown as a different color

+/- R Code



The alternative form for this image is as a list of positions and a corresponding value

$$\hat{I} = (\vec{x}, \vec{f})$$

+/- R Code

x	y	Intensity
1	1	70
2	1	88
3	1	23
4	1	37
5	1	20
1	2	72

This representation can be called the **feature vector** and in this case it only has Intensity

Why Feature Vectors

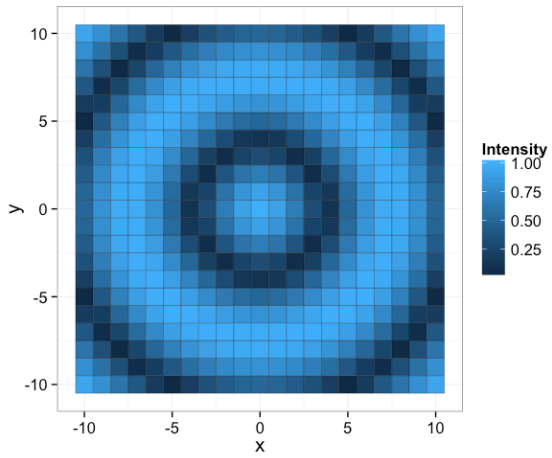
If we use feature vectors to describe our image, we are no longer to worrying about how the images will be displayed, and can focus on the segmentation/thresholding problem from a classification rather than a image-processing stand point.

Example

So we have an image of a cell and we want to identify the membrane (the ring) from the nucleus (the point in the middle).

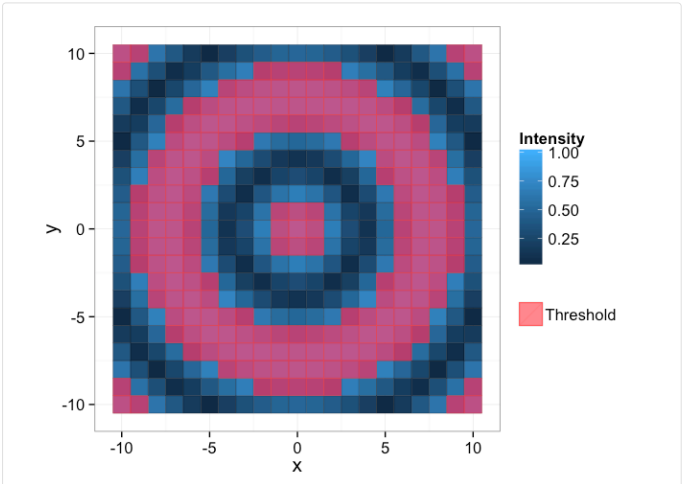
+/- R Code

+/- R Code



A simple threshold doesn't work because we identify the point in the middle as well. We could try to use morphological tricks to get rid of the point in the middle, or we could better tune our segmentation to the ring structure.

+/- R Code

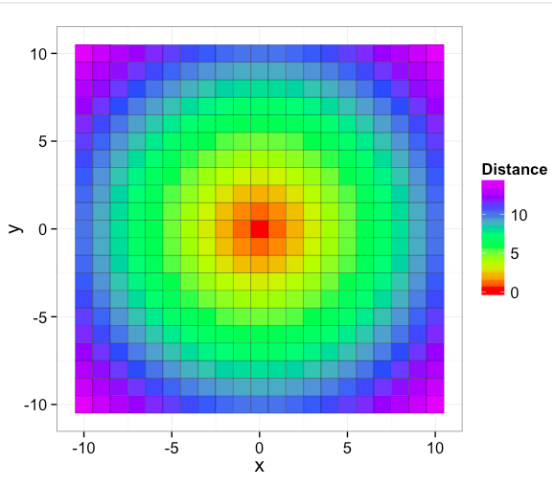


Adding a new feature

In this case we add a very simple feature to the image, the distance from the center of the image (distance).

+/- R Code

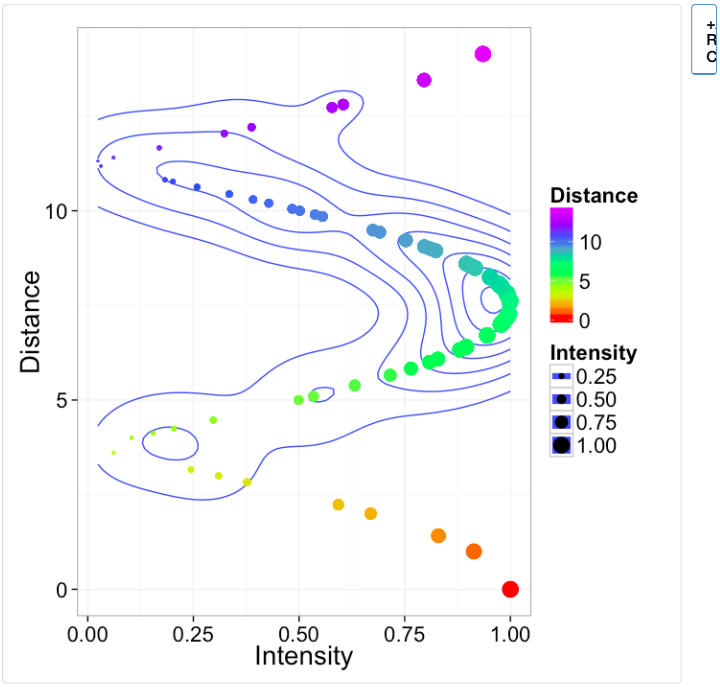
+/- R Code



+/- R Code

x	y	Intensity	Distance
-10	-10	0.9350683	14.14214
-10	-9	0.7957197	13.45362
-10	-8	0.6045178	12.80625
-10	-7	0.3876575	12.20656
-10	-6	0.1692429	11.66190
-10	-5	0.0315481	11.18034

We now have a more complicated image, which we can't as easily visualize, but we can incorporate these two pieces of information together.

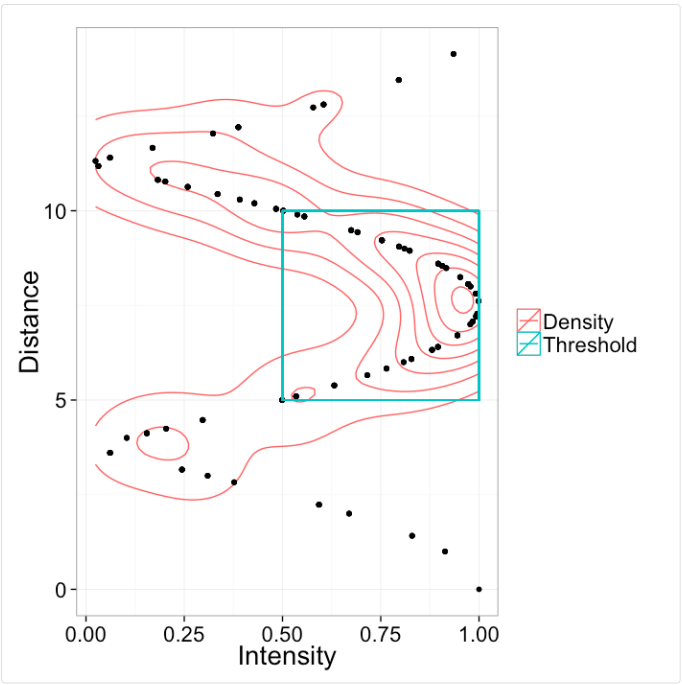


+/- R Code

Applying two criteria

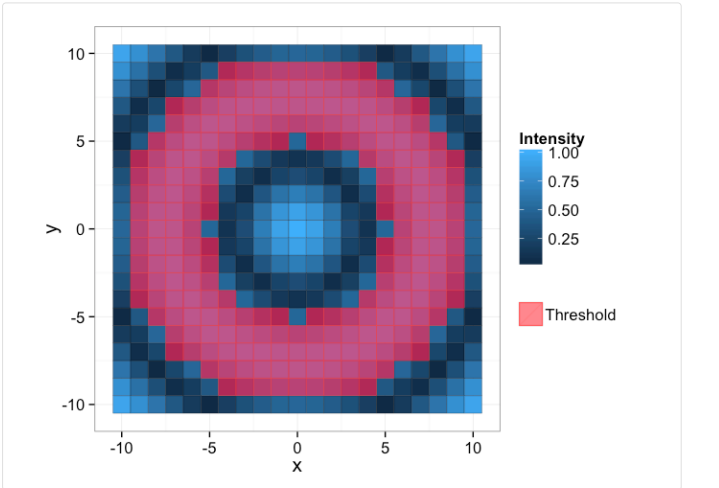
Now instead of trying to find the intensity for the ring, we can combine density and distance to identify it

$$iff(5 < \text{Distance} < 10 \\ \&0.5 < \text{Intensity} > 1.0)$$



+/- R Code

+/- R Code



Common Features

The distance while illustrative is not a commonly used features, more common various filters applied to the image

- Gaussian Filter (information on the values of the surrounding pixels)
- Sobel / Canny Edge Detection (information on edges in the vicinity)
- Entroy (information on variability in vicinity)

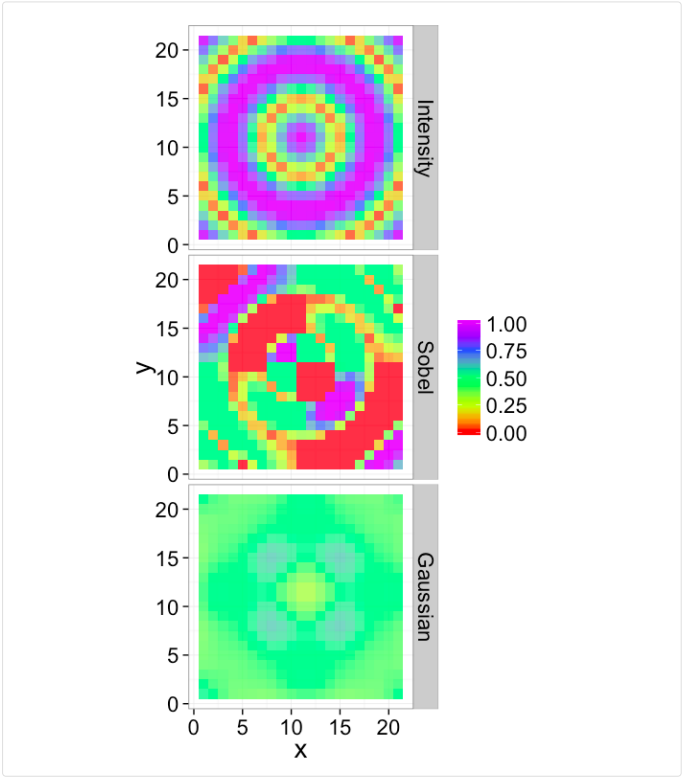
+/- R Code

+/- R Code

x	y	Intensity	Sobel	Gaussian
1	1	0.94	0.32	0.53

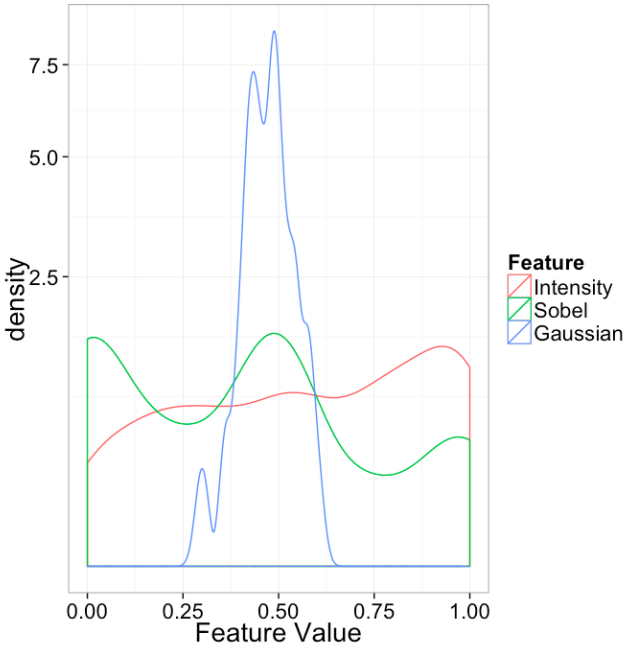
1	10	0.48	0.50	0.45
1	11	0.50	0.50	0.46
1	12	0.48	0.64	0.46
1	13	0.43	0.78	0.45
1	14	0.33	0.94	0.42

+/- R Code

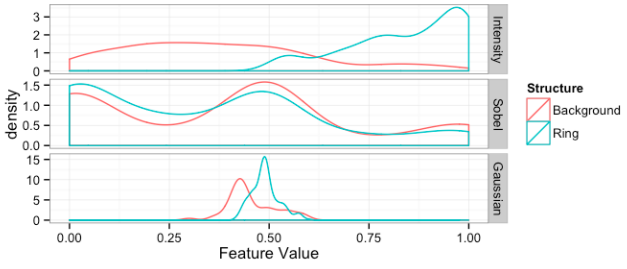


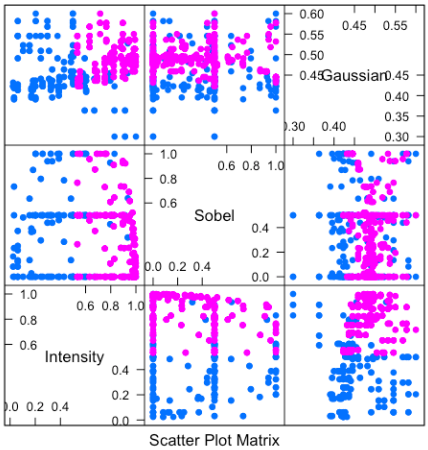
Analyzing the feature vector

The distributions of the features appear very different and can thus likely be used for identifying different parts of the images.



Combine this with our *a priori* information (called supervised analysis)





K-Means Clustering / Classification (Unsupervised)

- Automatic clustering of multidimensional data into groups based on a distance metric
- Fast and scalable to petabytes of data (Google, Facebook, Twitter, etc. use it regularly to classify customers, advertisements, queries)
- **Input** = feature vectors, distance metric, number of groups
- **Output** = a classification for each feature vector to a group

K-Means Example

Input

- Feature vectors ($\vec{v}$)

+/- R Code

	x	y	Intensity	Sobel	Gaussian
1	1	1	0.94	0.32	0.53
2	1	10	0.48	0.50	0.45
3	1	11	0.50	0.50	0.46
4	1	12	0.48	0.64	0.46
5	1	13	0.43	0.78	0.45
6	1	14	0.33	0.94	0.42

- Distance metric

$$D_{ij} = \|\vec{v}_i - \vec{v}_j\|$$

- Group Count ($N = 2$)

↓

Output

- Group 1

+/- R Code

	x	y	Intensity	Sobel	Gaussian
20	1	8	0.33	0.50	0.40
21	1	9	0.43	0.50	0.42
22	10	1	0.48	0.14	0.45
23	10	10	0.83	0.50	0.42
24	10	11	0.91	0.50	0.36

- Group 2

+/- R Code

	x	y	Intensity	Sobel	Gaussian
100	13	4	1.00	0.16	0.49
101	13	5	0.88	0.74	0.49
102	13	6	0.63	0.96	0.52
103	13	7	0.30	0.94	0.55
104	13	8	0.06	0.00	0.55

K-Means Algorithm

We give as an initial parameter the number of groups we want to find and possible a criteria for removing groups that are too similar

1. Randomly create center points (groups) in vector space
2. Assigns group to data point by the “closest” center
3. Recalculate centers from mean point in each group
4. Go back to step 2 until the groups stop changing

What vector space to we have?

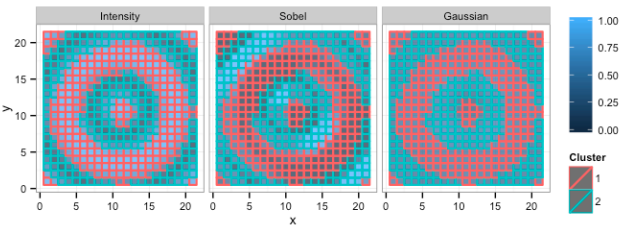
- Sometimes represent physical locations (classify swiss people into cities)
- Can include intensity or color (K-means can be used as a thresholding technique when you give it image intensity as the vector and tell it to find two or more groups)

- Can also include orientation, shape, or in extreme cases full spectra (chemically sensitive imaging)

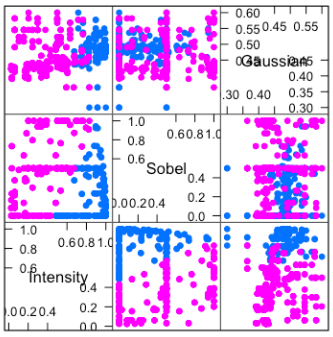
Note: If you look for N groups you will almost always find N groups with K-Means, whether or not they make any sense

K-Means Example

Continuing with our previous image and applying K-means to the Intensity, Sobel and Gaussian channels looking for 2 groups we find



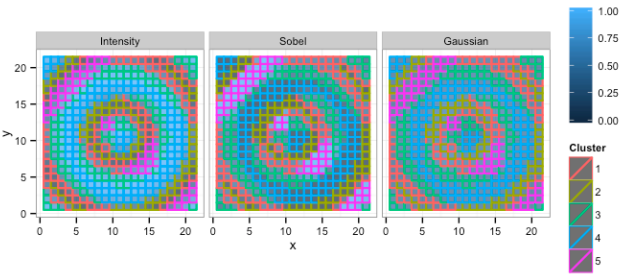
+/- R Code

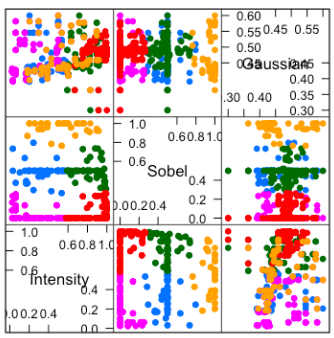


Scatter Plot Matrix

Looking for 5 groups

+/- R Code

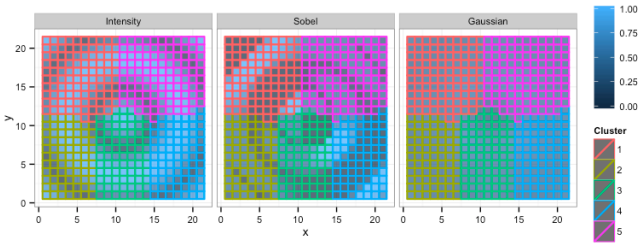




Scatter Plot Matrix

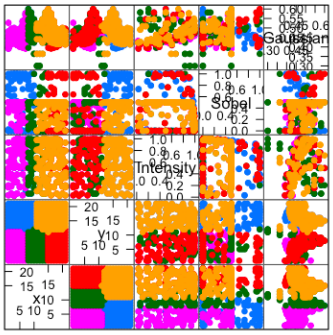
Changing the feature vector

Including the position in the features as well



+/- R Code

+/- R Code



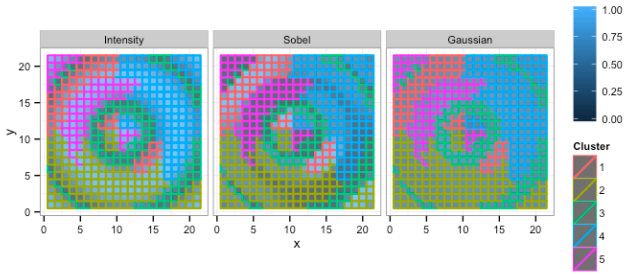
Scatter Plot Matrix

Rescaling components

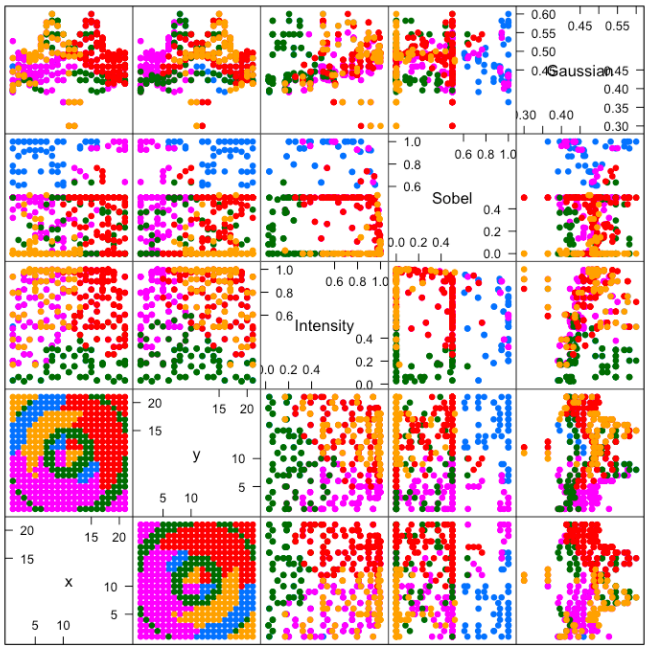
Since the distance is currently calculated by $\|\vec{v}_i - \vec{v}_j\|$ and the values for the position is much larger than the values for the *Intensity*, *Sobel* or *Gaussian* they need to be rescaled so they all fit on the same axis

$$\vec{v} = \left\{ \frac{x}{10}, \frac{y}{10}, \text{Intensity}, \text{Sobel}, \text{Gaussian} \right\}$$

+/- R Code



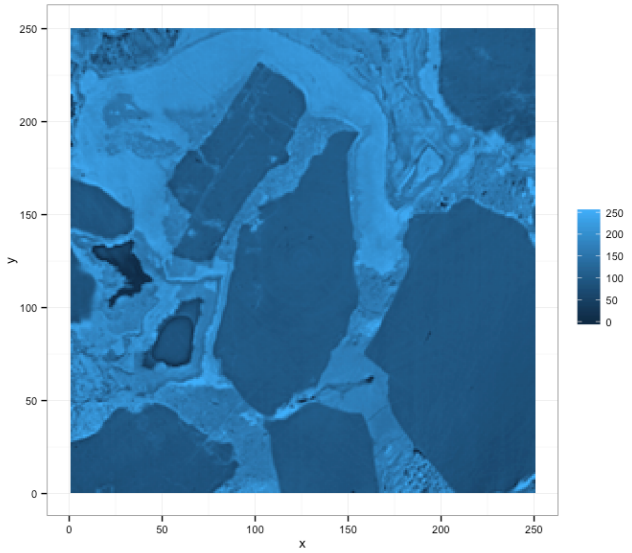
+/- R Code



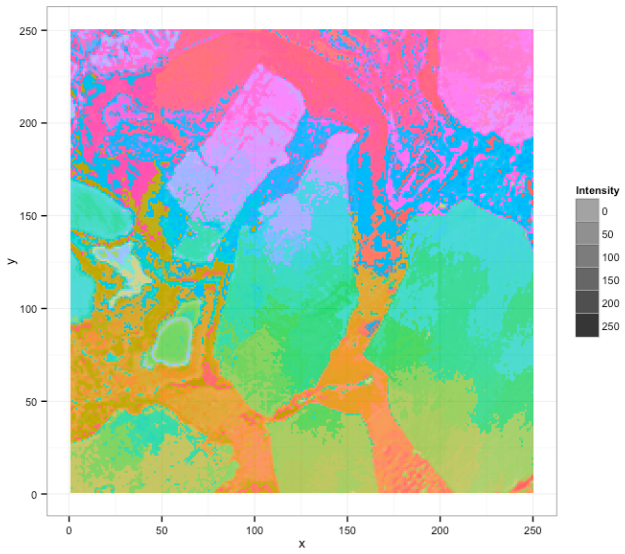
Scatter Plot Matrix

Super-pixels

An approach for simplifying images by performing a clustering and forming super-pixels from groups of similar pixels.



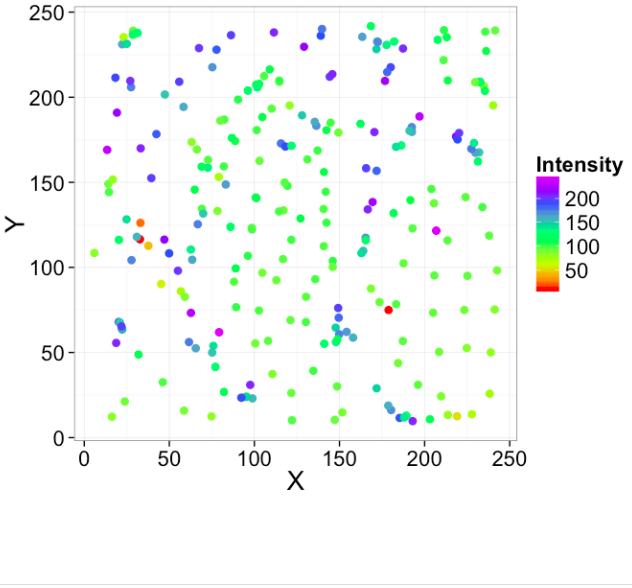
+/- R Code



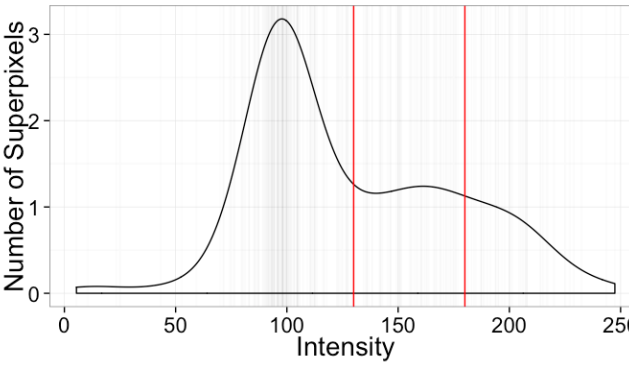
Why use superpixels

Drastically reduced data size, serves as an initial segmentation showing spatially meaningful groups

+/- R Code



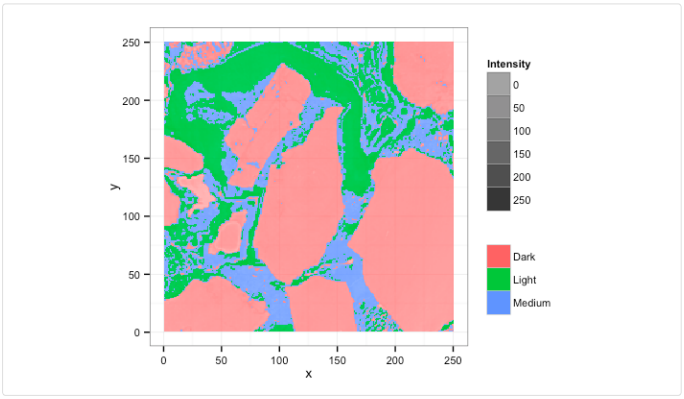
+/- R Code



Segment the superpixels and apply them to the whole image (only a fraction of the data and much smaller datasets)

+/- R Code

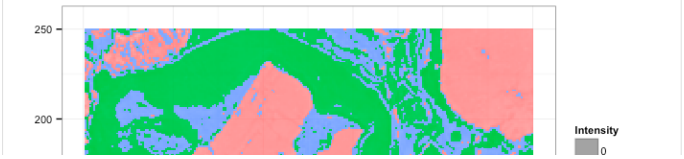
+/- R Code

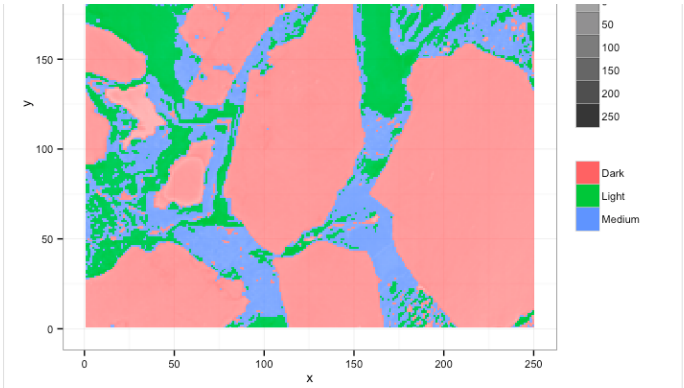


Superpixels vs Standard Segmentation

Superpixels (0.06% of the original)

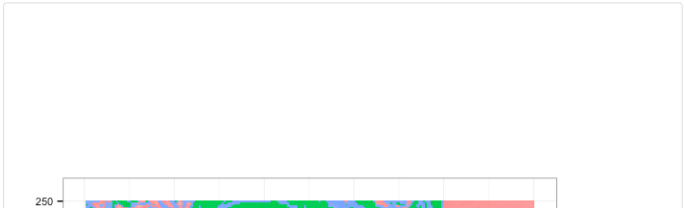
+ R C

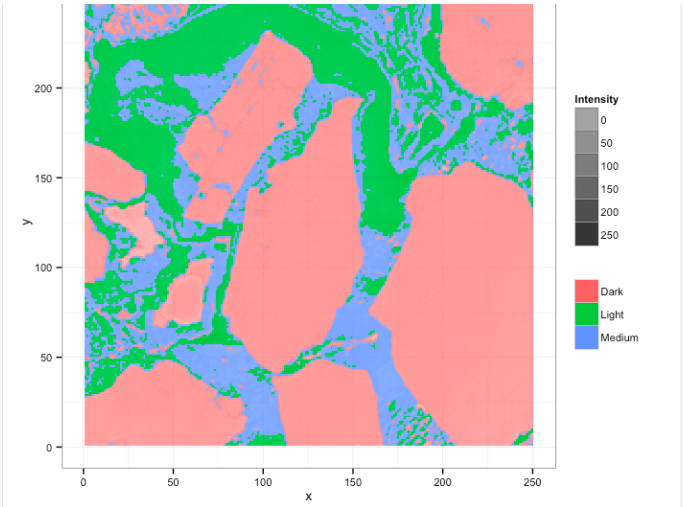




Original

+ R C





Probabilistic Models of Segmentation

A more general approach is to use a probabilistic model to segmentation. We start with our image $I(\vec{x}) \forall \vec{x} \in \mathbb{R}^N$ and we classify it into two phases α and β

$$P(\{\vec{x}, I(\vec{x})\}|\alpha) \propto P(\alpha) + P(I(\vec{x})|\alpha) + P(\sum_{x' \in \mathcal{N}} I(\vec{x'})|\alpha)$$

- $P(\{\vec{x}, f(\vec{x})\}|\alpha)$ the probability a given pixel is in phase α given we know it's position and value (what we are trying to estimate)
- $P(\alpha)$ probability of any pixel in an image being part of the phase (expected volume fraction of that phase)
- $P(I(\vec{x})|\alpha)$ probability adjustment based on knowing the value of I at the given point (standard threshold)
- $P(f(\vec{x'})|\alpha)$ are the collective probability adjustments based on knowing the value of a pixels neighbors (very simple version of Markov Random Field (http://en.wikipedia.org/wiki/Markov_random_field) approaches)

Contouring

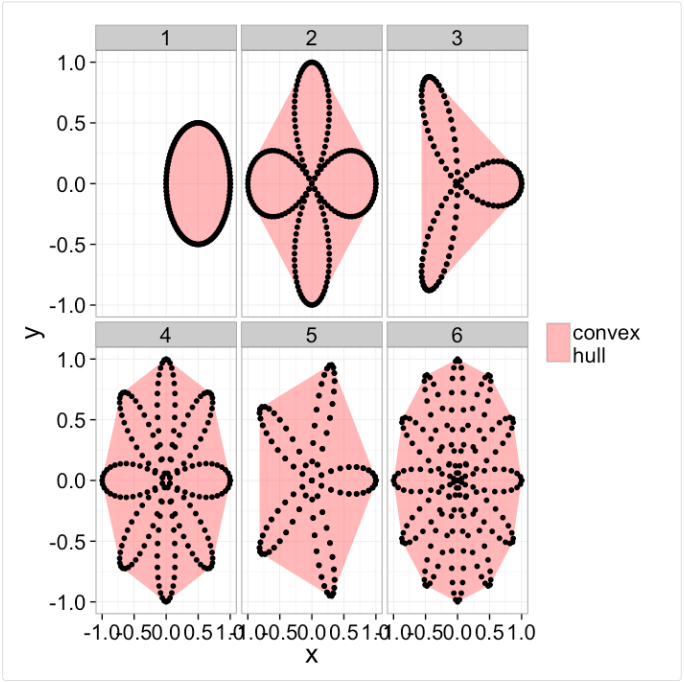
Expanding on the hole filling issues examined before, a general problem in imaging is identifying regions of interest with in an image.

- For samples like brains it is done to identify different regions of the brain which are responsible for different functions.
- In material science it might be done to identify a portion of the sample being heated or understress.
- There are a number of approaches depending on the clarity of the data and the

Convex Hull Approach

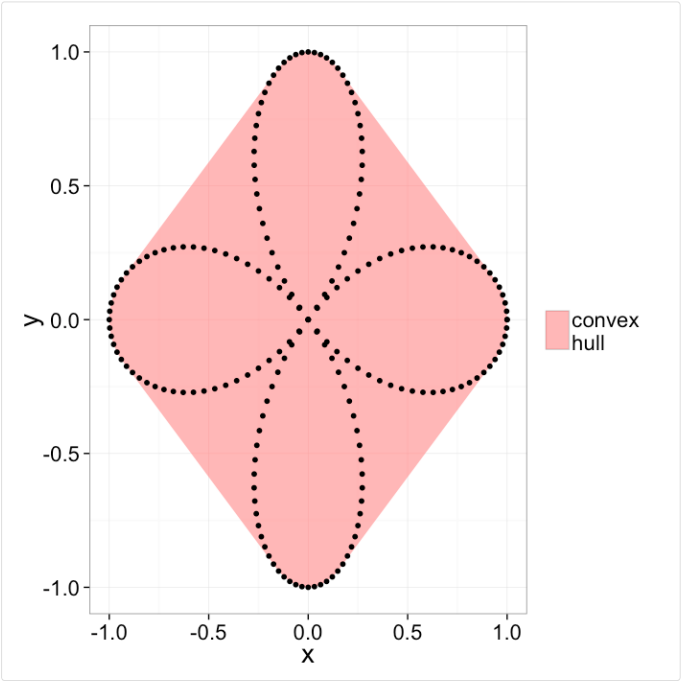
takes all of the points in a given slice or volume and finds the smallest convex 2D area or 3D volume (respectively) which encloses all of those points.

+/- R Code



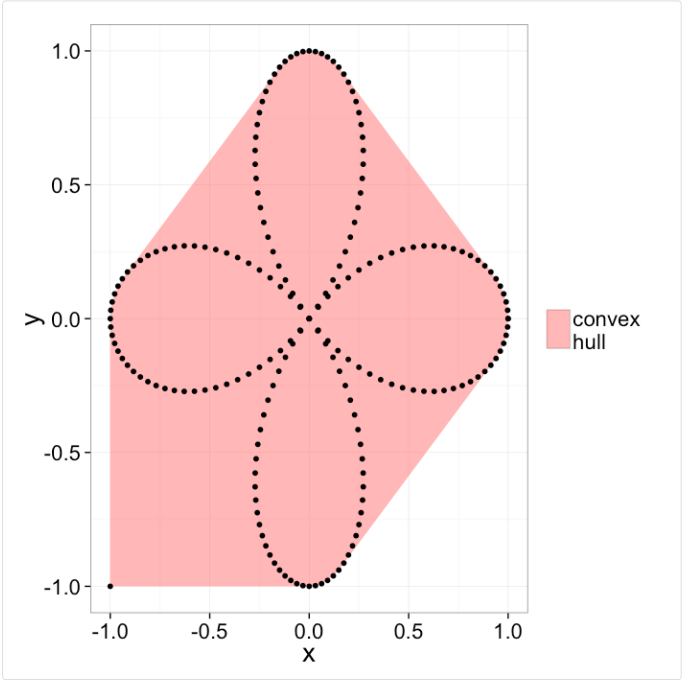
Depending on the type of sample the convex hull can make sense for filling in the gaps and defining the boundaries for a sample.

+/- R Code



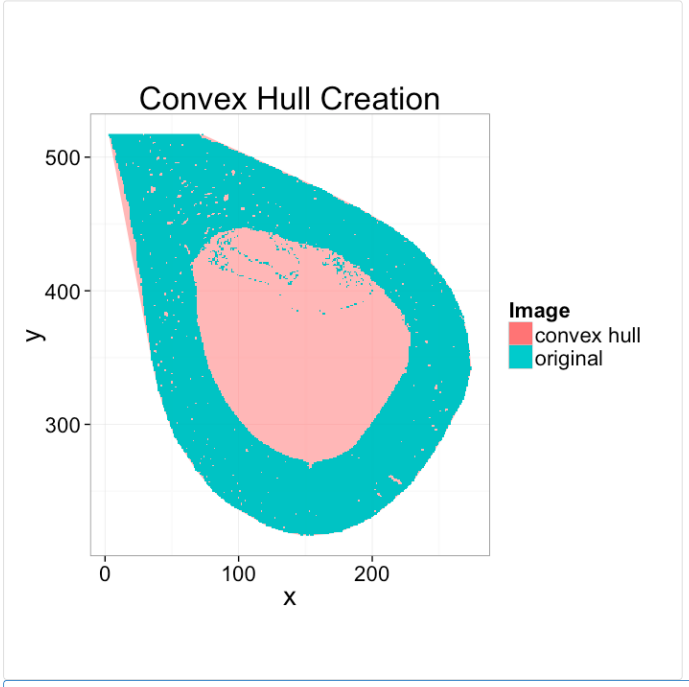
The critical short coming is it is very sensitive to single outlier points.

+/- R Code



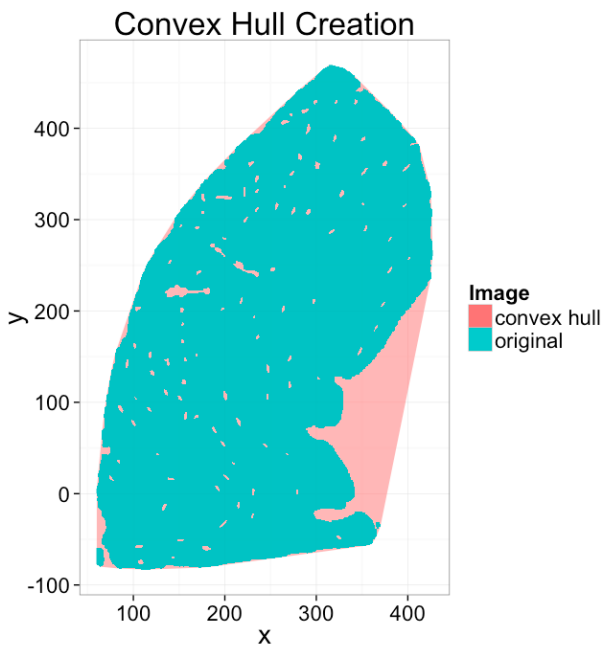
Convex Hull Example

The convex hull very closely matches the area we would define as 'bone' without requiring any parameter adjustment, resolution specific adjustments, or extensive image-processing, for such a sample a convex hull is usually sufficient.



+/- R Code

Here is an example of the convex hull applied to a region of a cortical bone sample. The green shows the bone and the red shows the convex hull. Compared to a visual inspection, the convex hull overestimates the bone area as we probably would not associate the region where the bone curves to the right with 'bone area'

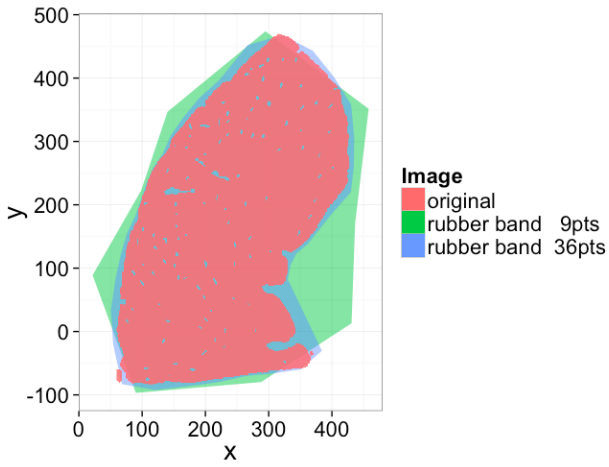


Rubber Band

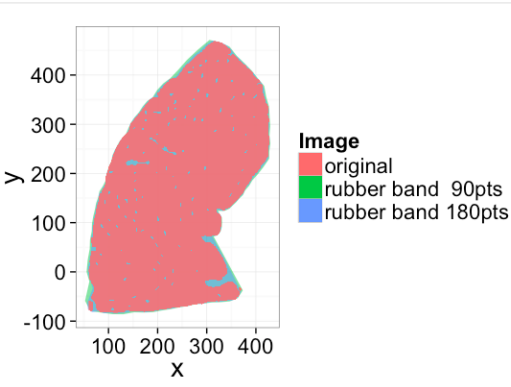
+/- R Code

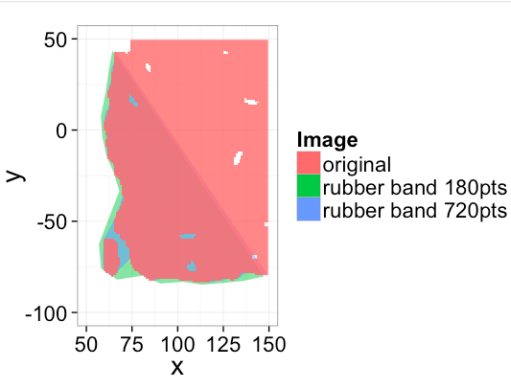
Useful for a variety of samples (needn't be radially symmetric) and offers more flexibility in step size, smoothing function etc than convex hull.

1. Calculates the center of mass.
2. Transforms sample into Polar Coordinates
3. Calculates a piecewise linear fit $r = f(\theta)$



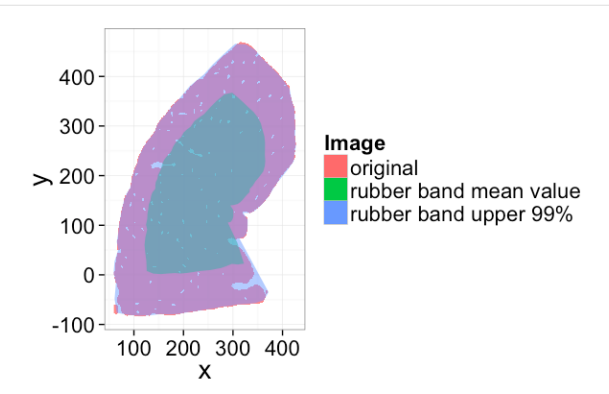
+/- R Code





Rubber Band: More flexible constraints

If we use quartiles or the average instead of the maximum value we can make the method less sensitive to outlier pixels



Contouring: Manual - Guided Methods

Many forms of guided methods exist, the most popular is known simply as the *Magnetic Lasso* in Adobe Photoshop (video (http://people.ee.ethz.ch/%7Emaderk/videos/MagneticLasso.swf)).

The basic principal behind many of these methods is to optimize a set of user given points based on local edge-like information in the image. In the brain cortex example, this is the small gradients in the gray values which our eyes naturally seperate out as an edge but which have many gaps and discontinuities.

Active Contours / Snakes (http://link.springer.com/article/10.1007%2FBF00133570#page-1)

Beyond

- A multitude of other techniques exist for classifying groups and courses in Data Science and Artificial Intelligence go into much greater details.
- These techniques are generally underused because they are complicated to explain and robustly test and can arouse suspicion from reviewers.

    - Because of their added complexity it is easier to manipulate these methods to get desired results from almost any dataset
    - But if the approach is based on a physical model of the images and the underyling system it is acceptable
- Additionally they usually require some degree of implementation (coding).

Fuzzy Classification

Fuzzy classification based on Fuzzy logic (http://en.wikipedia.org/wiki/Fuzzy_logic) and Fuzzy set theory (http://www.academia.edu/4978200/Applications_of_Fuzzy_Set_Theory_and_Fuzzy_Logic_in_Image_Processing) and is a general catagory for multi-value logic instead of simply **true** and **false** and can be used to build **IF** and **THEN** statements from our probabilistic models.

Instead of

$$P(\{\vec{x}, I(\vec{x})\}|\alpha) \propto P(\alpha) + P(I(\vec{x})|\alpha) +$$

$$P(\sum_{x' \in \mathcal{N}} I(\overrightarrow{x'})|\alpha)$$

Clear simple rules

which encompass aspects of filtering, thresholding, and morphological operations

- **IF** the intensity if dark (< 100)
    - **AND** a majority of the neighborhood ($\mathcal{N}$) values are dark (< 100)
- **THEN** it is a cell