Automated Microscope Image Interpretation

Robert F Murphy

Ray & Stephanie Lane Professor of Computational Biology
Departments of Biological Sciences, Biomedical Engineering and Machine Learning and

RAY AND STEPHANIE LANE
Center for Computational Biology

Carnegie Mellon

Automated Interpretation

- Traditional analysis of fluorescence microscope images has occurred by visual inspection
- My group's goal over the past fourteen years has to been automate the interpretation, to yield better
  - Objectivity
  - Sensitivity
  - Reproducibility

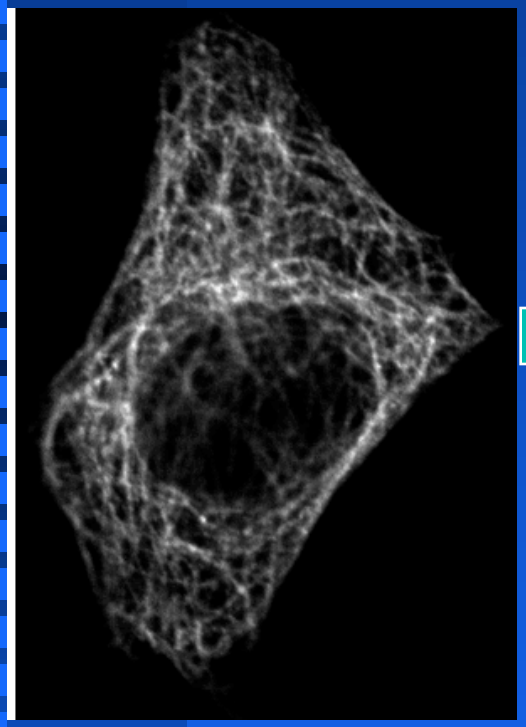
Focus on subcellular location analysis

- We will focus on analysis of subcellular location, but most of the methods we will discuss are equally applicable to other levels or organization/resolution

But first a word about acquisition

- Carefully consider
  - What resolution/dimension images to you need for your task?
  - How many images/cells do you need per condition?
- Keep conditions (especially microscopy settings) constant!

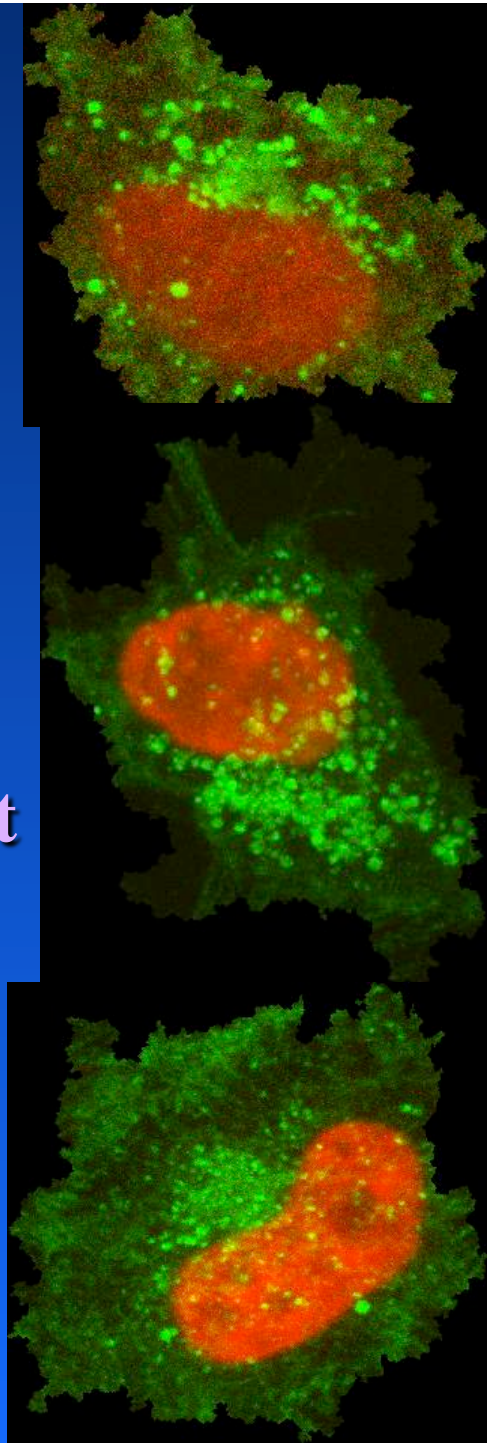
Initial Goal: Supervised Learning



Assign proteins to *major* subcellular structures using fluorescent microscopy

The Challenge

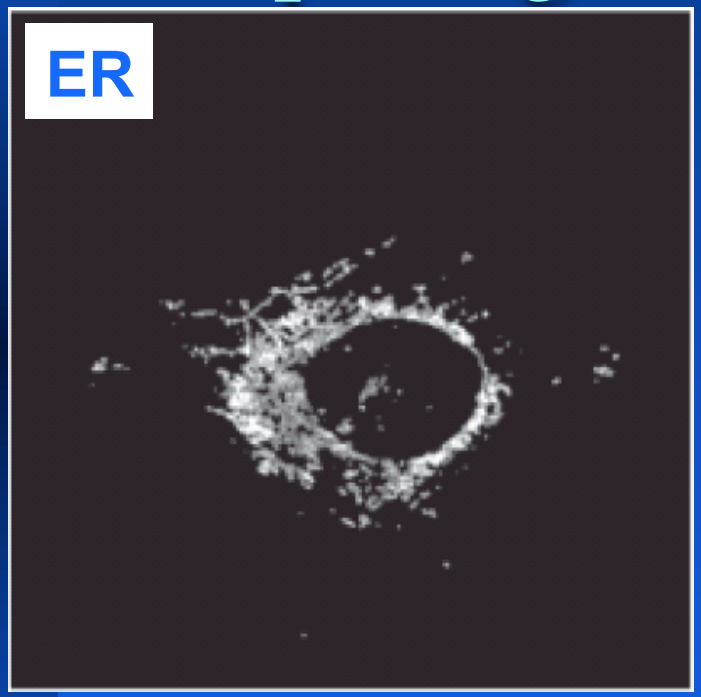
- Pixel-by-pixel or region-by-region matching will not work for cell patterns because different cells have different **shapes, sizes, orientations**
- Organelles/structures within cells are **not found in fixed locations**
- ***Instead, describe each image numerically and compare the descriptors***

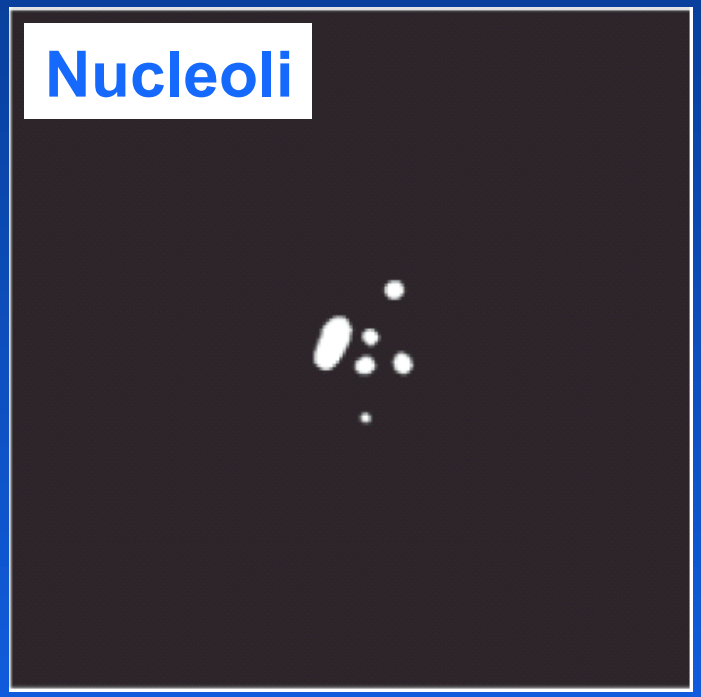


Feature-based, Supervised learning approach

1. Create sets of images showing the location of many different proteins (each set defines one **class** of pattern)
2. Reduce each image to a set of numerical values ("**features**") that are insensitive to position and rotation of the cell
3. Use **machine learning methods** to "learn" how to distinguish each class using the features

Example of classification using Morphological Features

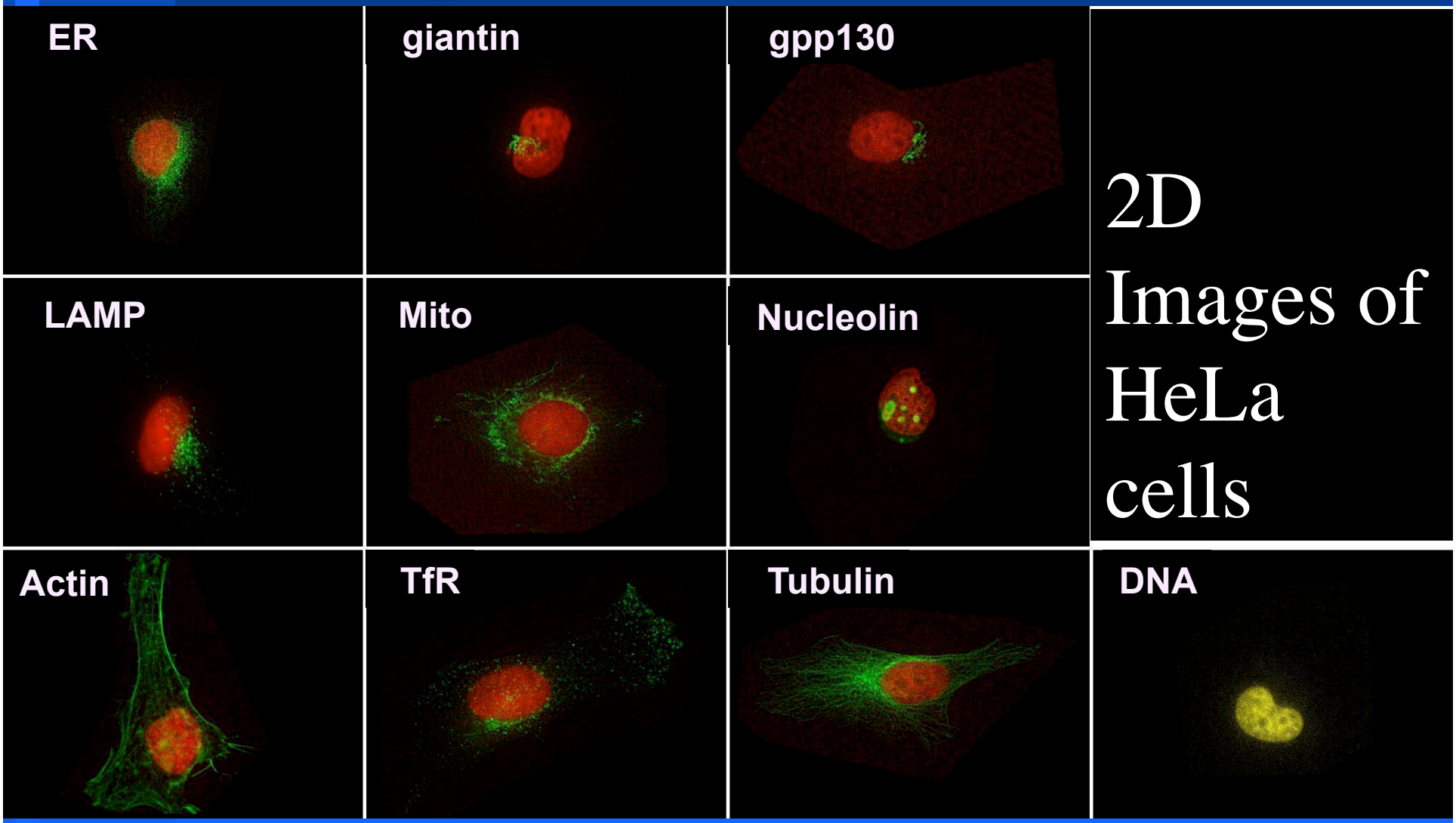




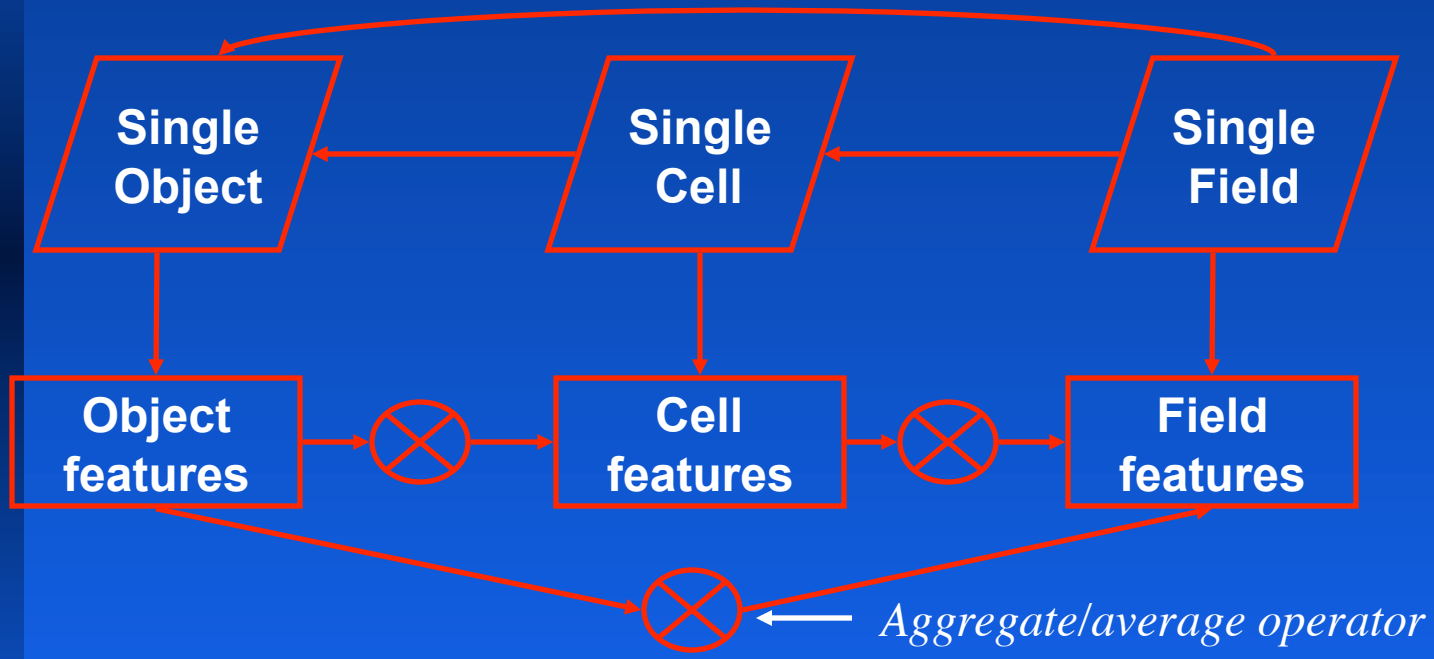
108	# of objects	6
83	Average size of objects	232
31	Average distance to COF	4

Any of these features could be used to distinguish these two classes

The goal: Learn to recognize all major subcellular patterns



Feature levels and granularity



Granularity: 2D, 3D, 2Dt, 3Dt

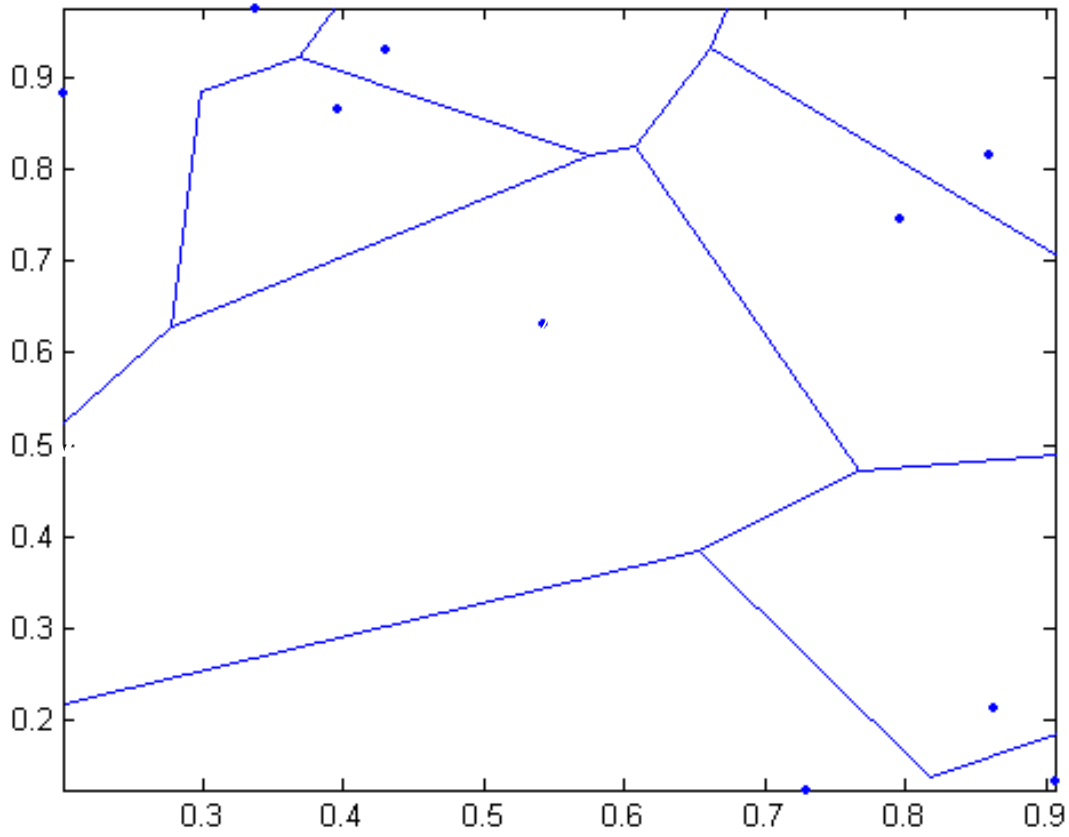
Cell Segmentation

Single cell segmentation approaches

- Voronoi
- Watershed
- Seeded Watershed
- Level Set Methods
- Graphical Models

Voronoi diagram

Given a set of seeds, draw vertices and edges such that each seed is enclosed in a single polygon where each edge is equidistant from the seeds on either side.



Voronoi Segmentation Process

- Threshold DNA image (downsample?)
- Find the objects in the image
- Find the centers of the objects
- Use as seeds to generate Voronoi diagram
- Create a mask for each region in the Voronoi diagram
- Remove regions whose object that does not have intensity/size/shape of nucleus

Thresholding

Gray-level image ⟶ Binary image

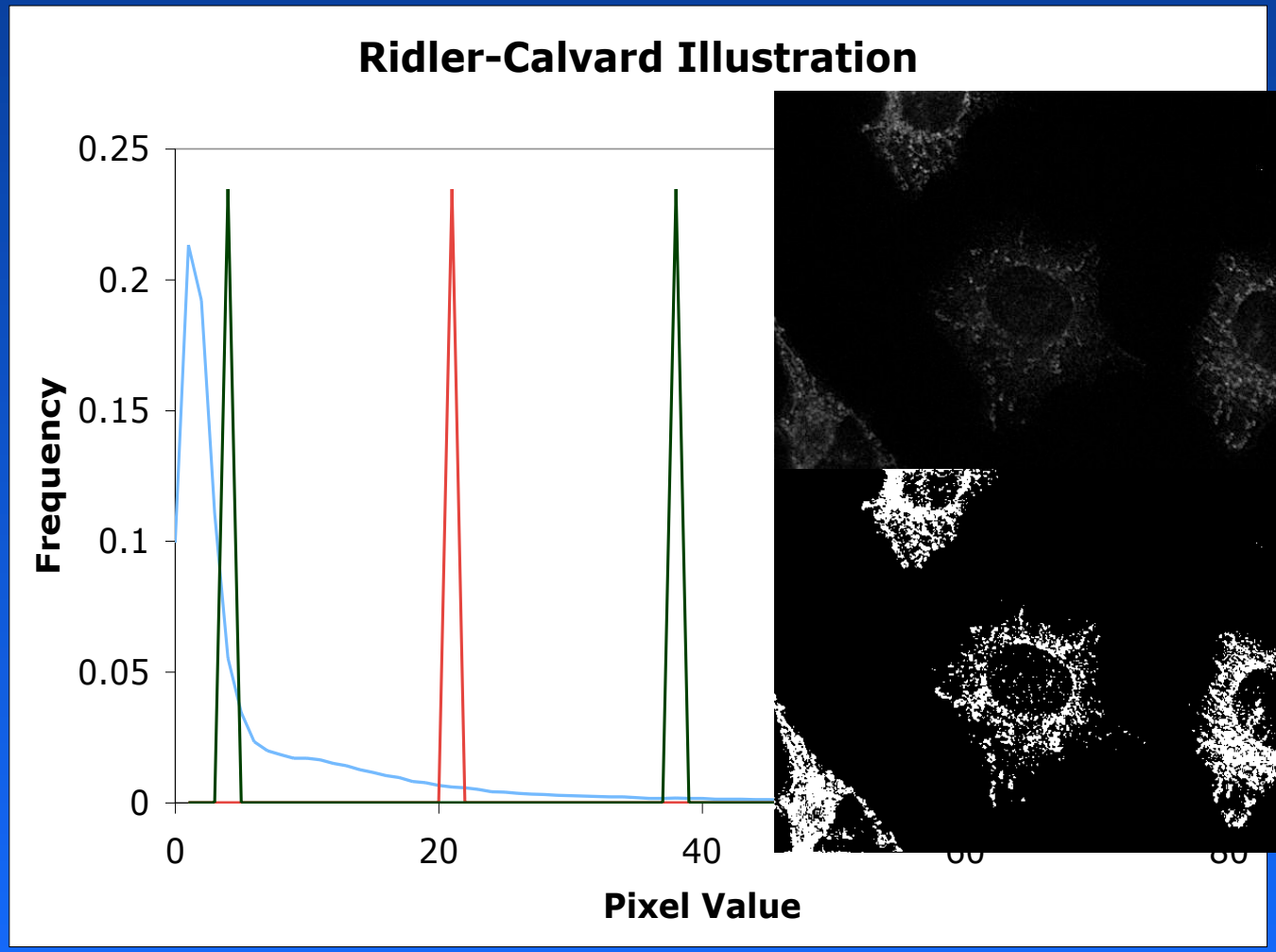
- Thresholding refers to the division of the pixels of an image into two classes: those below a certain value (the **threshold**) and those at or above it. The two classes are often shown in white and black, respectively.
- Thresholding serves as a means to consider only a *subset* of the pixels of an images.

Ridler-Calvard Method

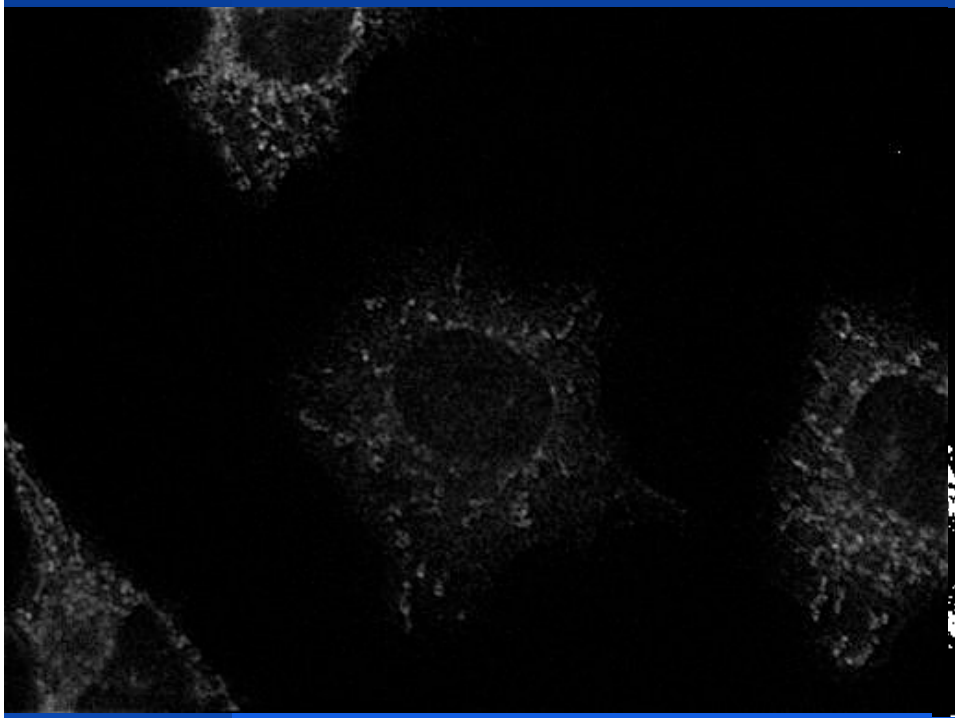
- Find threshold that is equidistant from the average intensity of pixels below and above it
- Ridler, T.W. and Calvard, S. (1978) Picture thresholding using an iterative selection method. *IEEE Transactions on Systems, Man, and Cybernetics* 8:630-632.

Ridler-Calvard Method

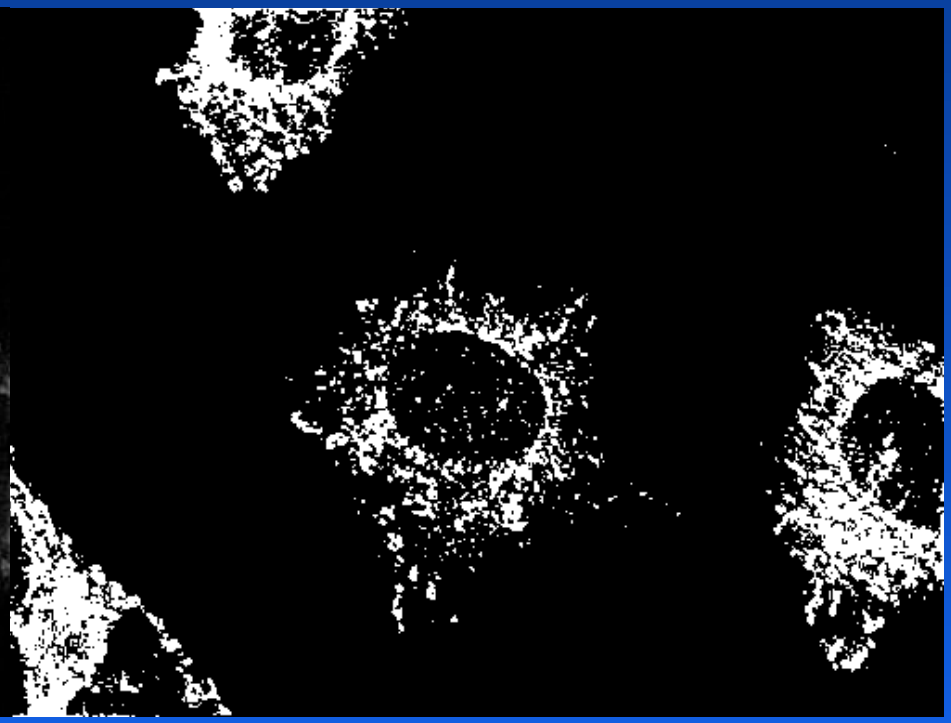
Blue line shows histogram of intensities, green lines show average to left and right of red line, red line shows midpoint between them or the RC threshold



Ridler-Calvard Method

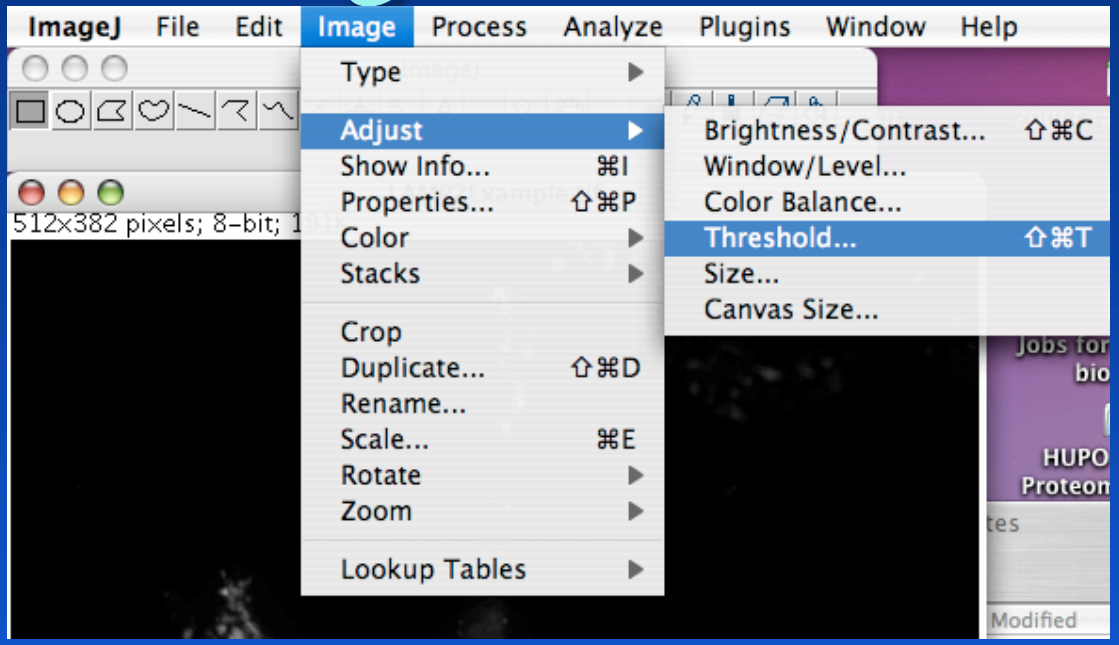


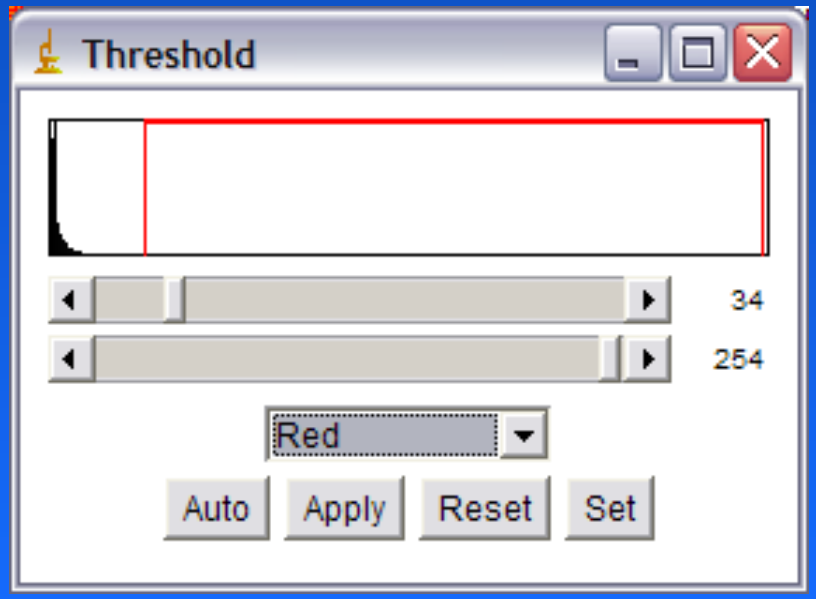
original



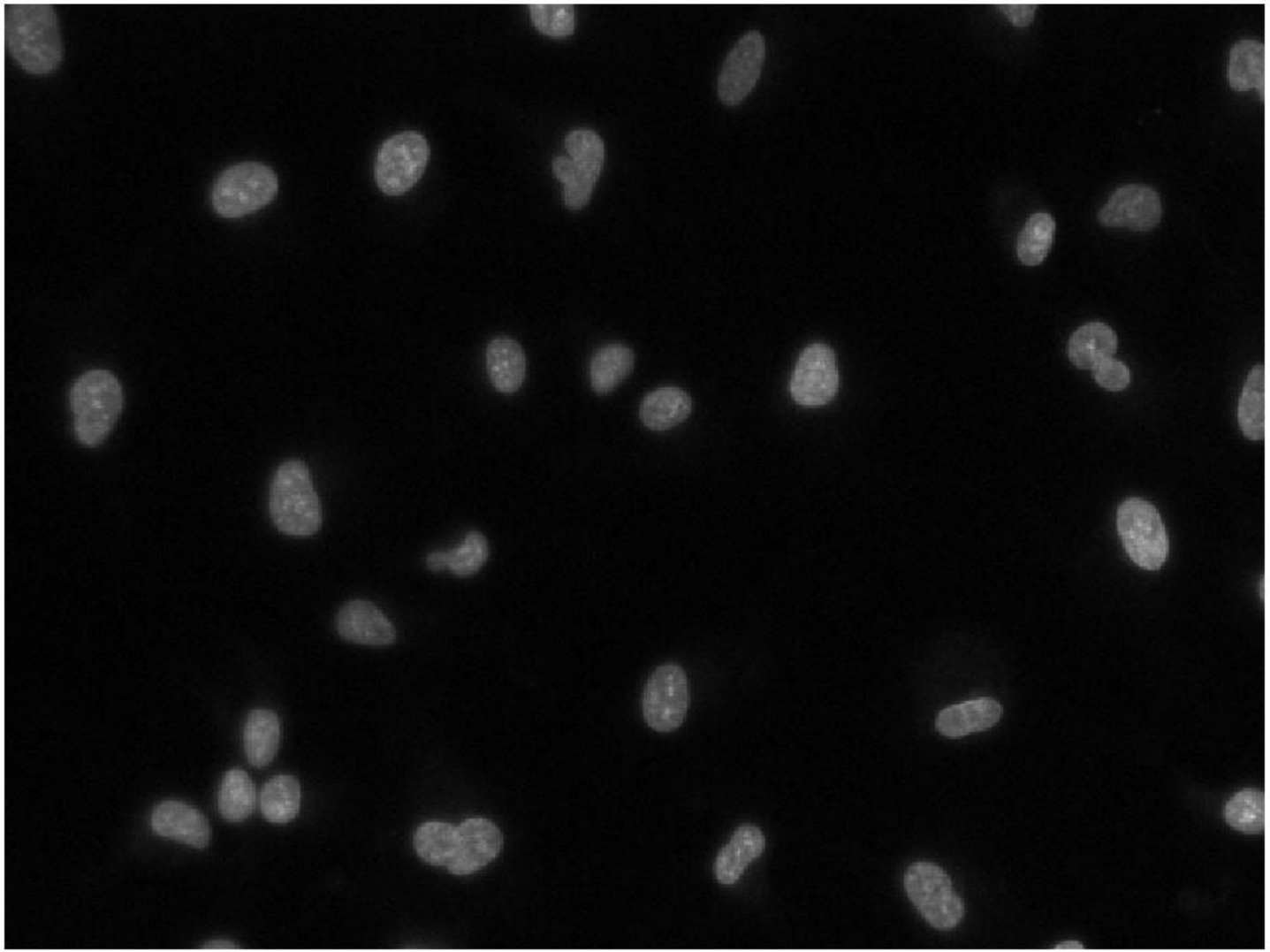
thresholded

Thresholding

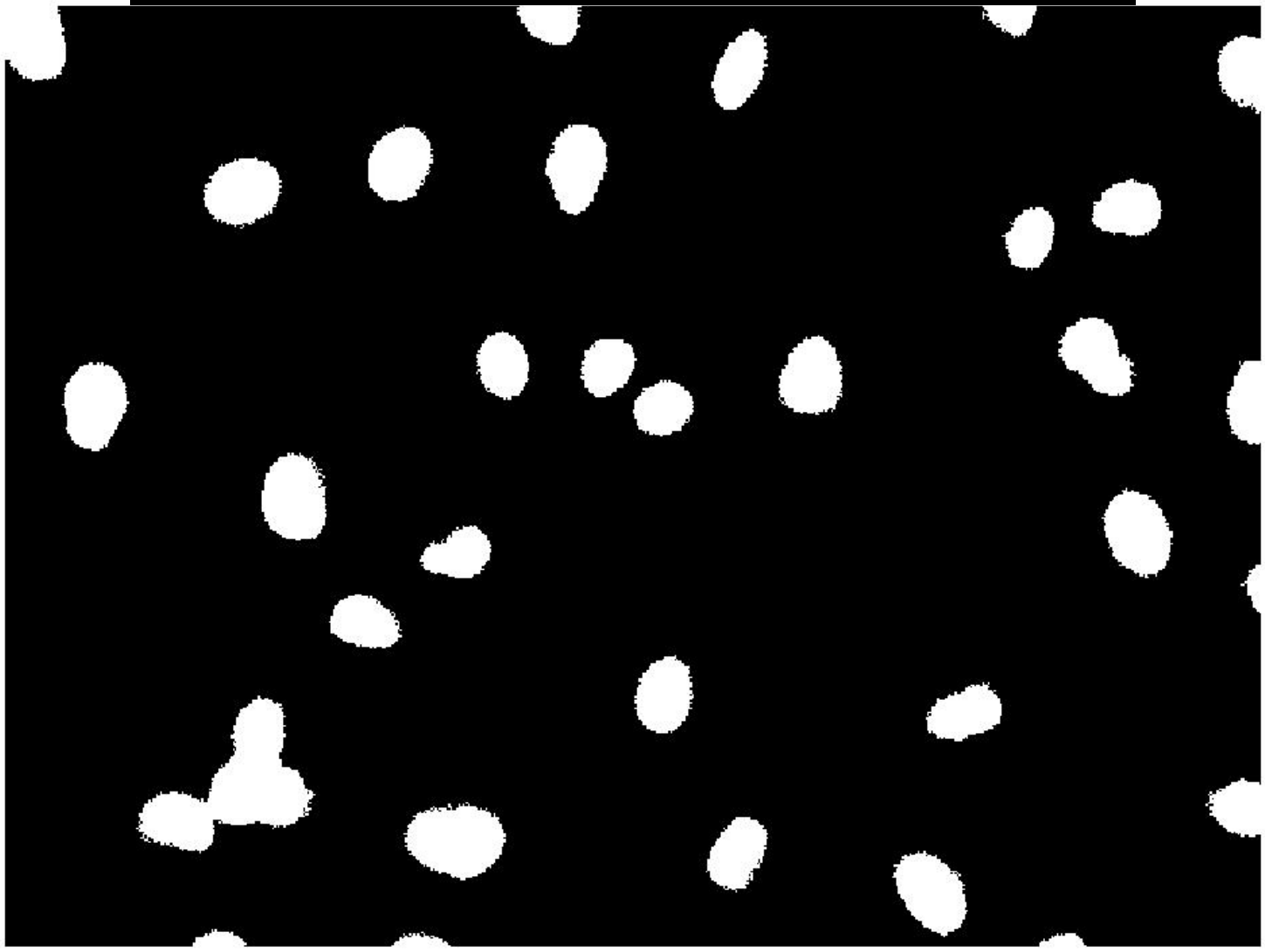




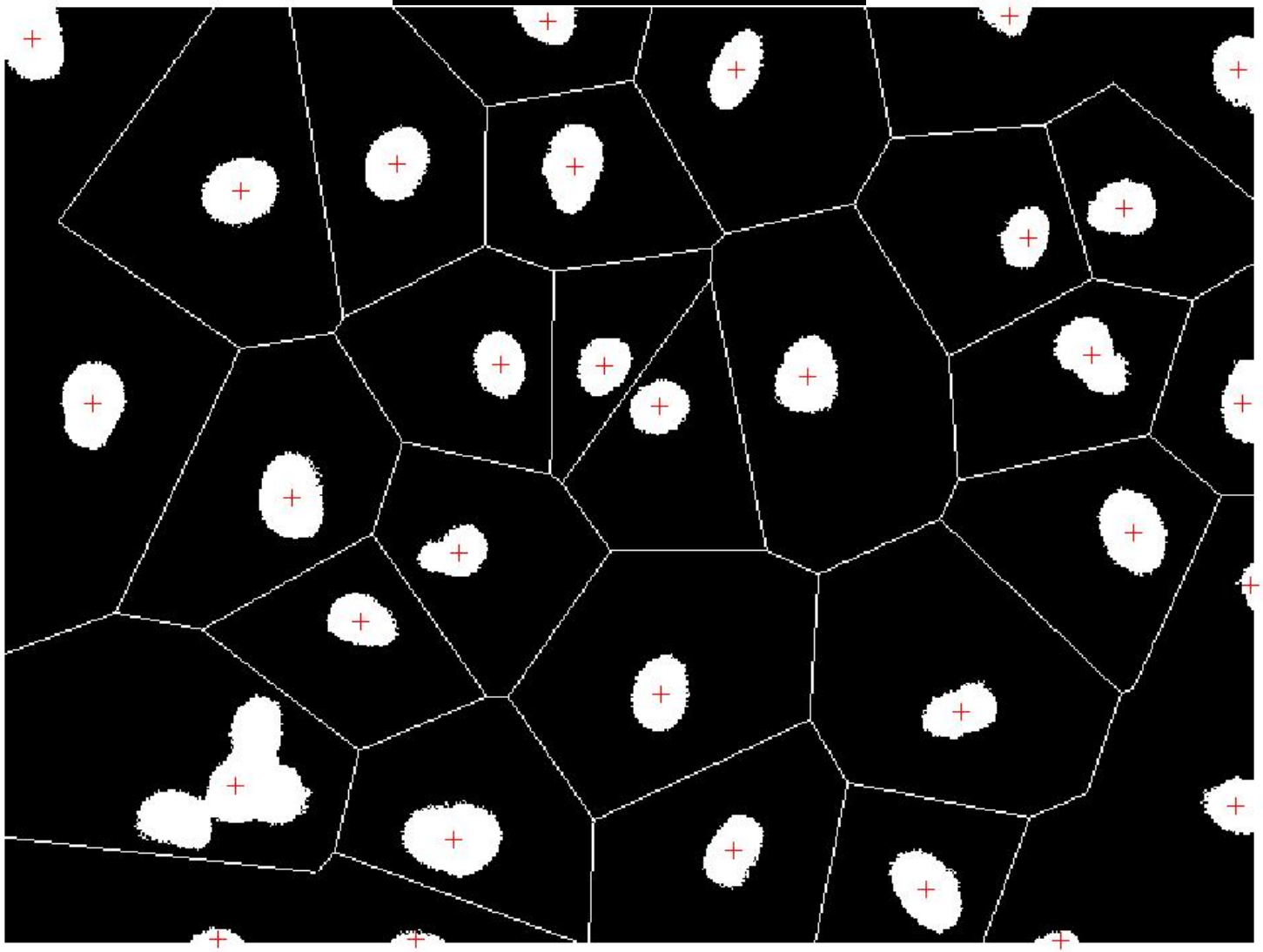
Original DNA image



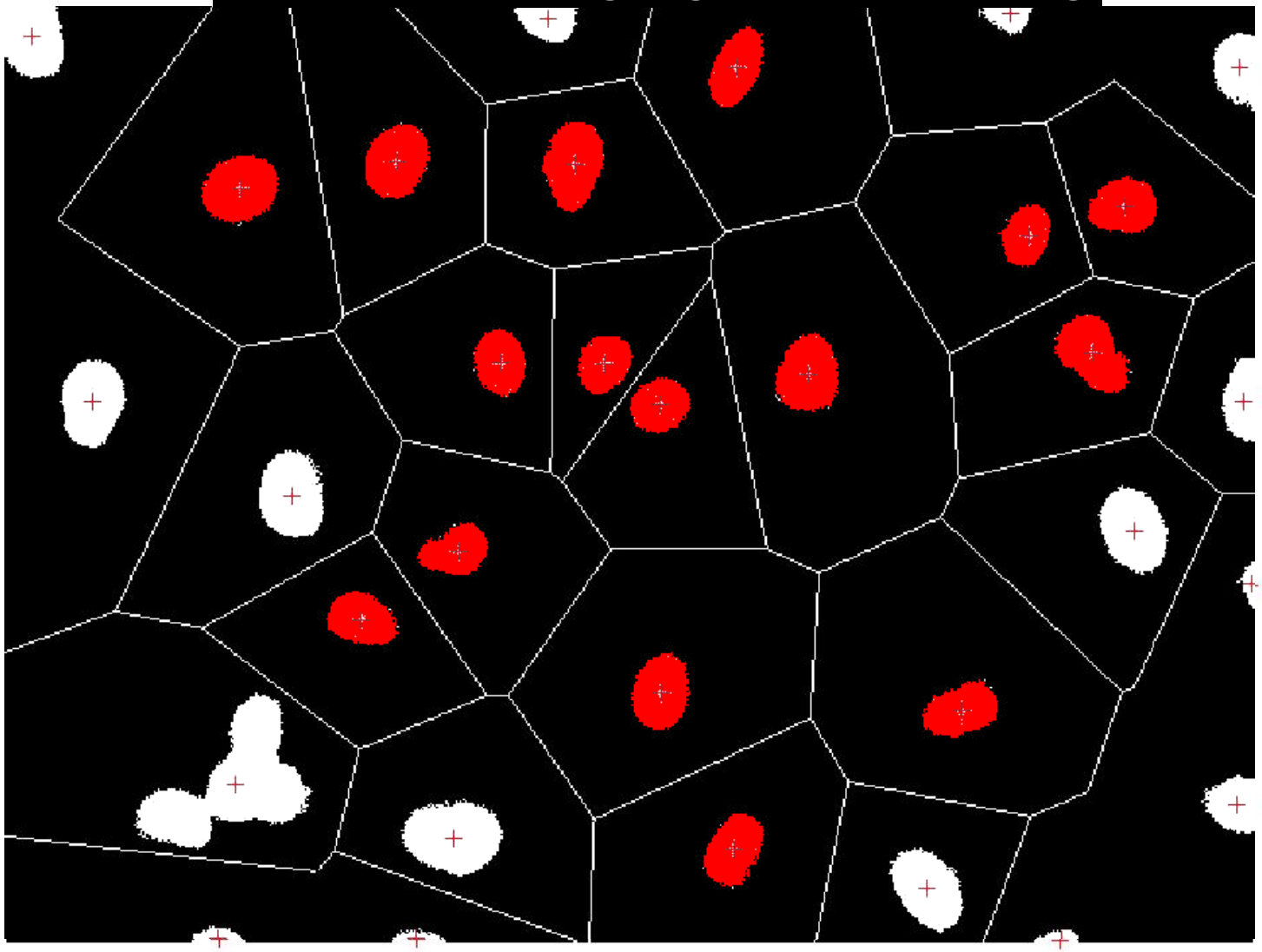
After thresholding and removing small objects



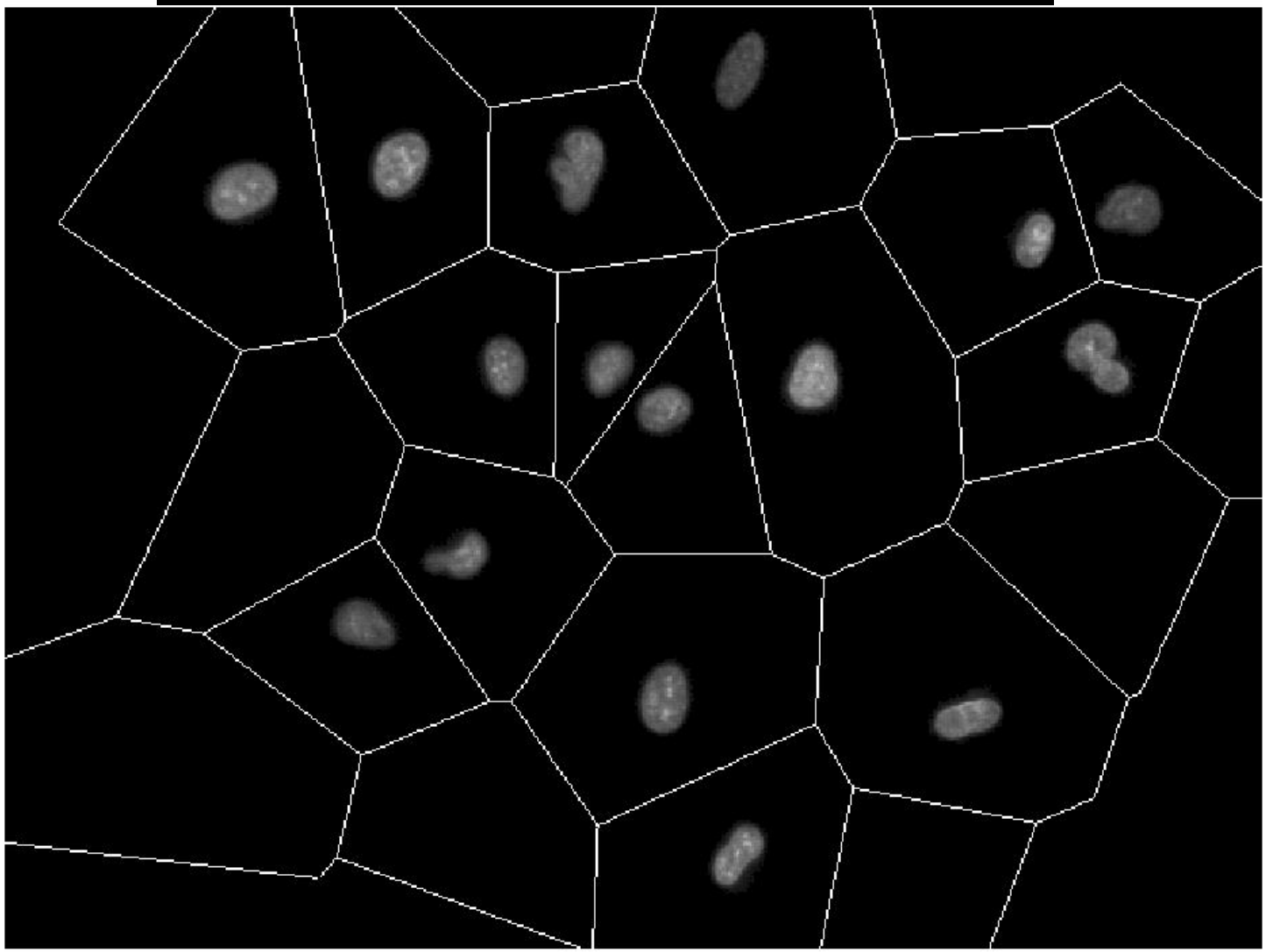
After triangulation



After removing edge cells and filtering

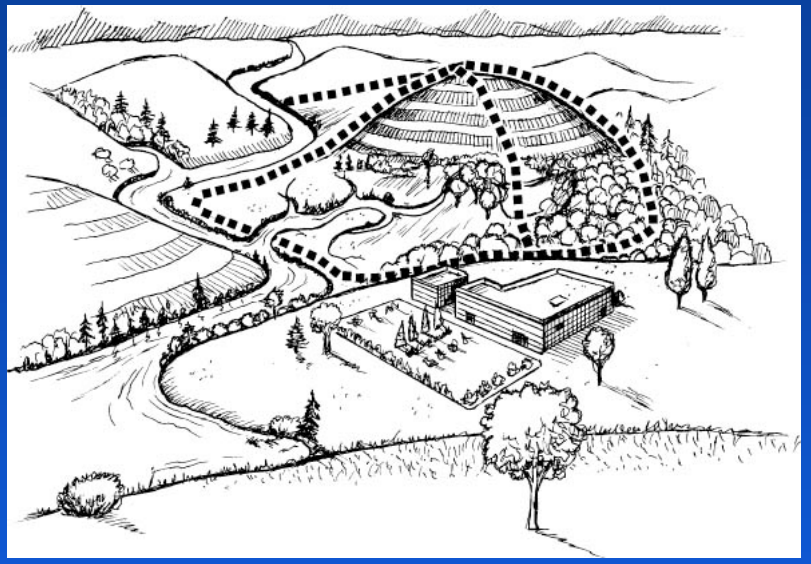


Final regions masked onto original image



Watershed Segmentation

- Intensity of an image ~ elevation in a landscape
  - Flood from minima
  - Prevent merging of "catchment basins"
  - Watershed borders built at contacts between basins

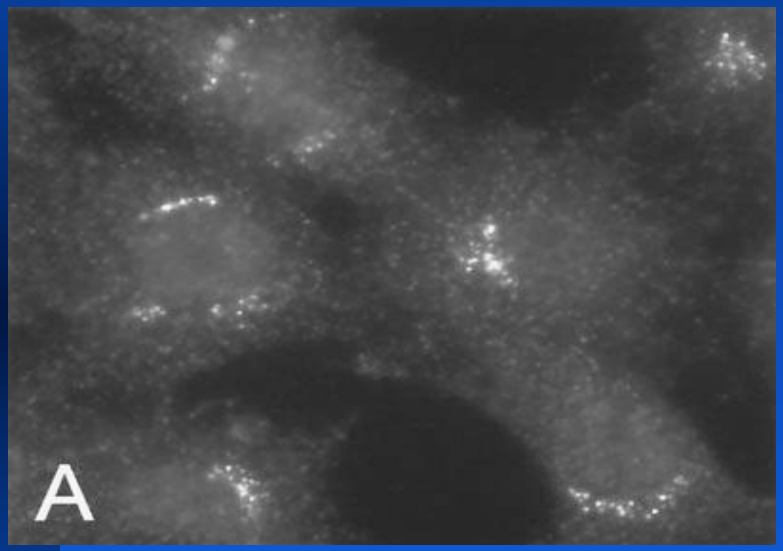


http://www.ctic.purdue.edu/KYW/glossary/whatisaws.html

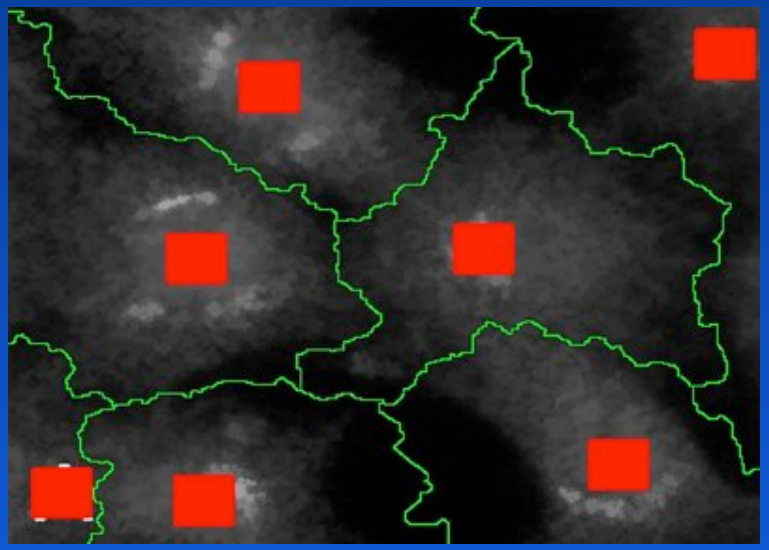
Seeded Watershed Segmentation

- Drawback is that the number of regions may not correspond to the number of cells
- Seeded watershed allows water to rise only from predefined sources (seeds)
- If DNA image available, can use same approach to generate these seeds as for Voronoi segmentation
- Can use seeds from DNA image but use total protein image or plasma membrane protein image for watershed segmentation

Seeded Watershed Segmentation



Original image



Seeds and boundary

Applied directly to protein image (no DNA image)

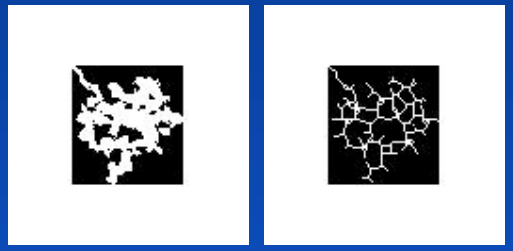
Note non-linear boundaries

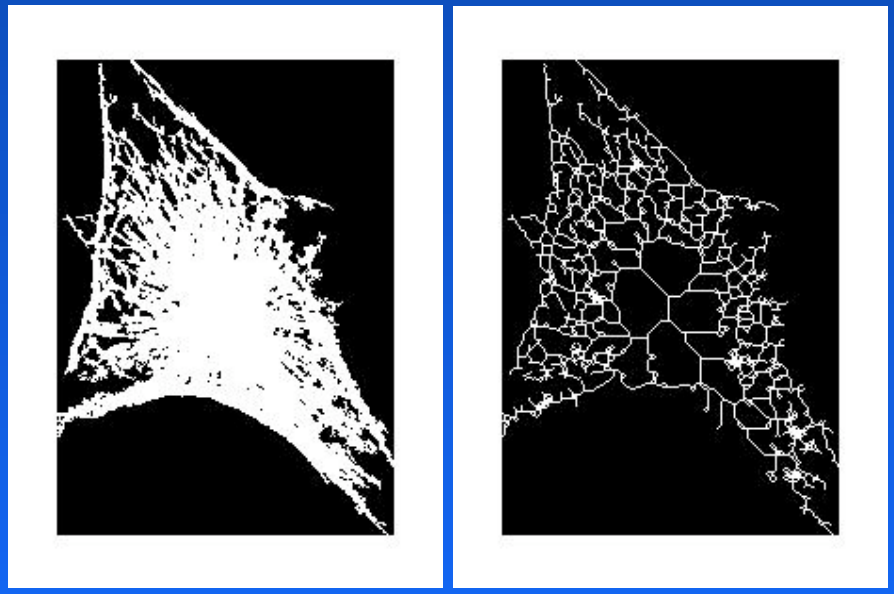
Feature Extraction

2D Subcellular Location Features

- Morphological (based on objects after thresholding)
  - Object number
  - Object size
  - Object shape (including skeleton features)
  - Object position
  - Object overlap with marker (DNA)
- Edge (amount, preferred orientation)
- Moments (Zernike)
- Texture (Haralick)
- Transform

Illustration – Skeleton





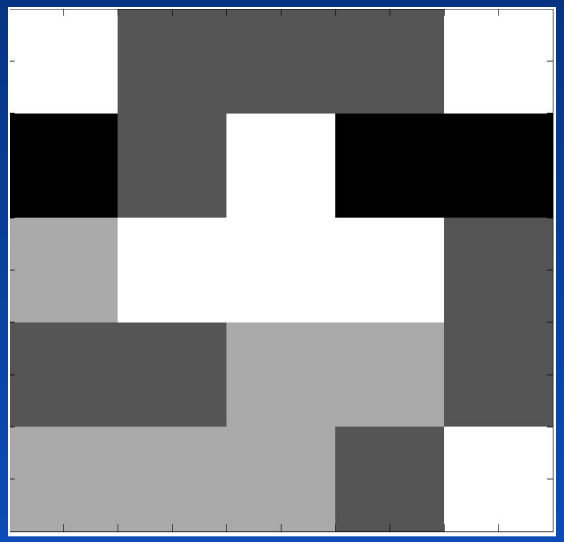
Haralick Texture Features

- Correlations of adjacent pixels in gray level images
- Start by calculating co-occurrence matrix P:

  N by N matrix, N=number of gray level.

  Element P(i,j) is the probability of pixels with value i being adjacent with pixels with value j
- Four directions in which a pixel can be adjacent

Co-occurrence Matrix



4	2	2	2	4
1	2	4	1	1
3	4	4	4	2
2	2	3	3	2
3	3	3	2	4

↕

	1	2	3	4
1	0	2	1	3
2	2	4	4	3
3	1	4	2	2
4	3	3	2	2

↔

	1	2	3	4
1	2	1	0	1
2	1	6	3	4
3	0	3	6	2
4	1	4	2	4

↗

	1	2	3	4
1	0	1	0	3
2	1	4	3	3
3	0	3	4	1
4	3	3	1	2

↘

	1	2	3	4
1	0	3	0	1
2	3	0	4	4
3	0	4	0	3
4	1	4	3	2

Pixel Resolution and Gray Levels

- Texture features are influenced by the number of gray levels and pixel resolution of the image
- Optimization for each image dataset required
- Alternatively, features can be calculated for many resolutions

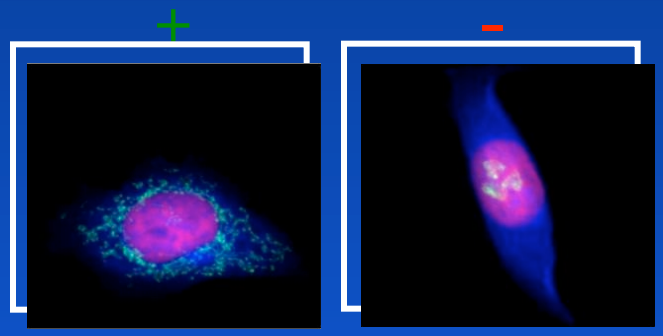
Transform features

- Can apply an image transform and then calculate features
  - Fourier transform
  - Wavelet transforms

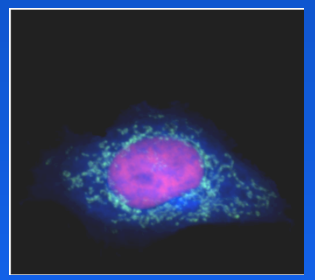
Feature selection

- Having too many features can confuse a classifier
- Can use comparison of feature distributions between classes to choose a subset of features that gets rid of uninformative or redundant features
- Some methods
  - Principal Components Analysis
  - Non-Linear Principal Components Analysis
  - Independent Components Analysis
  - Information Gain
  - Stepwise Discriminant Analysis

Simple two class problem



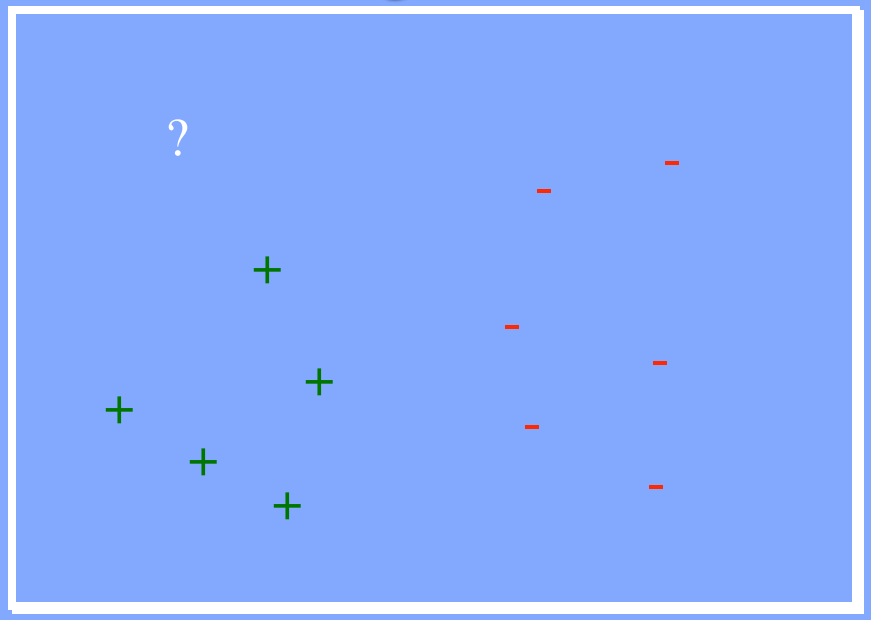
???



Describe each image by features
Train classifier

Classification illustration

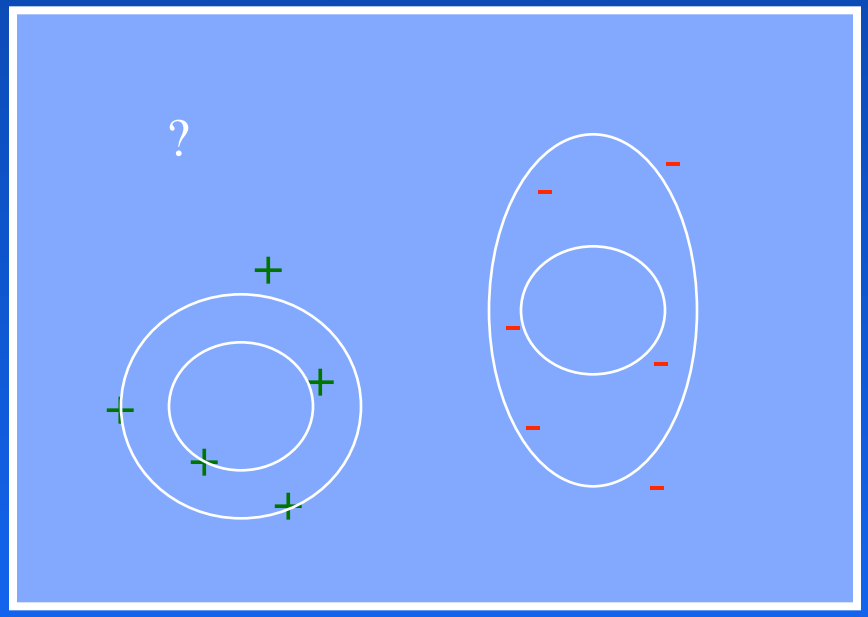
- Given + and – images, we want to label '?'



Linear Discriminants

- Fit multivariate Gaussian to each class
- Measure distance from ? to each Gaussian

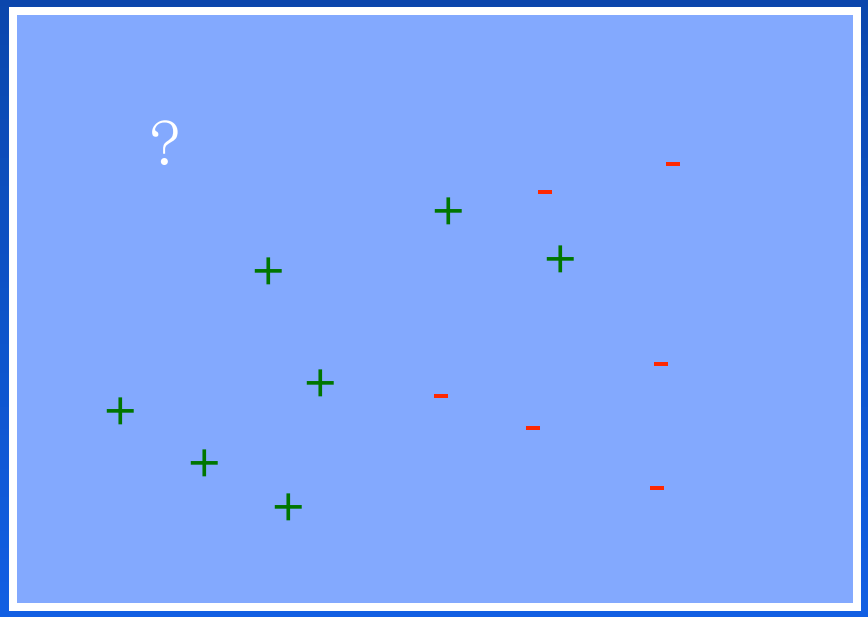
bright.



area

Decision trees

- Again we want to label '?'



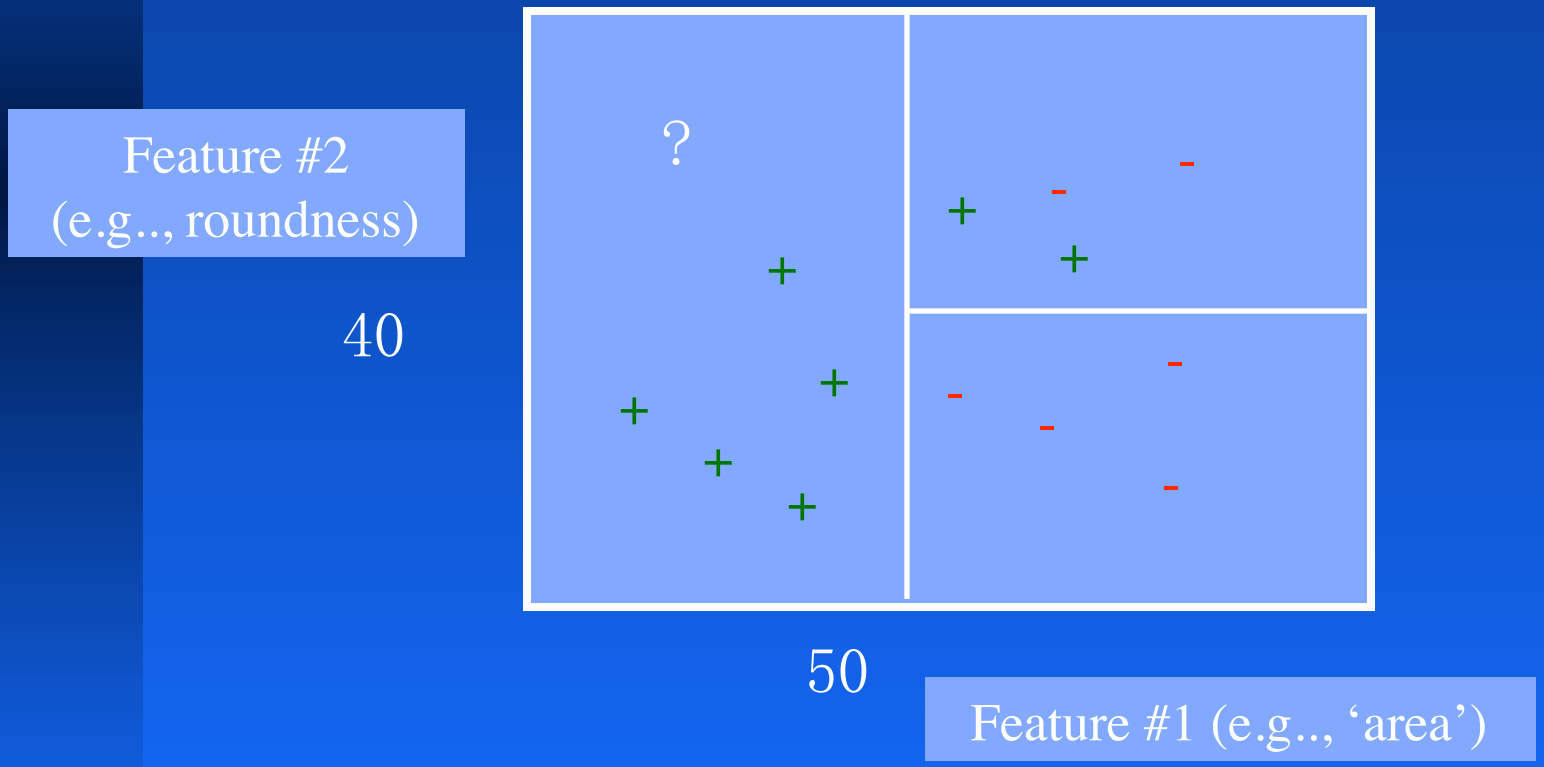
Feature #2
(e.g.., roundness)

Feature #1 (e.g.., 'area')

Slide courtesy of Christos Faloutsos

Decision trees

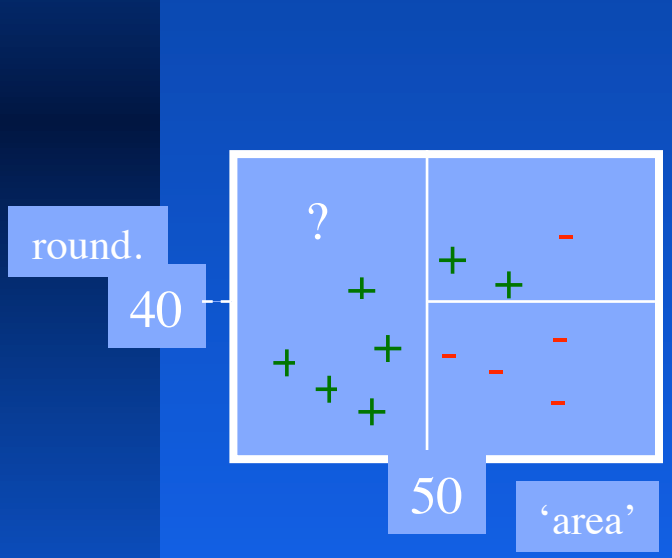
- so we build a decision tree:

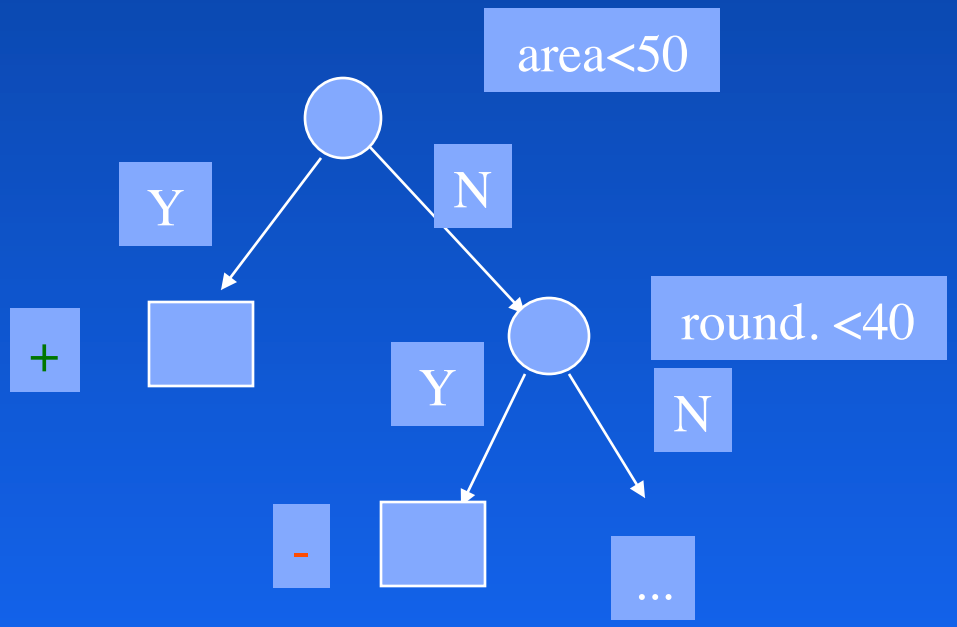


Slide courtesy of Christos Faloutsos

Decision trees

- so we build a decision tree:

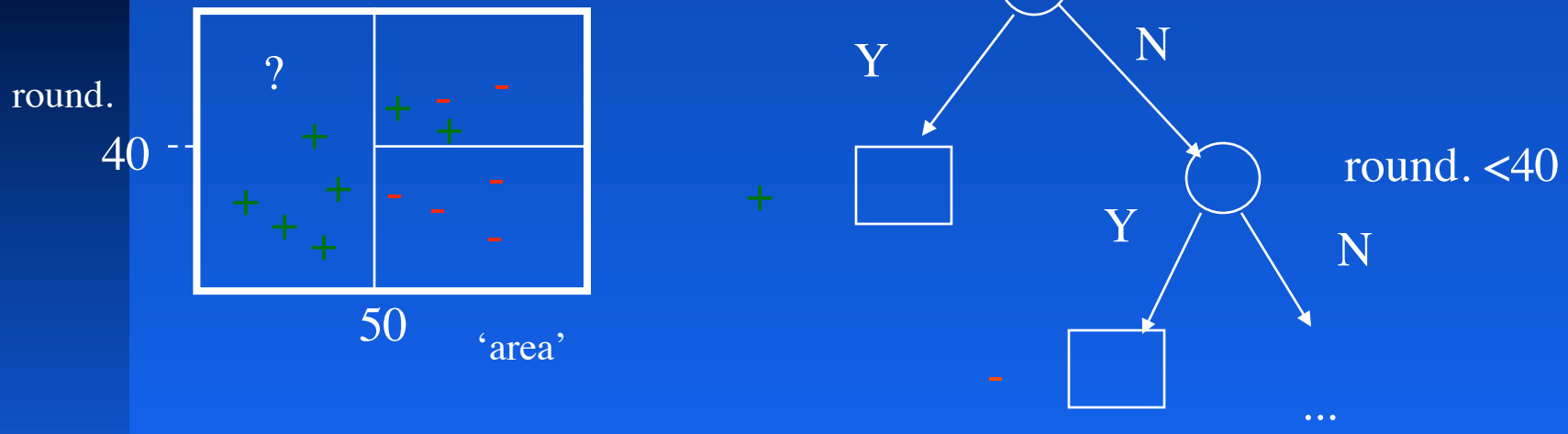




Slide courtesy of Christos Faloutsos

Decision trees

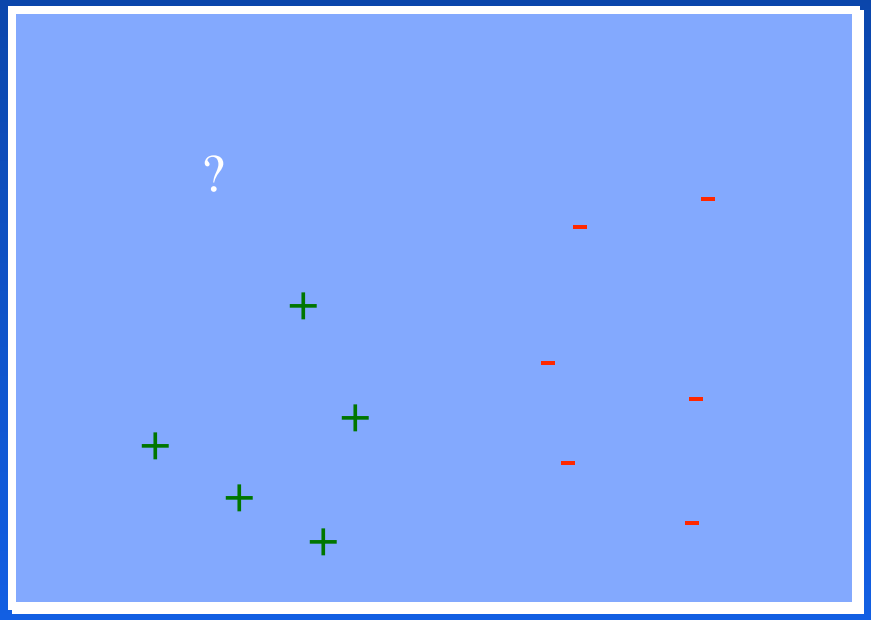
- Goal: split address space in (almost) homogeneous regions



Slide courtesy of Christos Faloutsos

Support vector machines

- Again we want to label '?'



Feature #2
(e.g.., roundness)

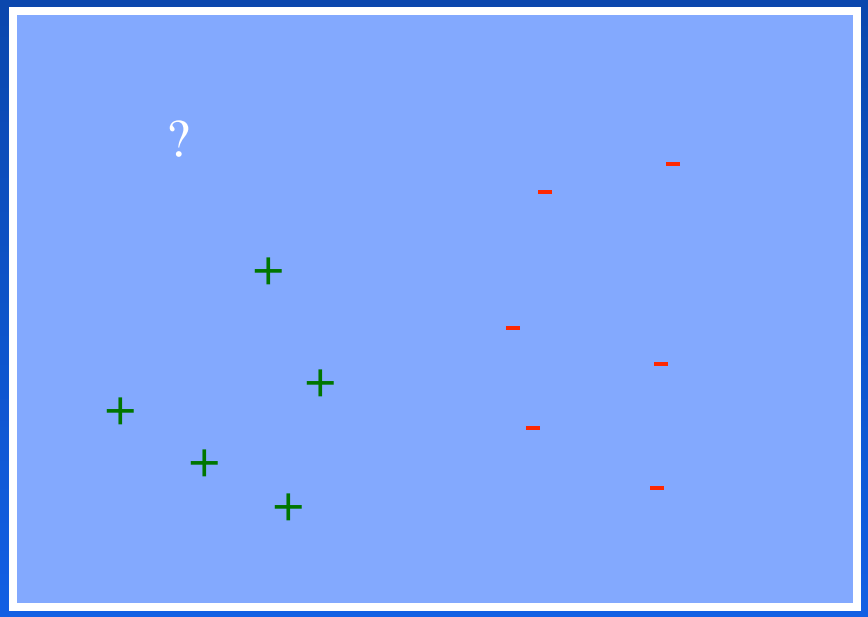
Feature #1 (e.g.., 'area')

Slide courtesy of Christos Faloutsos

Support Vector Machines (SVMs)

- Use single linear separator??

round.



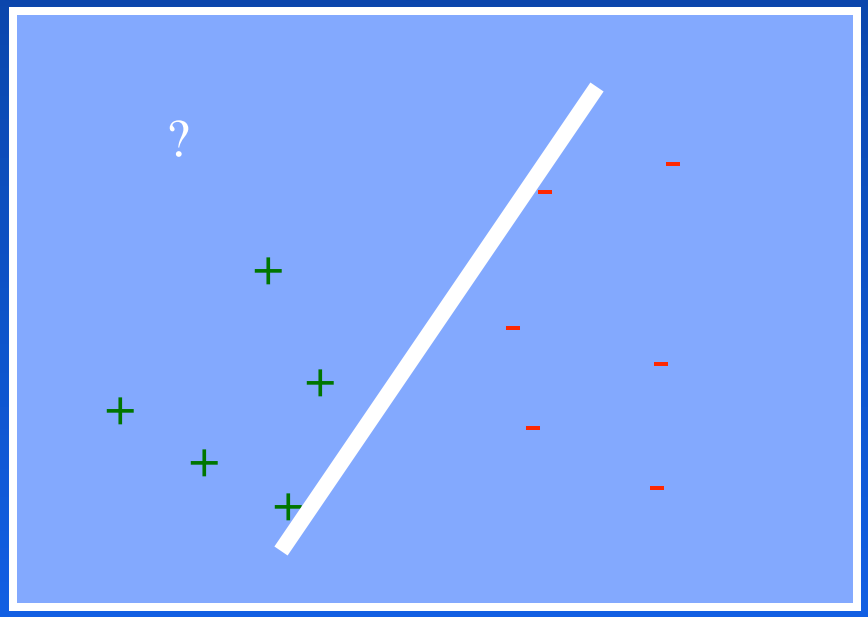
area

Slide courtesy of Christos Faloutsos

Support Vector Machines (SVMs)

- Use single linear separator??

round.



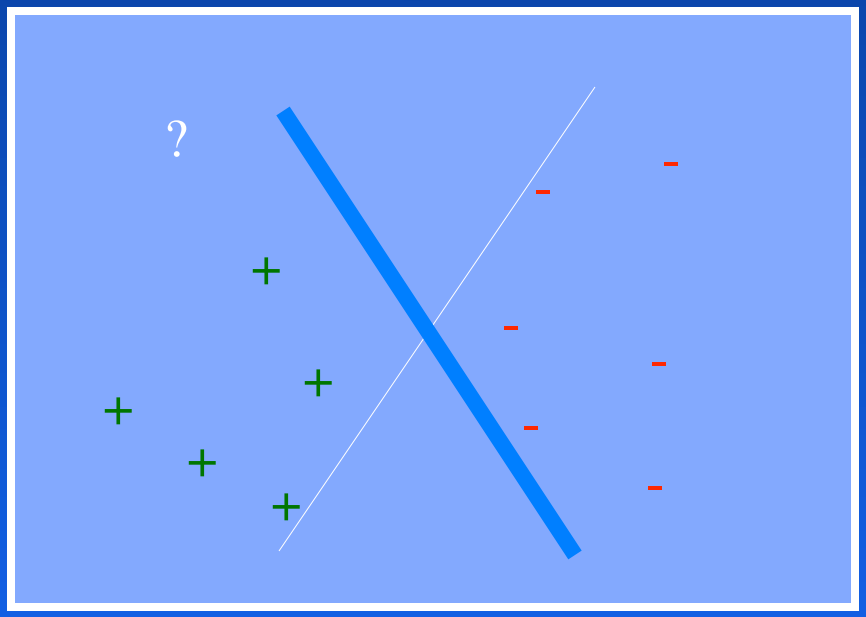
area

Slide courtesy of Christos Faloutsos

Support Vector Machines (SVMs)

- Use single linear separator??

round.



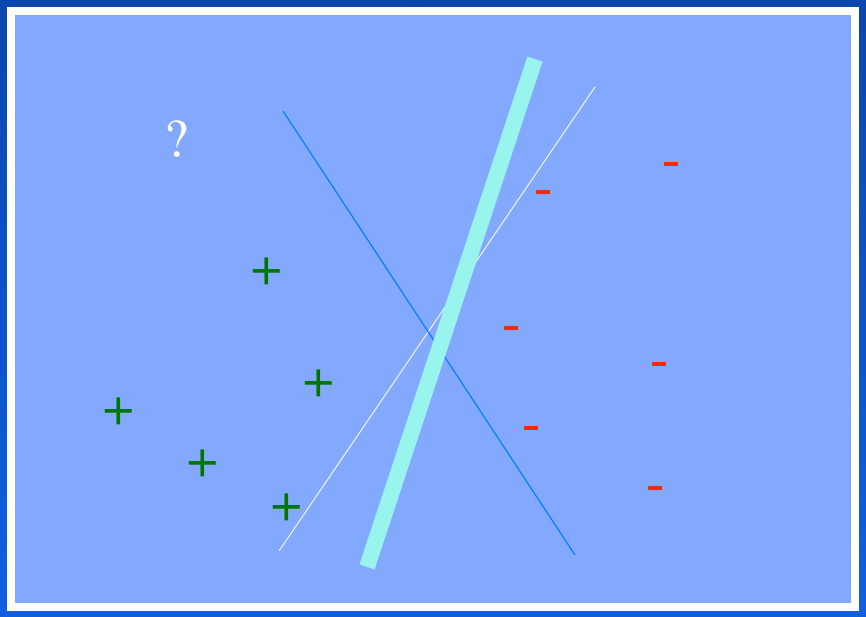
area

Slide courtesy of Christos Faloutsos

Support Vector Machines (SVMs)

- Use single linear separator??

round.



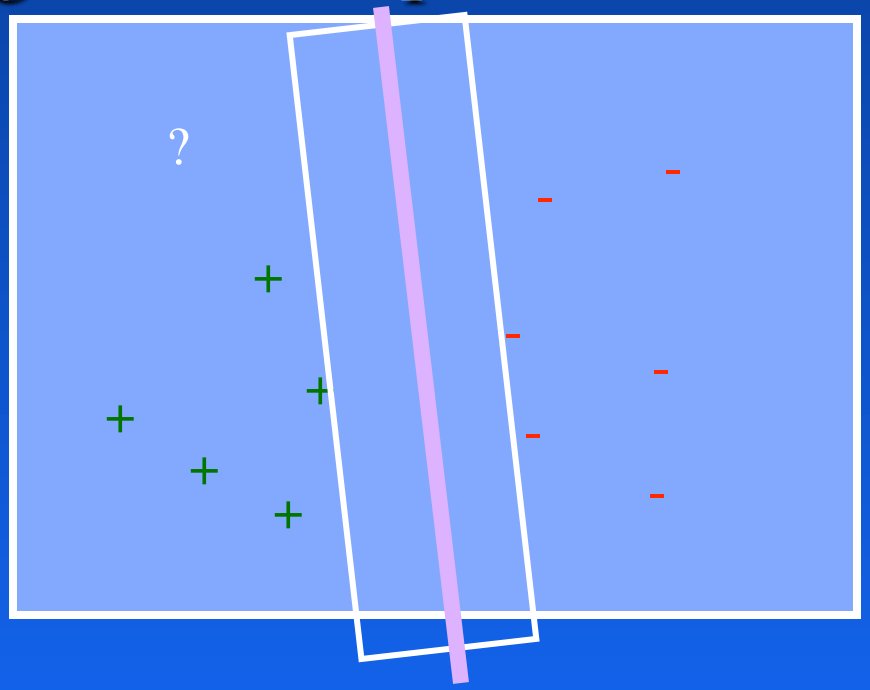
area

Slide courtesy of Christos Faloutsos

Support Vector Machines (SVMs)

- Use single linear separator??

round.



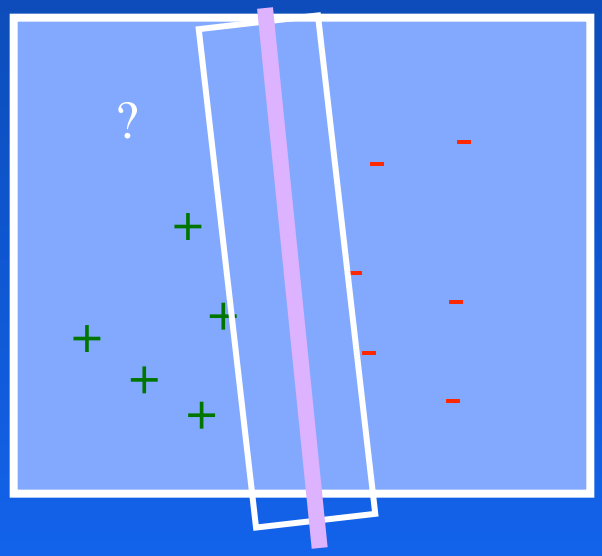
area

Slide courtesy of Christos Faloutsos

Support Vector Machines (SVMs)

- we want to label '**?**' - linear separator??
- A: the one with the widest corridor!

round.



area

Slide courtesy of Christos Faloutsos

Support Vector Machines (SVMs)

- What if the points for each class are not readily separated by a straight line?
- Use the "kernel trick" – project the points into a higher dimensional space in which we hope that straight lines will separate the classes
- "kernel" refers to the function used for this projection

Support Vector Machines (SVMs)

- Definition of SVMs explicitly considers only two classes
- What if we have more than two classes?
- Train multiple SVMs
- Two basic approaches
  - One against all (one SVM for each class)
  - Pairwise SVMs (one for each pair of classes)
    - Various ways of implementing this

Cross-Validation

- If we train a classifier to minimize error on a set of data, have no ability to estimate (generalize) error that will be seen on new dataset
- To calculate *generalizable* accuracy, we use ***n*-fold cross-validation**
- Divide images into *n* sets, train using *n*-1 of them and test on the remaining set
- Repeat until each set is used as test set and average results across all trials
- Variation on this is called ***leave-one-out***

2D Classification Results

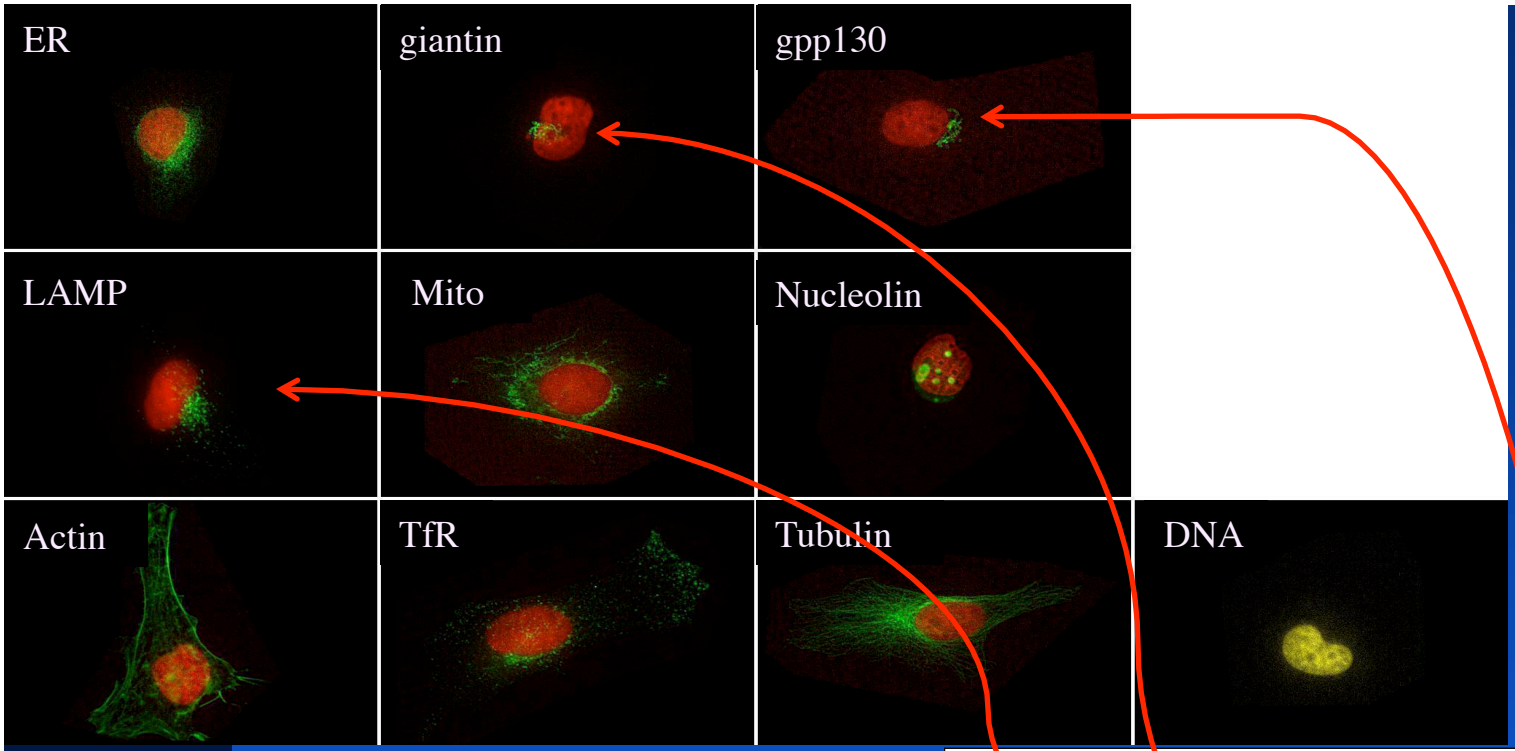
True Class	Output of the Classifier									
	DNA	ER	Gia	Gpp	Lam	Mit	Nuc	Act	TfR	Tub
DNA	**99**	1	0	0	0	0	0	0	0	0
ER	0	**97**	0	0	0	2	0	0	0	1
Gia	0	0	**91**	7	0	0	0	0	2	0
Gpp	0	0	14	**82**	0	0	2	0	1	0
Lam	0	0	1	0	**88**	1	0	0	10	0
Mit	0	3	0	0	0	**92**	0	0	3	3
Nuc	0	0	0	0	0	0	**99**	0	1	0
Act	0	0	0	0	0	0	0	**100**	0	0
TfR	0	1	0	0	12	2	0	1	**81**	2
Tub	1	2	0	0	0	1	0	0	1	**95**

Overall accuracy = 92%

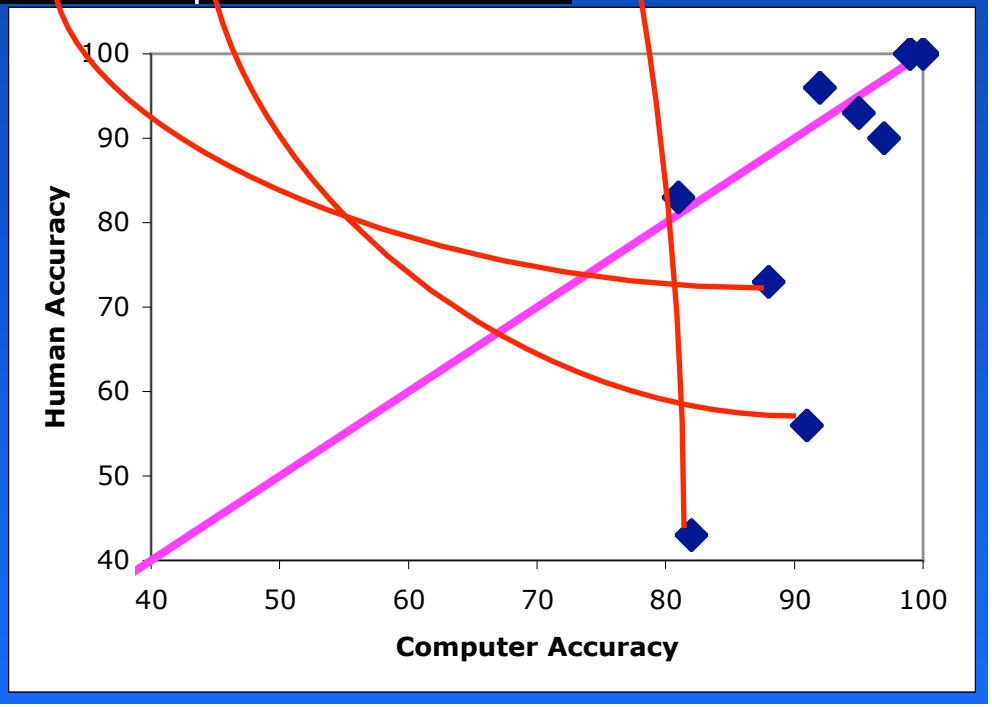
Human Classification Results

True Class	Output of the Classifier									
	DNA	ER	Gia	Gpp	Lam	Mit	Nuc	Act	TfR	Tub
DNA	**100**	0	0	0	0	0	0	0	0	0
ER	0	**90**	0	0	3	6	0	0	0	0
Gia	0	0	**56**	36	3	3	0	0	0	0
Gpp	0	0	54	**33**	0	0	0	0	3	0
Lam	0	0	6	0	**73**	0	0	0	20	0
Mit	0	3	0	0	0	**96**	0	0	0	3
Nuc	0	0	0	0	0	0	**100**	0	0	0
Act	0	0	0	0	0	0	0	**100**	0	0
TfR	0	13	0	0	3	0	0	0	**83**	0
Tub	0	3	0	0	0	0	0	3	0	**93**

Overall accuracy = 83%



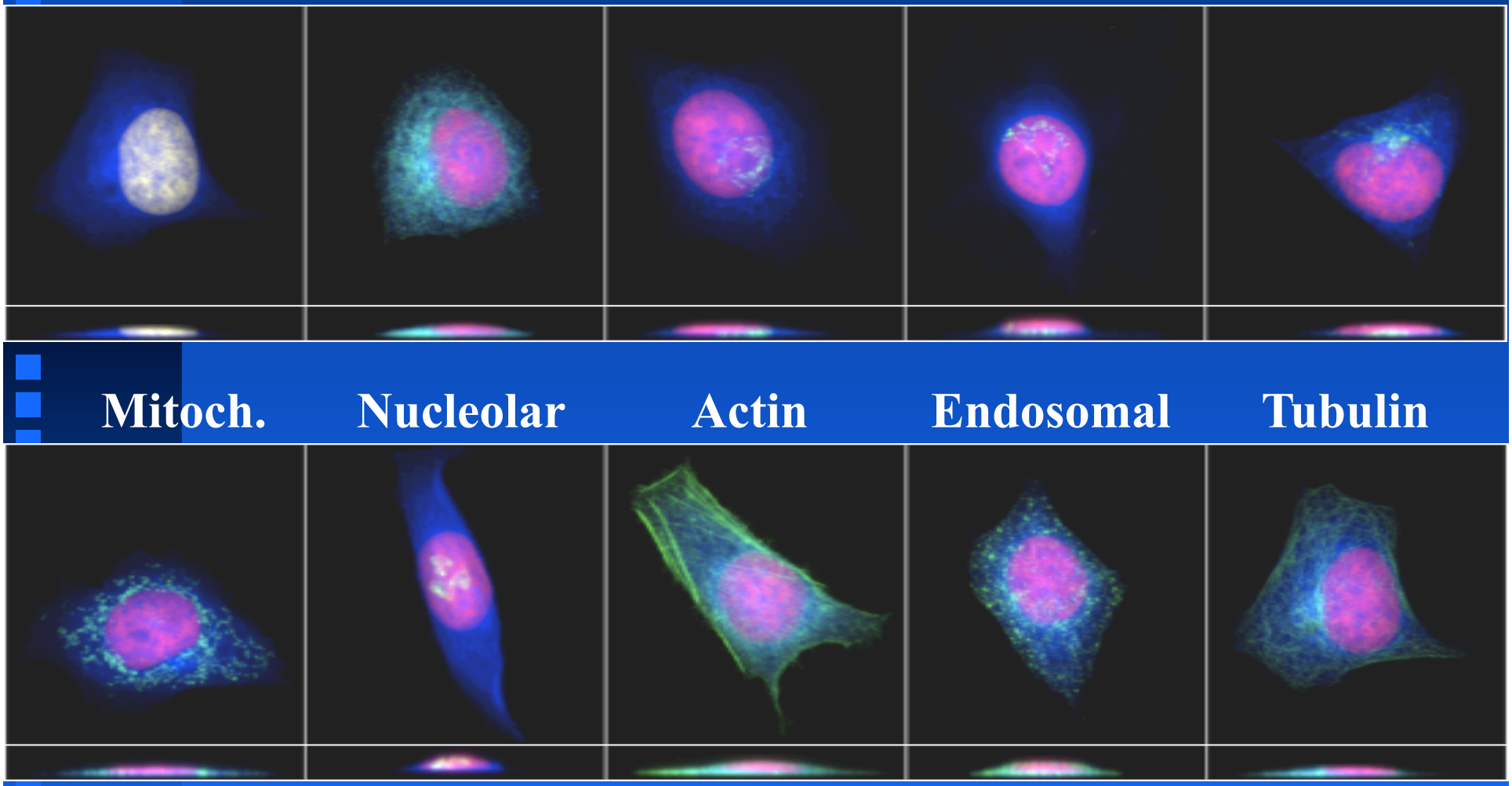
Subcellular Pattern Classification: Computer vs. Human



Even better results using multiresolution methods

Even better results for 3D images

3D HeLa cell images



Images collected using facilities at the Center for Biologic Imaging courtesy of Simon Watkins

3D Classification Results

True Class	Output of the Classifier									
	DNA	ER	Gia	Gpp	Lam	Mit	Nuc	Act	TfR	Tub
DNA	**98**	2	0	0	0	0	0	0	0	0
ER	0	**100**	0	0	0	0	0	0	0	0
Gia	0	0	**100**	0	0	0	0	0	0	0
Gpp	0	0	0	**96**	4	0	0	0	0	0
Lam	0	0	0	4	**95**	0	0	0	0	2
Mit	0	0	2	0	0	**96**	0	2	0	0
Nuc	0	0	0	0	0	0	**100**	0	0	0
Act	0	0	0	0	0	0	0	**100**	0	0
TfR	0	0	0	0	2	0	0	0	**96**	2
Tub	0	2	0	0	0	0	0	0	0	**98**

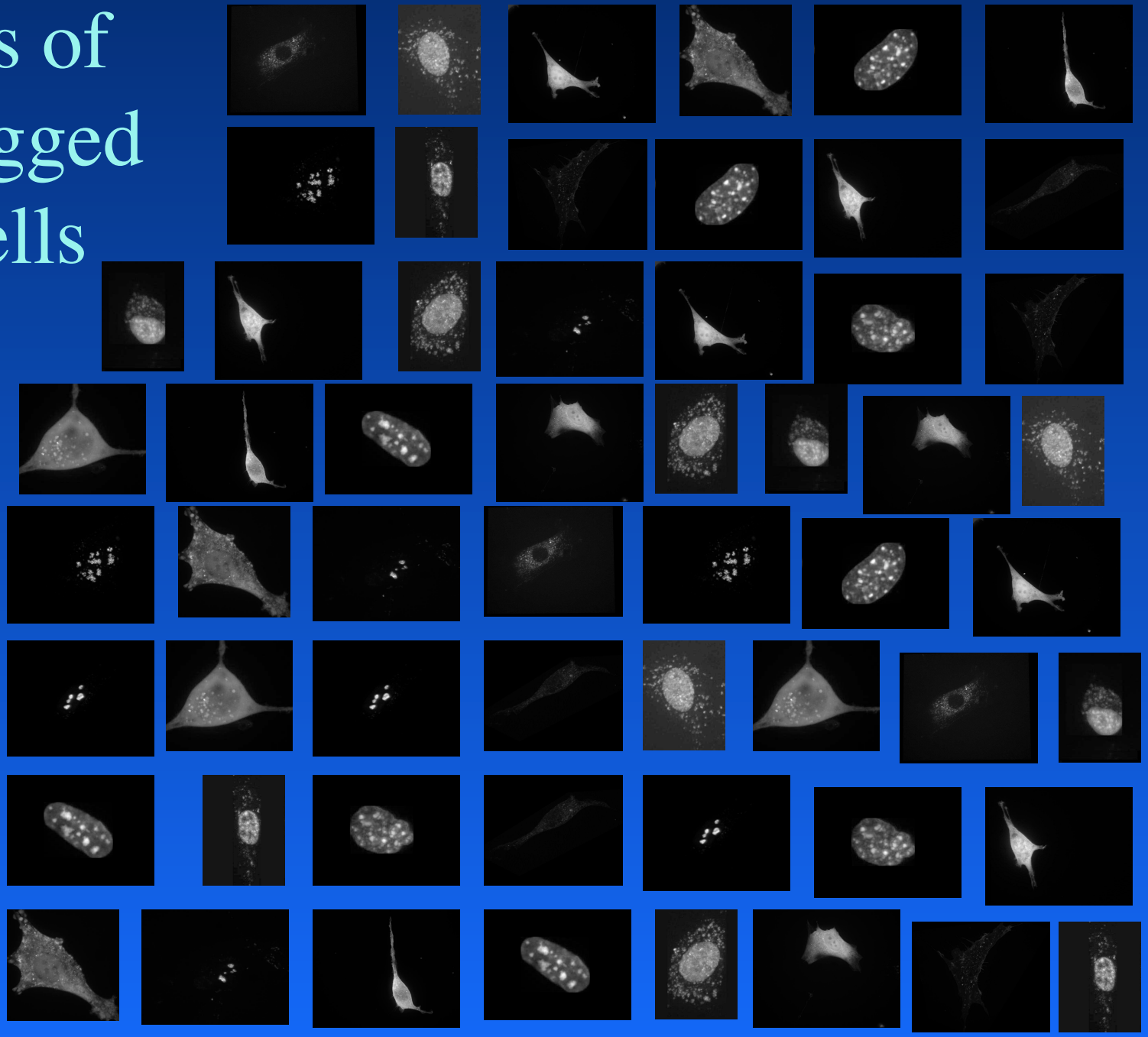
Overall accuracy = 98%

High content screening/analysis

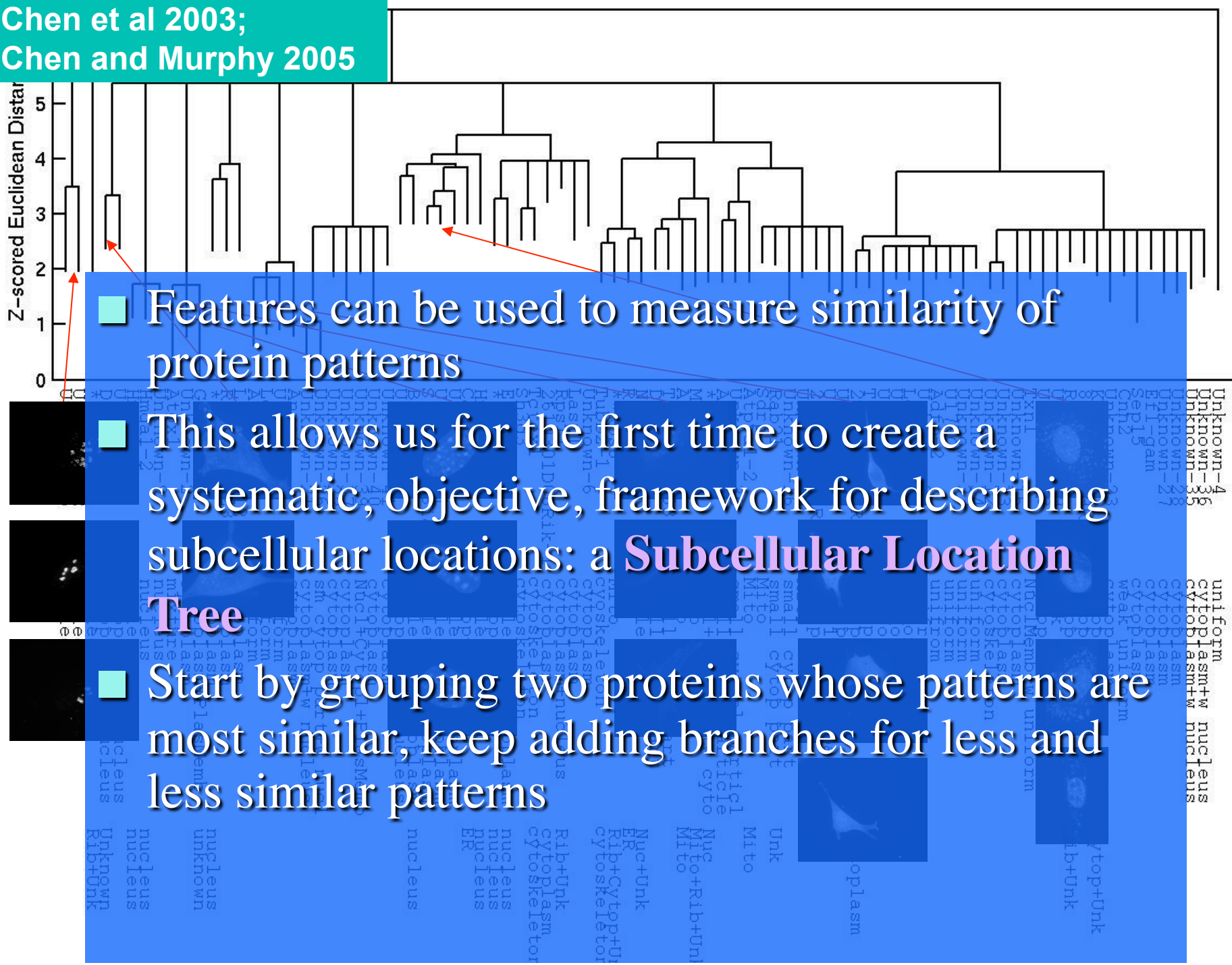
- Commercially available systems for automated microscopy, coupled with systems for analyzing images
- Typically involve segmentation, feature calculation, classification
- Typically involve hand-tuned feature sets and classifiers
- Mainly used for drug screening

Unsupervised Learning to Identify High-Resolution Protein Patterns

Images of CD-tagged 3T3 cells



Chen et al 2003;
Chen and Murphy 2005



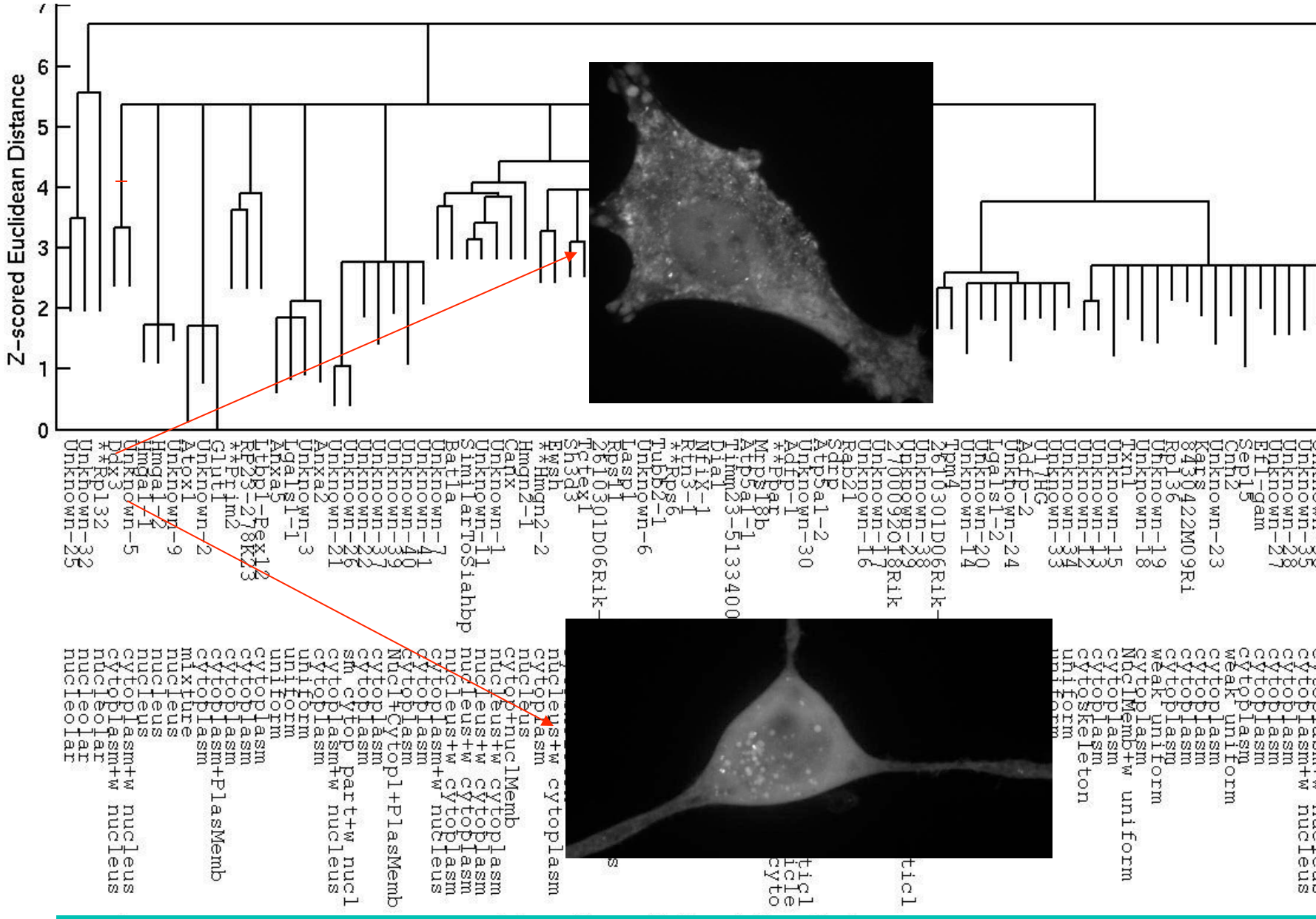
- Features can be used to measure similarity of protein patterns
- This allows us for the first time to create a systematic, objective, framework for describing subcellular locations: a **Subcellular Location Tree**
- Start by grouping two proteins whose patterns are most similar, keep adding branches for less and less similar patterns

Z-scored Euclidean Distance
0
1
2
3
4
5
6
http://murphylab.web.cmu.edu/services/PSLID/tree.html
Protein name
Human description
From databases

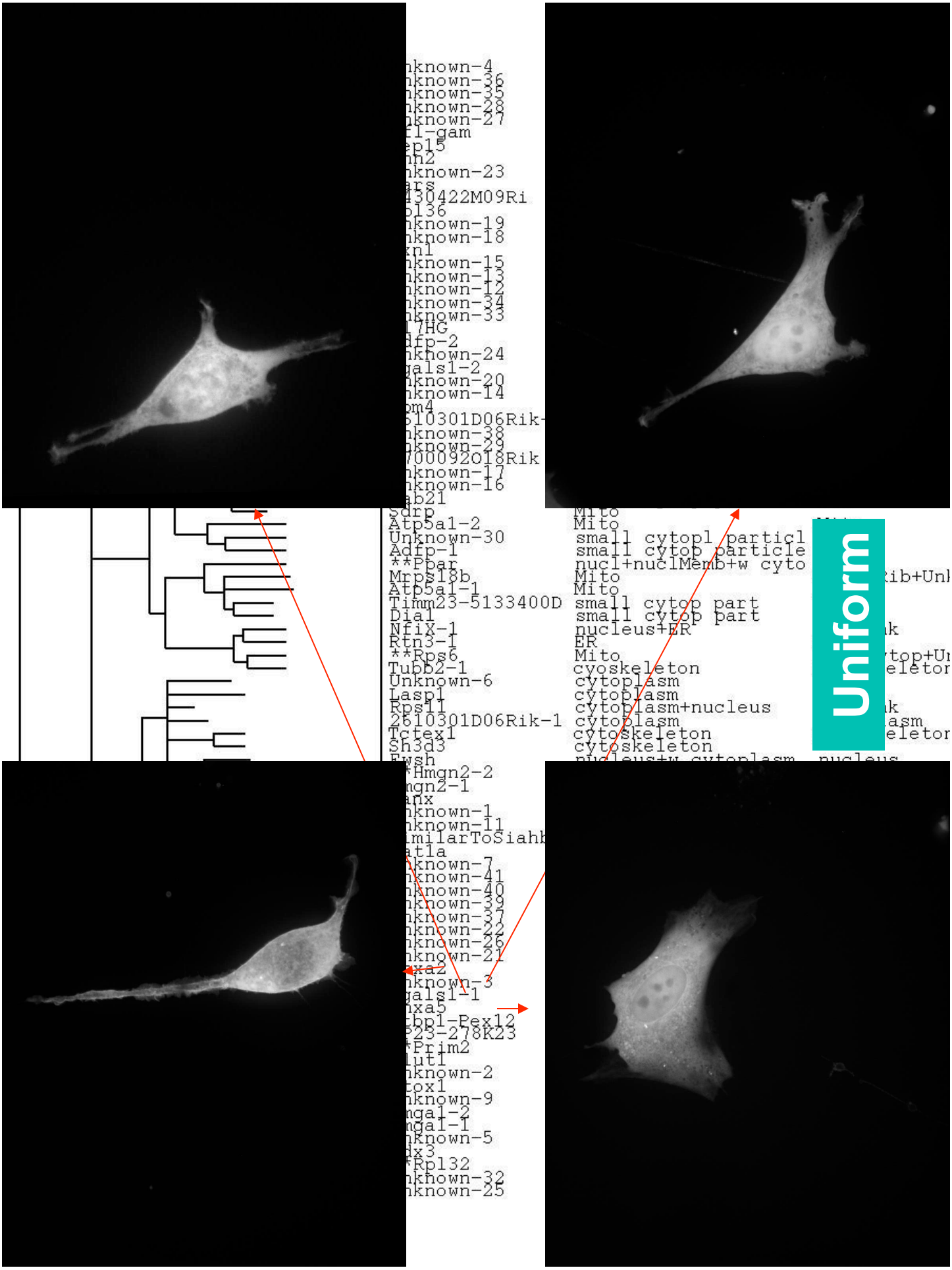
Z-scored Euclidean Distance
0
1
2
3
4
5
6
7
Nucleolar Proteins

Z-scored Euclidean Distance
0
1
2
3
4
5
6
7
Punctate Nuclear Proteins

Predominantly
Nuclear
Proteins with
Some Punctate
Cytoplasmic
Staining



Nuclear and Cytoplasmic Proteins with Some Punctate Staining


Uniform

Decomposing (unmixing) complex patterns

Decomposing mixture patterns

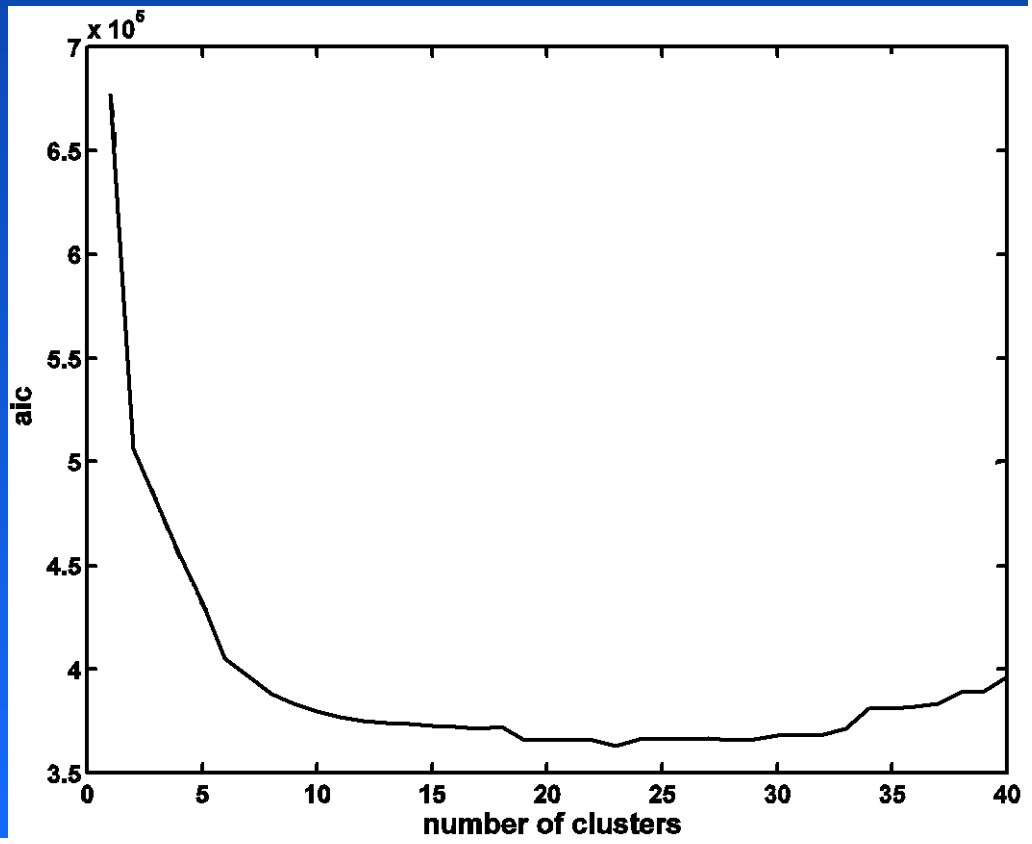
- Clustering or classifying whole cell patterns will consider each combination of two or more “basic” patterns as a unique new pattern
- Desirable to have a way to *decompose* mixtures instead
- One approach would be to assume that each basic pattern has a recognizable combination of *different types of objects*

Object type determination

- Rather than specifying object types, we can choose to learn them from the data
- Use subset of SLFs to describe objects
- Perform *k*-means clustering for *k* from 2 to 40
- Evaluate goodness of clustering using Akaike Information Criterion
- Choose *k* that gives lowest AIC

Cluster Number Selection

- Akaike Information Criterion (AIC) = 2k – 2ln(L)
- k=number of clusters
- L=likelihood of model given data



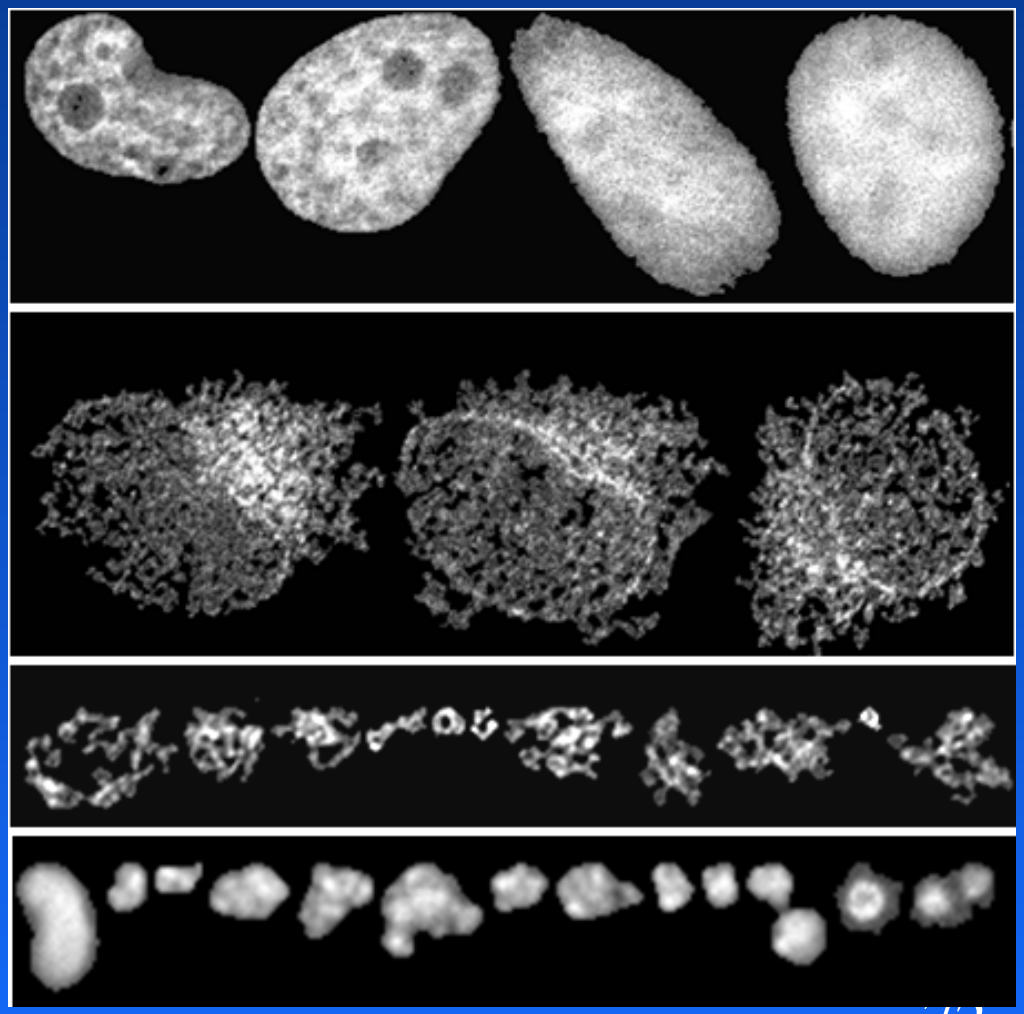
Example of Object Types

Type A

Type B

Type C

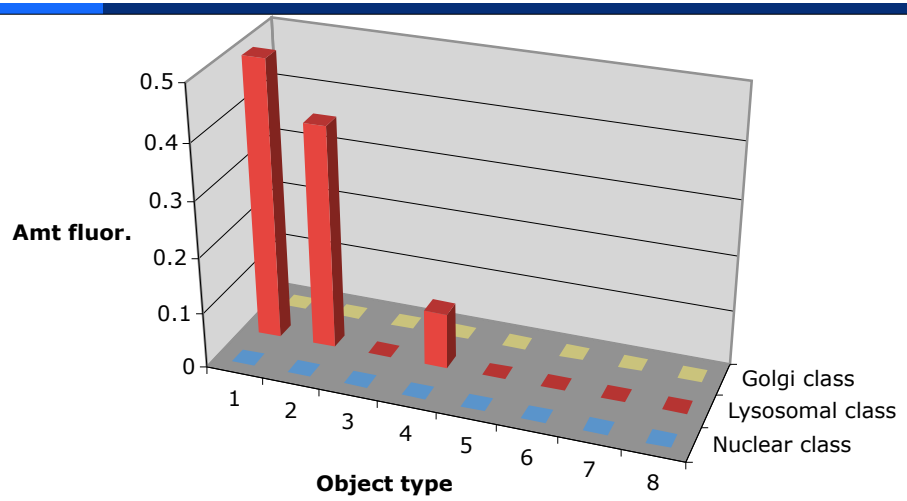
Type D



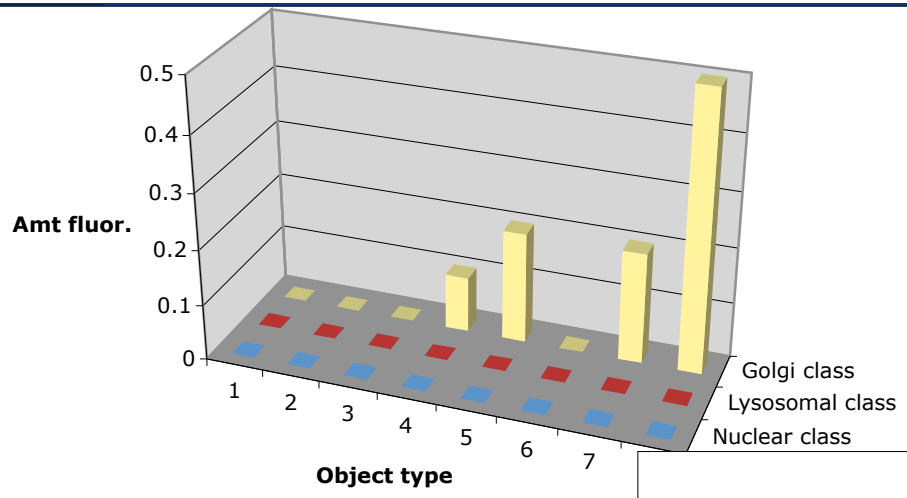
Unmixing: Learning strategy

- Once object types are known, each cell in the training (pure) set can be represented as a vector of the amount of fluorescence for each object type
- Learn probability model for these vectors for each class
- Mixed images can then be represented using mixture fractions times the probability distribution of objects for each class

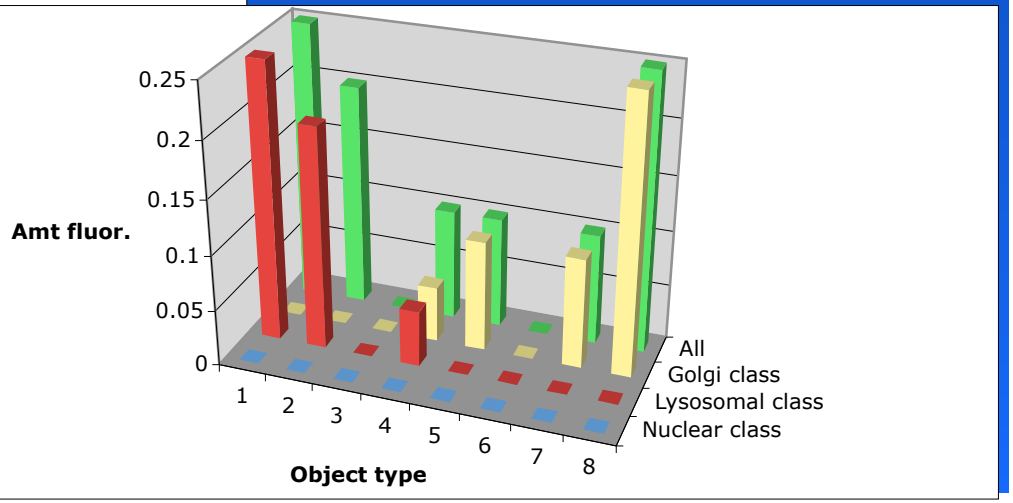
Pure Lysosomal Pattern



Pure Golgi Pattern



50% mix of each



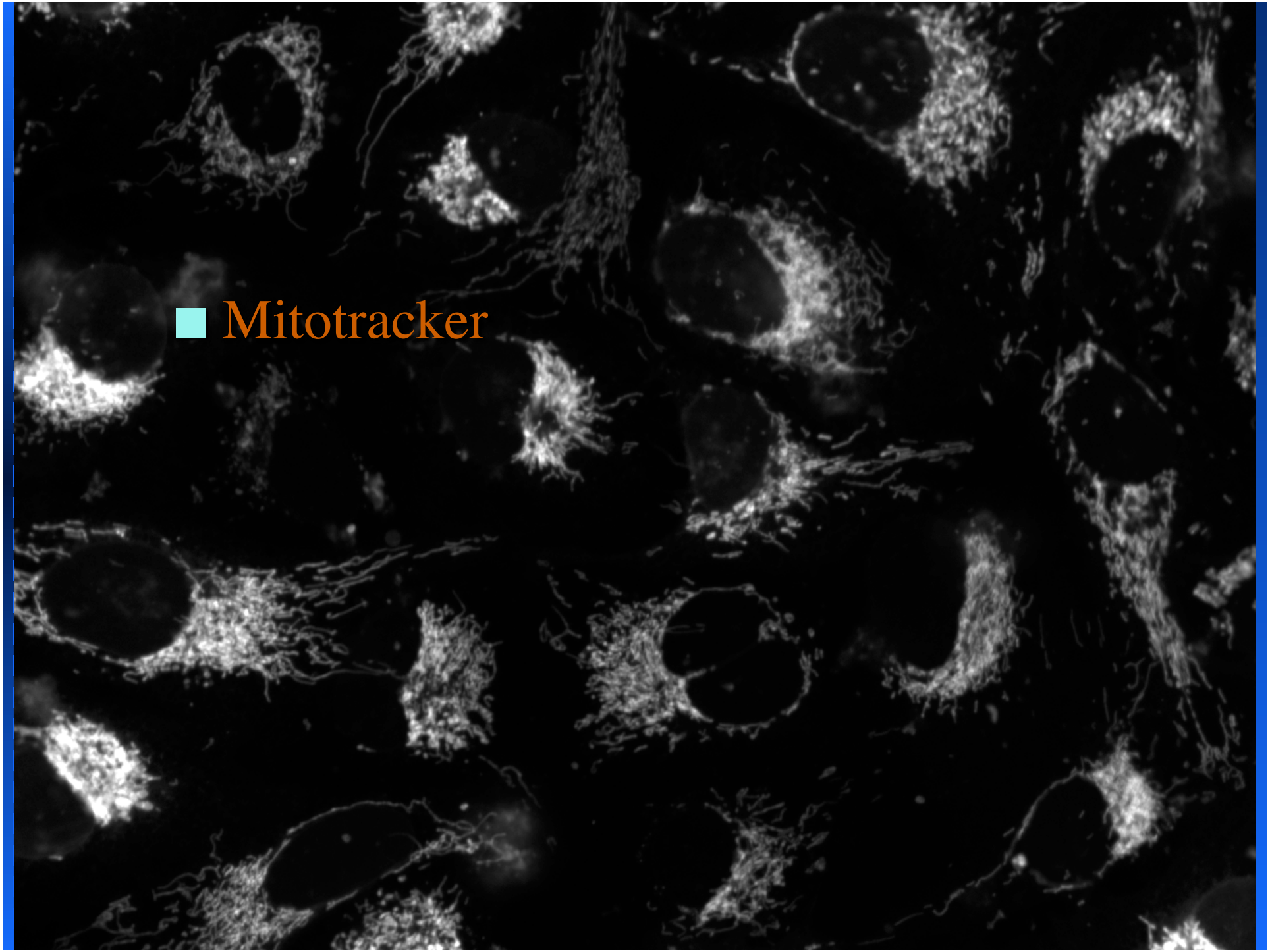
Two-stage Strategy for unmixing unknown image

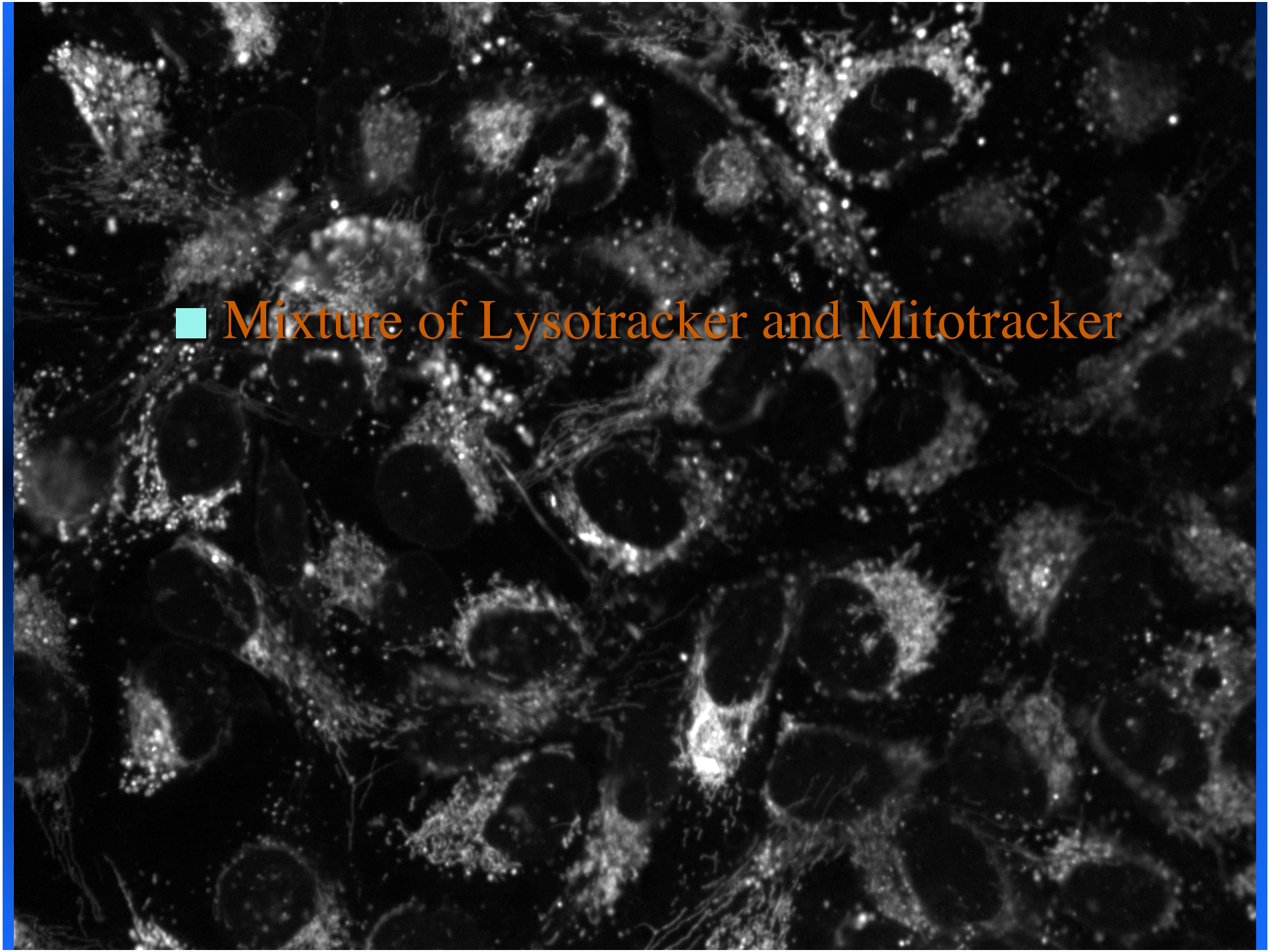
- Find objects in unknown (test) image, classify each object into one of the object types using learned object type classifier built with all objects from training images
- For each test image, make list of how often each object type is found
- Find the fractions of each class that give “best” match to this list

Test samples

- How do we test a subcellular pattern unmixing algorithm?
- Need images of known mixtures of pure patterns – difficult to obtain “naturally”
- Created test set by mixing different proportions of two probes that localize to different cell parts (lysosomes and mitochondria)

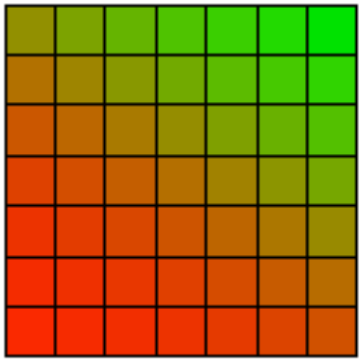

Tao Peng, Ghislain Bonamy, Estelle Glory, Sumit Chanda, Dan Rines (Genome Research Institute of Novartis Foundation)
Lysotracker


Mitotracker

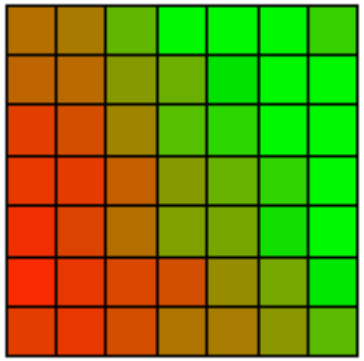

Mixture of Lysotracker and Mitotracker

Pattern unmixing results

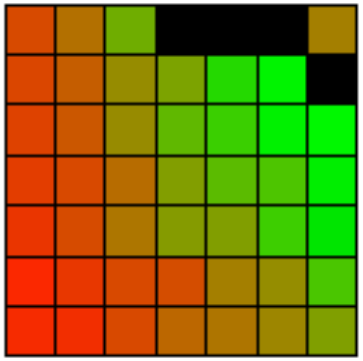
Predicted pattern fractions



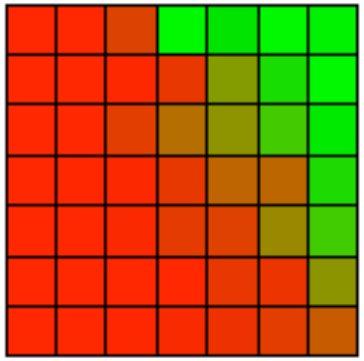
Linear unmixing



Multinomial unmixing

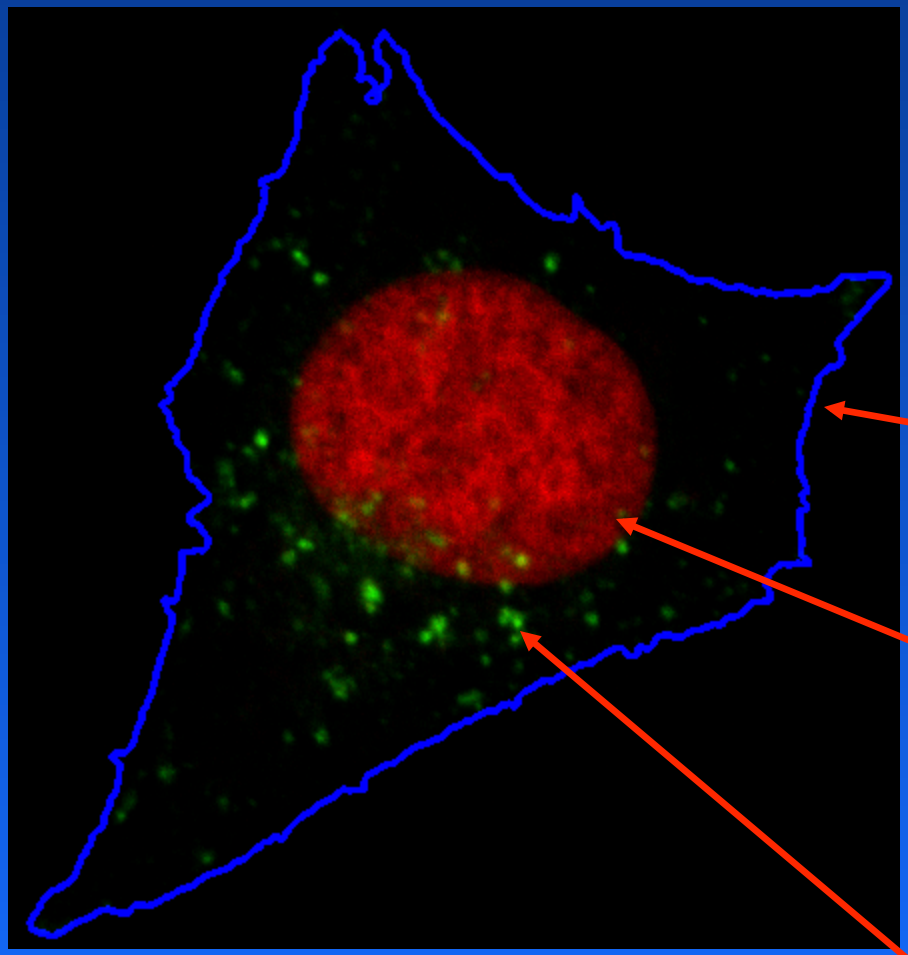


Fluorescence fraction unmixing



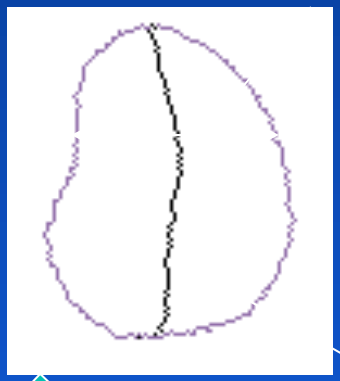
Generative models of subcellular patterns

LAMP2 pattern

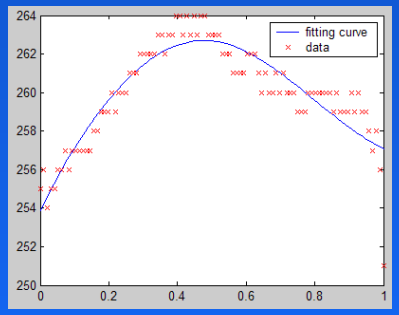


Nuclear Shape - Medial Axis Model

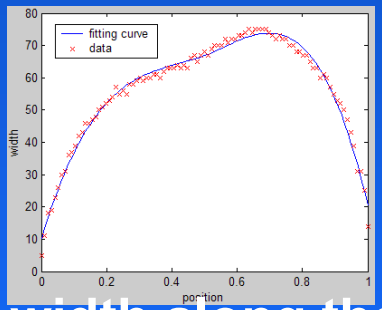
Rotate



Represented by two curves



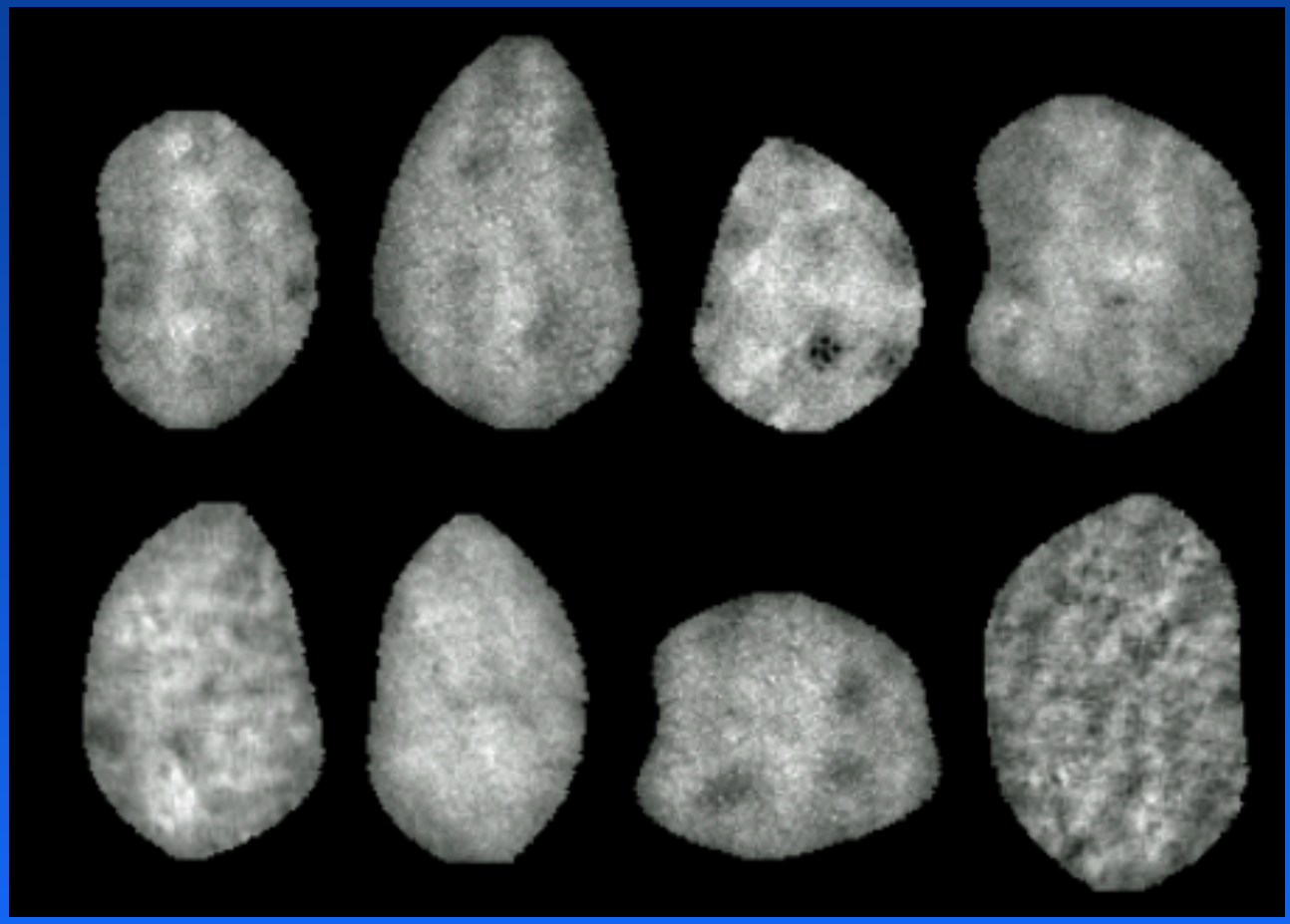
the medial axis



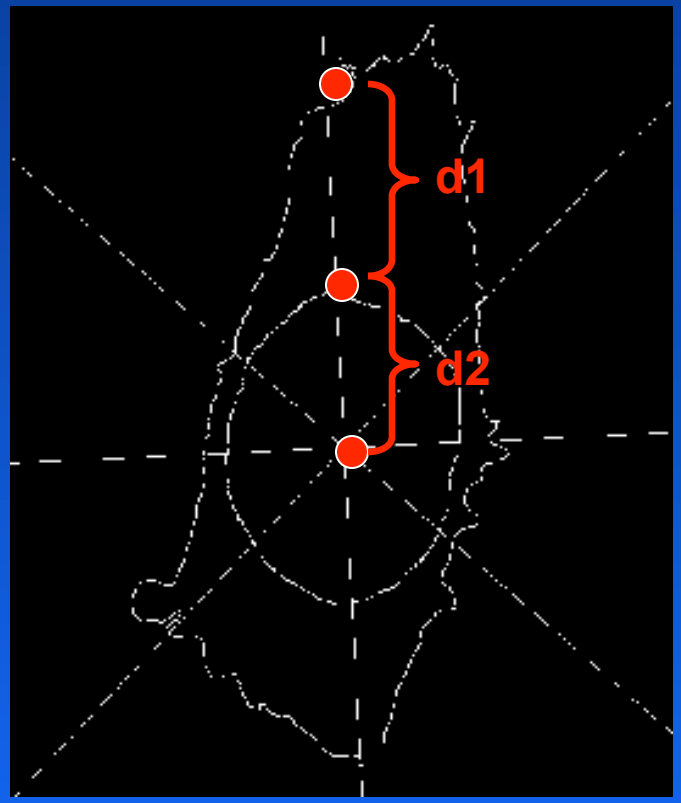
width along the medial axis

Synthetic Nuclear Shapes

With added nuclear texture



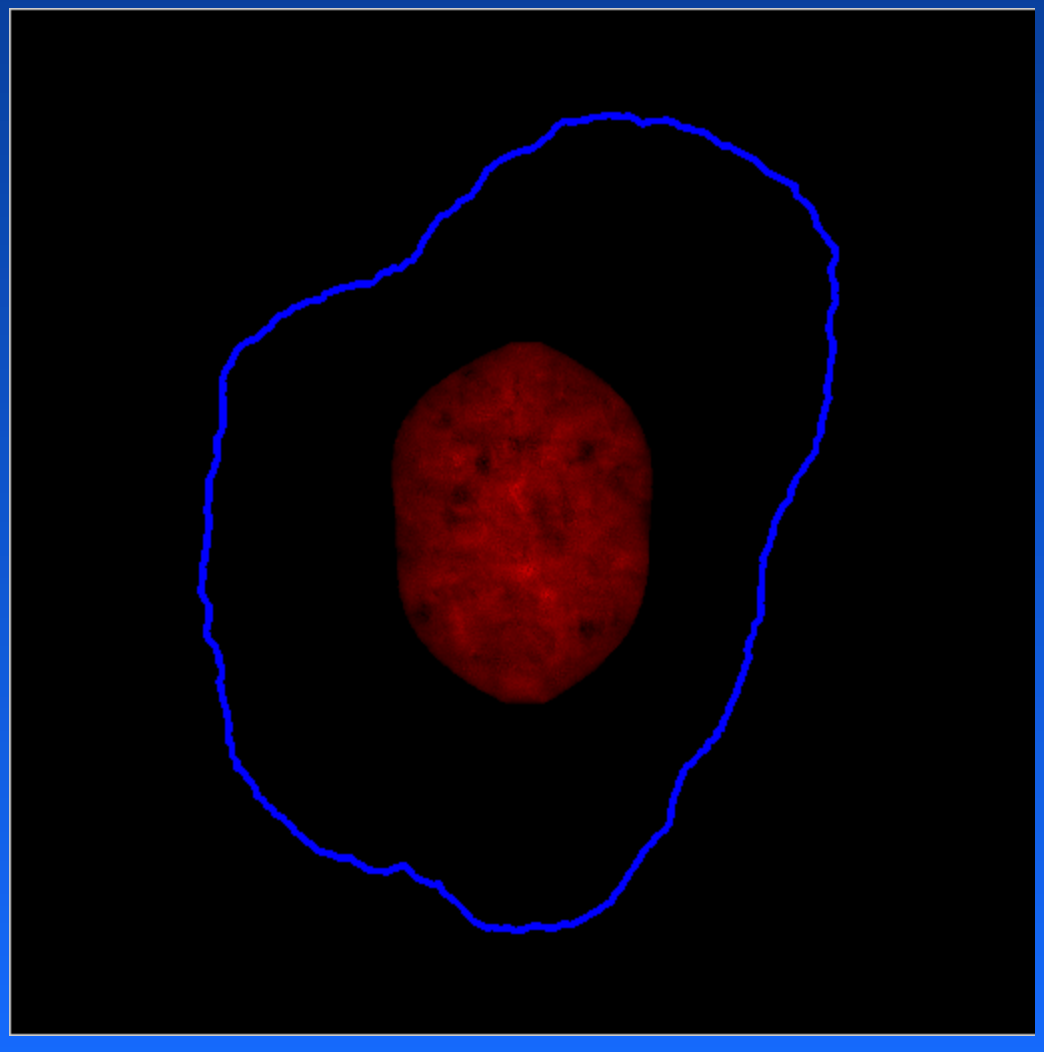
Cell Shape Description: Distance Ratio



$$r = \frac{d_1 + d_2}{d_2}$$

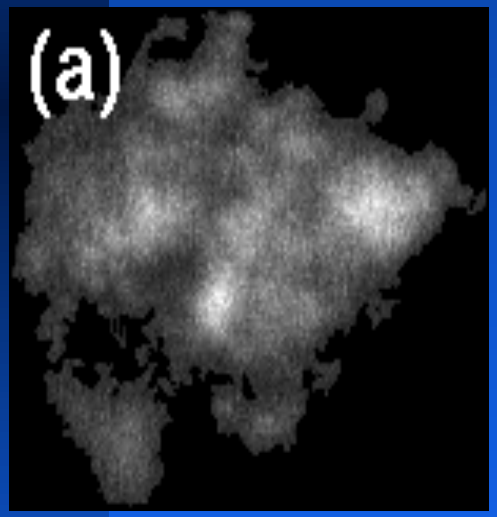
Capture variation as a principal components model

Generation

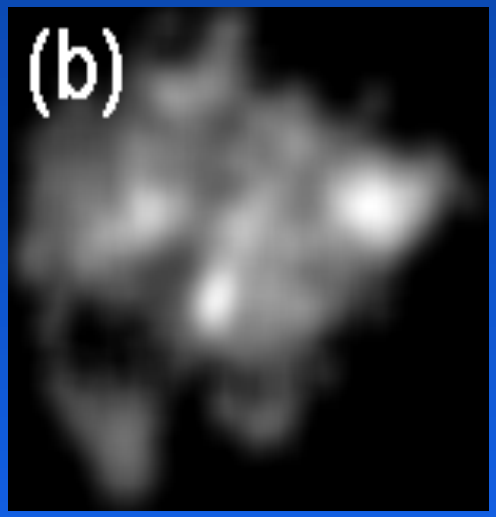


Modeling Vesicular Organelles

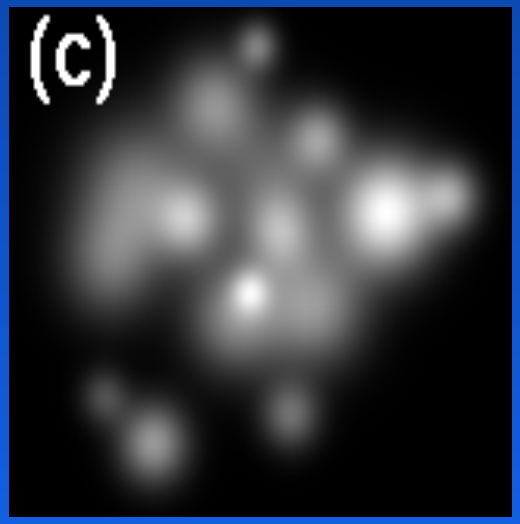
Original



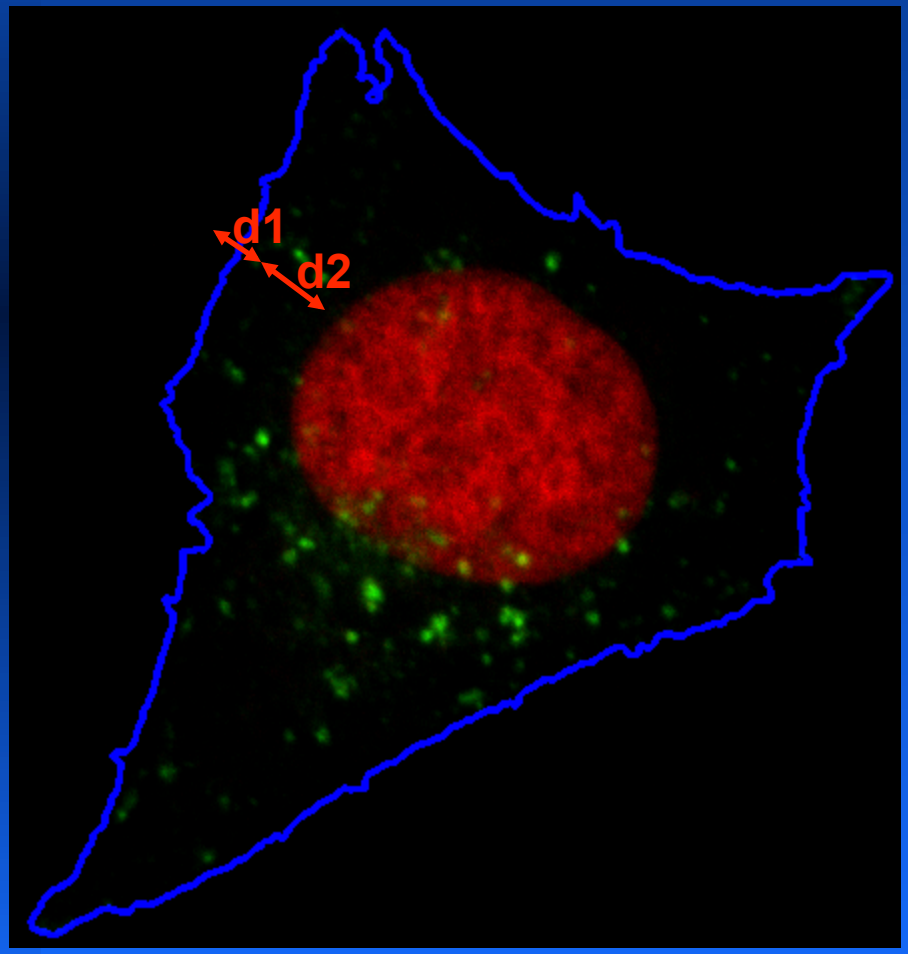
Filtered



Fitted Gaussians

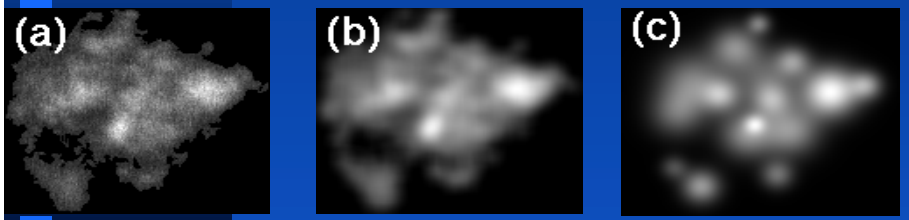


Object Positions

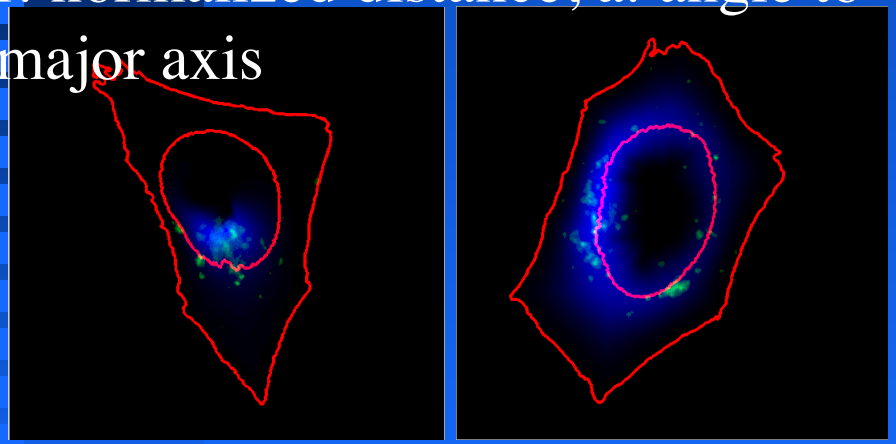


$$r = \frac{d_2}{d_1 + d_2}$$

Models for protein-containing objects

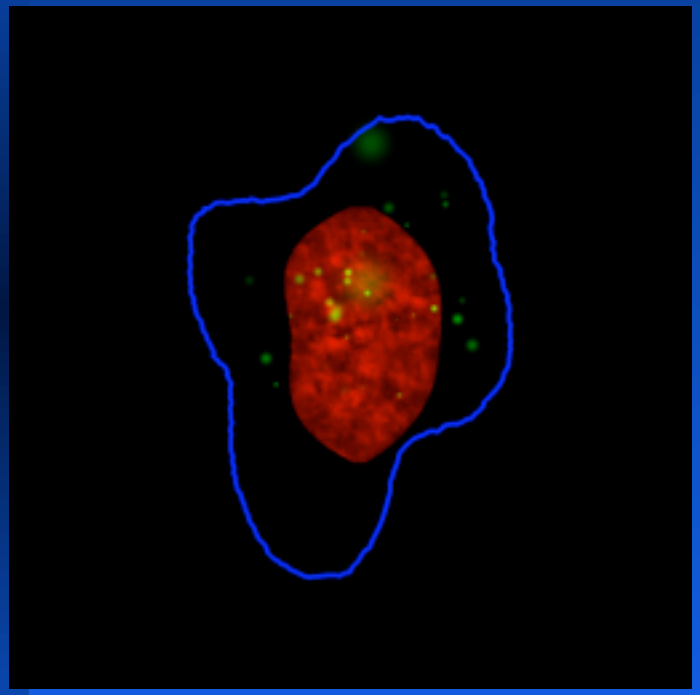


r: normalized distance, a: angle to major axis

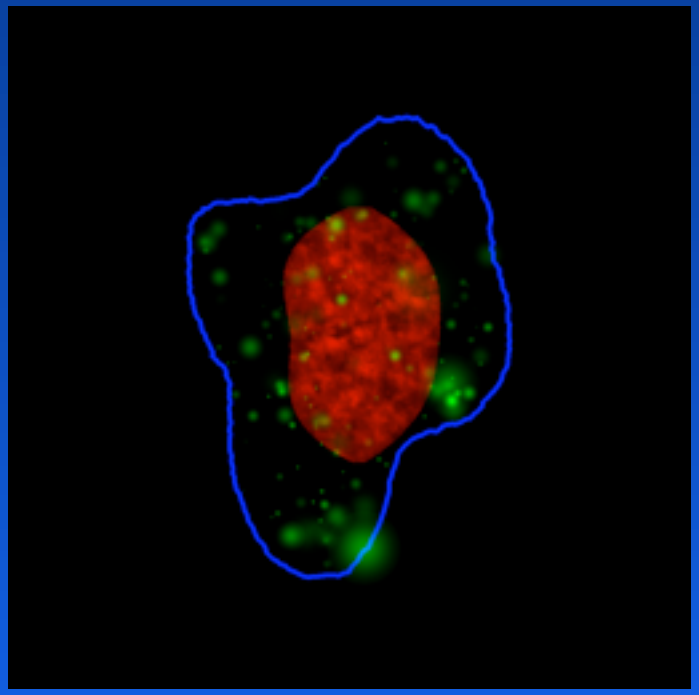


- Mixture of Gaussian objects
- Learn distributions for number of objects and object size
- Learn probability density function for objects relative to nucleus and cell

Synthesized Images



Lysosomes



Endosome

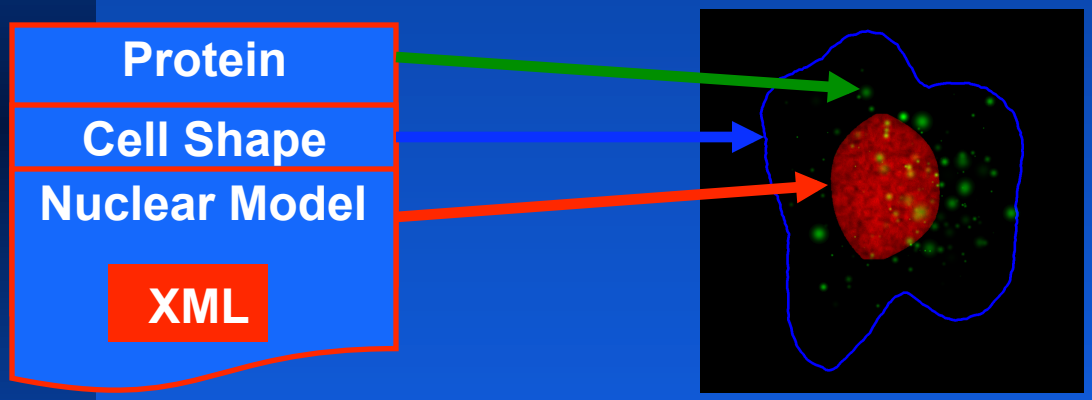
SLML toolbox - Ivan Cao-Berg, Tao Peng, Ting Zhao

- Have portable tool for generating images from a model

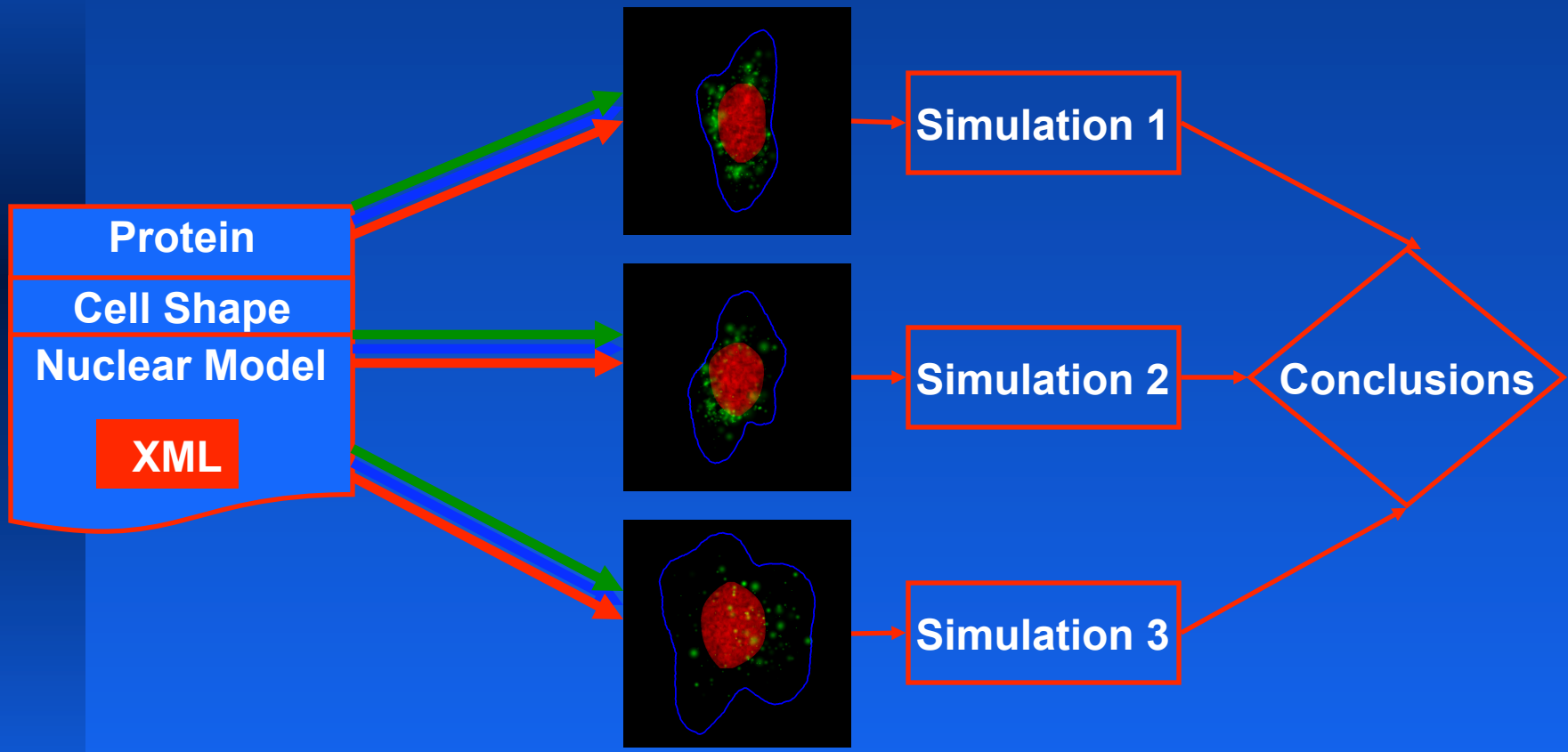
Model Distribution

- Generative models provide better way of distributing what is known about “subcellular location families” (or other imaging results, such as illustrating change due to drug addition)
- Have initial XML design for capturing the models for distribution
- Have portable tool for generating images from the model

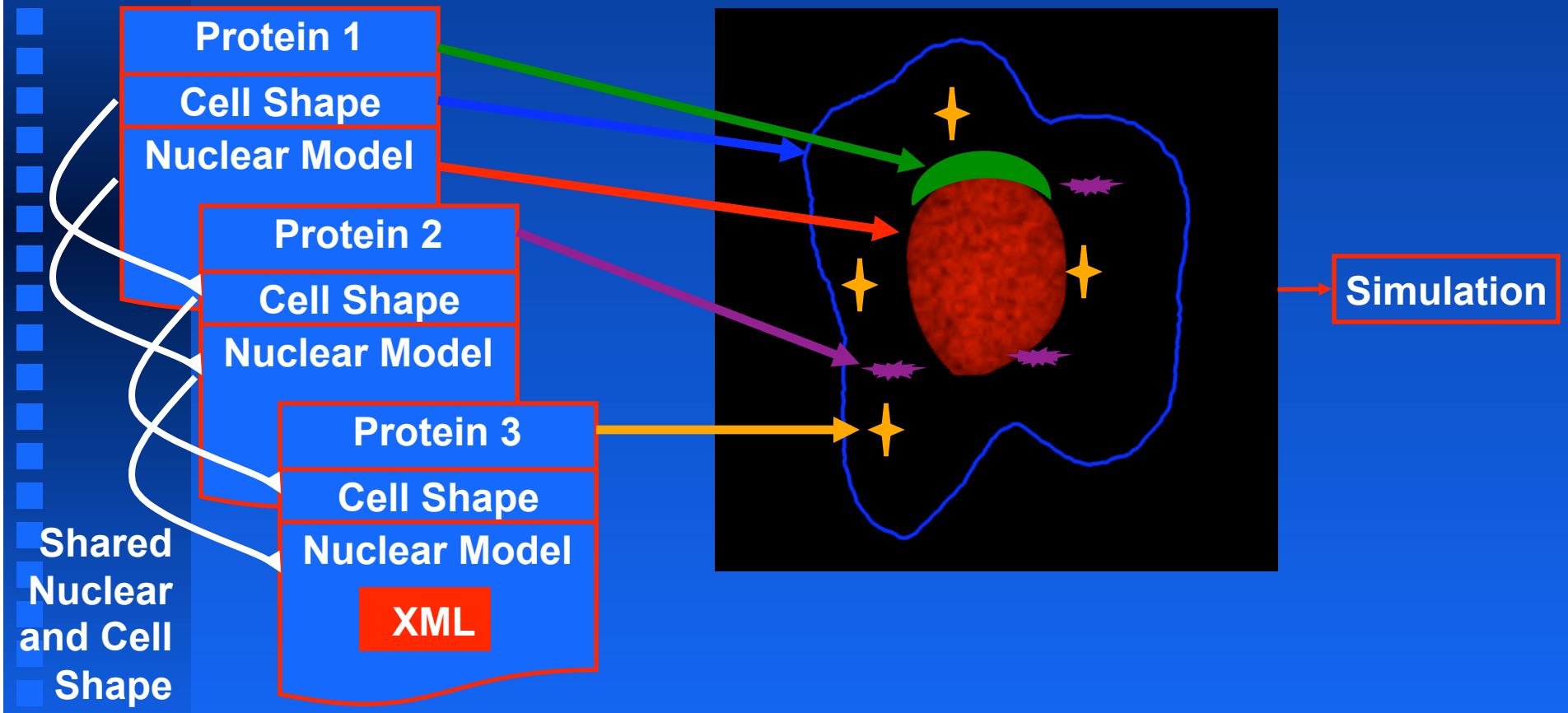
Generation Process



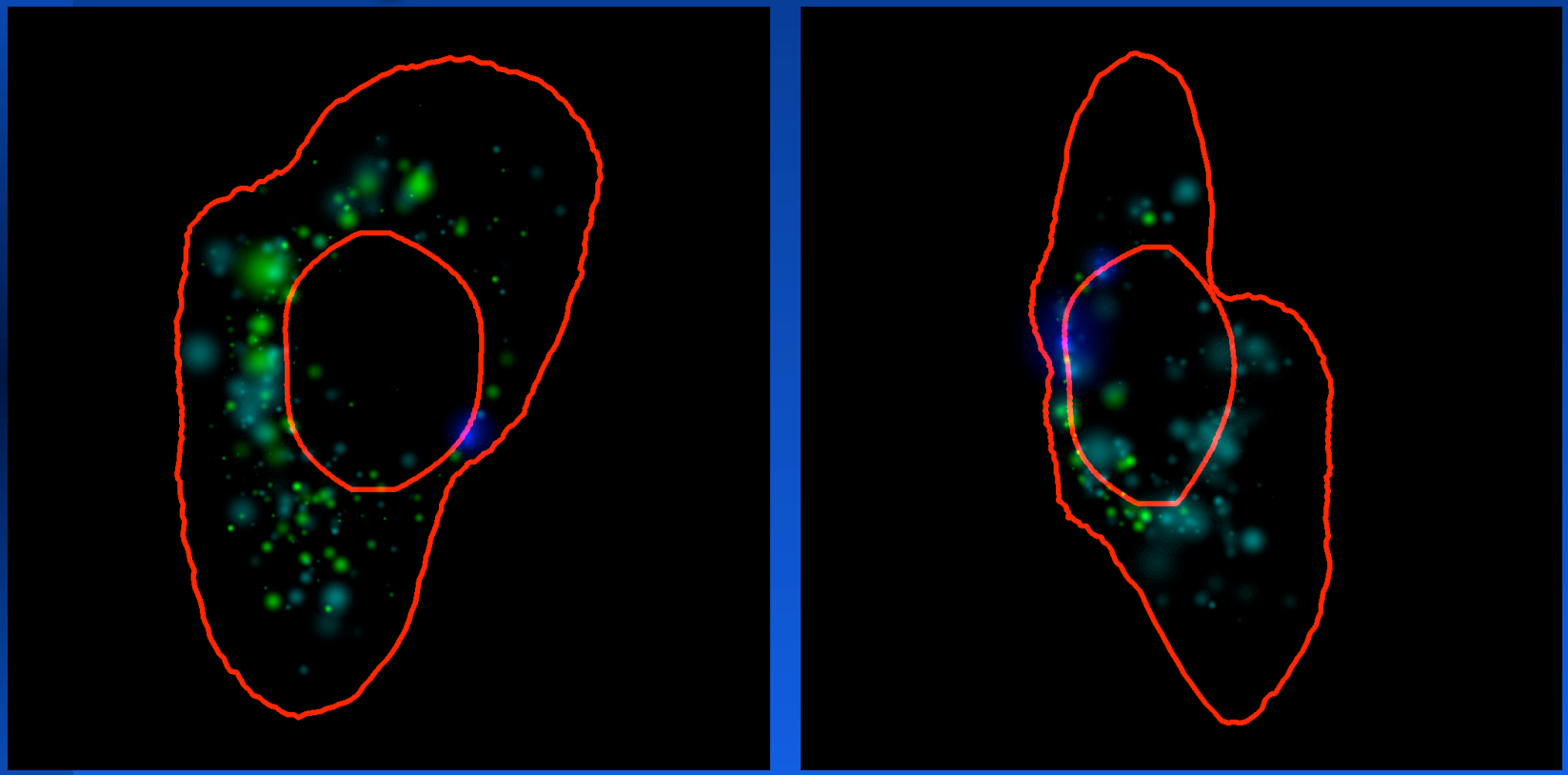
Generating Multiple Distributions for Simulations



Combining Models for Cell Simulations



Example combination



Red = nuclear membrane, plasma membrane
Blue = Golgi
Green = Lysosomes
Cyan = Endosomes

Conclusions

- Computers better than people at recognizing complex subcellular patterns
- Automated analysis of subcellular patterns in tissues demonstrated – useful for potential biomarker discovery
- Complex patterns can be unmixed – useful for monitoring transitions between patterns (e.g., translocations)
- Generative models can be built directly or indirectly from data to summarize results and make predictions – useful for cell simulations

Software availability

- http://murphylab.web.cmu.edu/software
- http://www.cellprofiler.org
- http://www.openmicroscopy.org
- http://www.cbi-tmhs.org/Dcelliq/