Automated Proteome-wide Determination and Modeling of Subcellular Location

Tutorial at Computational Methods for Systems Biology

October 13, 2008

Robert F Murphy

Ray & Stephanie Lane Professor of Computational Biology

Departments of Biological Sciences, Biomedical Engineering and

Machine Learning and

Center for Bioimage Informatics

from image to knowledge

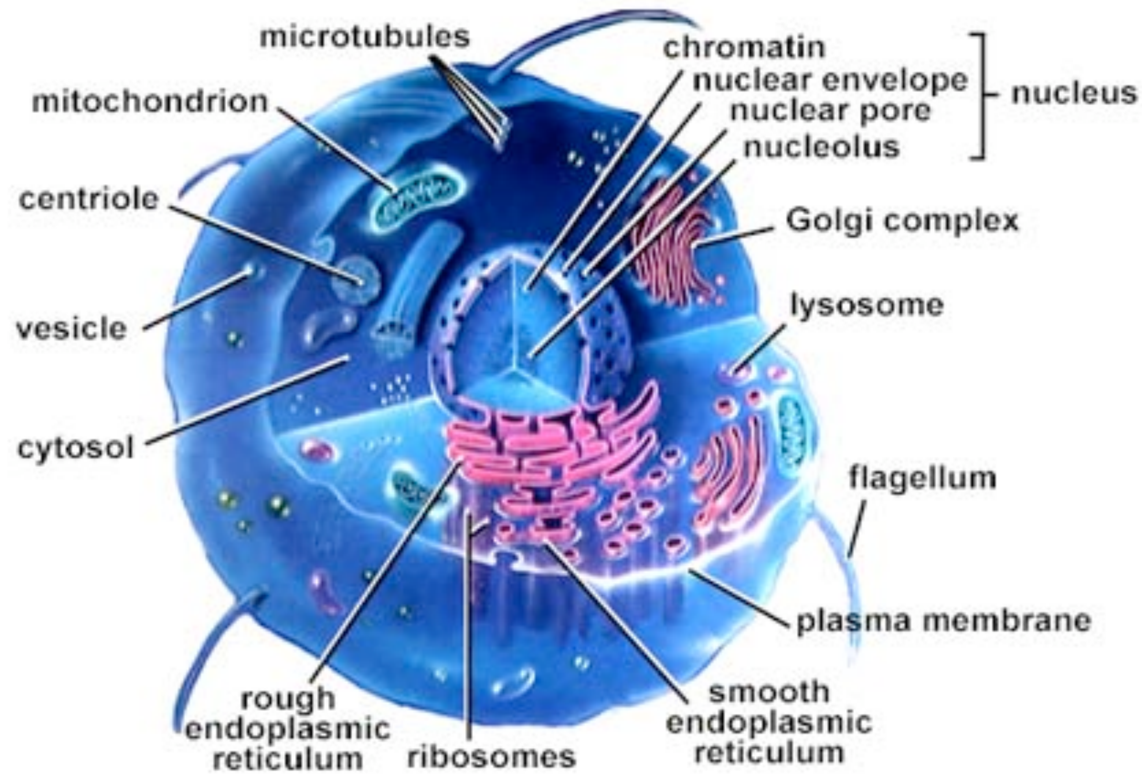
RAY AND STEPHANIE LANE

Center for Computational Biology

Carnegie Mellon

Subcellular Location

- Proteins can be found in many places within cells
- Task is to learn what places (patterns) are possible and which proteins are in which



Approaches to subcellular location

- Prediction
- Determination
  - Deterministic
    - Classification
    - Clustering
  - Probabilistic
    - Pattern unmixing
    - Generative models

Tagging methods

- Antibodies
  - Monoclonal
  - Monospecific polyclonal
    - Human Protein Atlas project
- cDNA tagging
  - UCSF yeast project
  - Heidelberg project
  - GNF project
- Genomic tagging
  - Randtag project

Microscopy

- Manual
  - UCSF project (widefield)
  - HPA cell line project (confocal)
  - Randtag project
- Automated
  - Widefield low magnification
    - Up to 40x air, most commercial systems
    - HPA main project
    - Randtag project
  - Widefield high magnification
    - Allow water or oil objectives
    - Heidelberg project
  - Confocal
    - GNF project
- Variations: SPIM, FCM

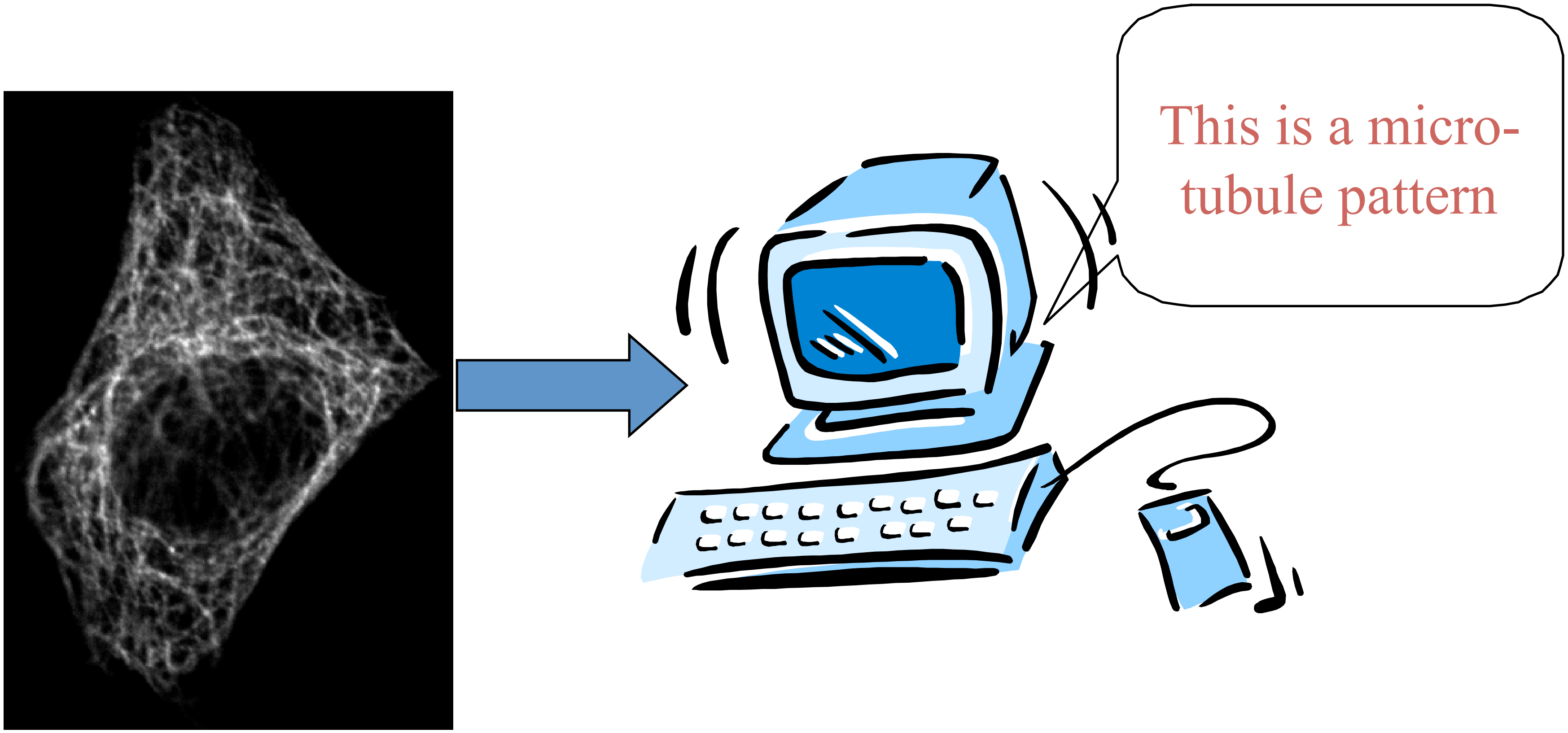
Acquisition Protocol

- What samples should be imaged?
  - Tissues or cell lines
  - Live or fixed
- What should be recorded?
  - Magnification
  - 2D or 3D
    - Z spacing?
  - Time series
  - Number of images

Acquisition Protocol

- What images should be collected beyond that of the tagged protein?
  - Transmitted light (Phase, DIC)
  - Markers
    - nucleus/DNA
    - plasma membrane
    - total protein
    - microtubules
    - ER
    - Small molecule probes (lysotracker…)

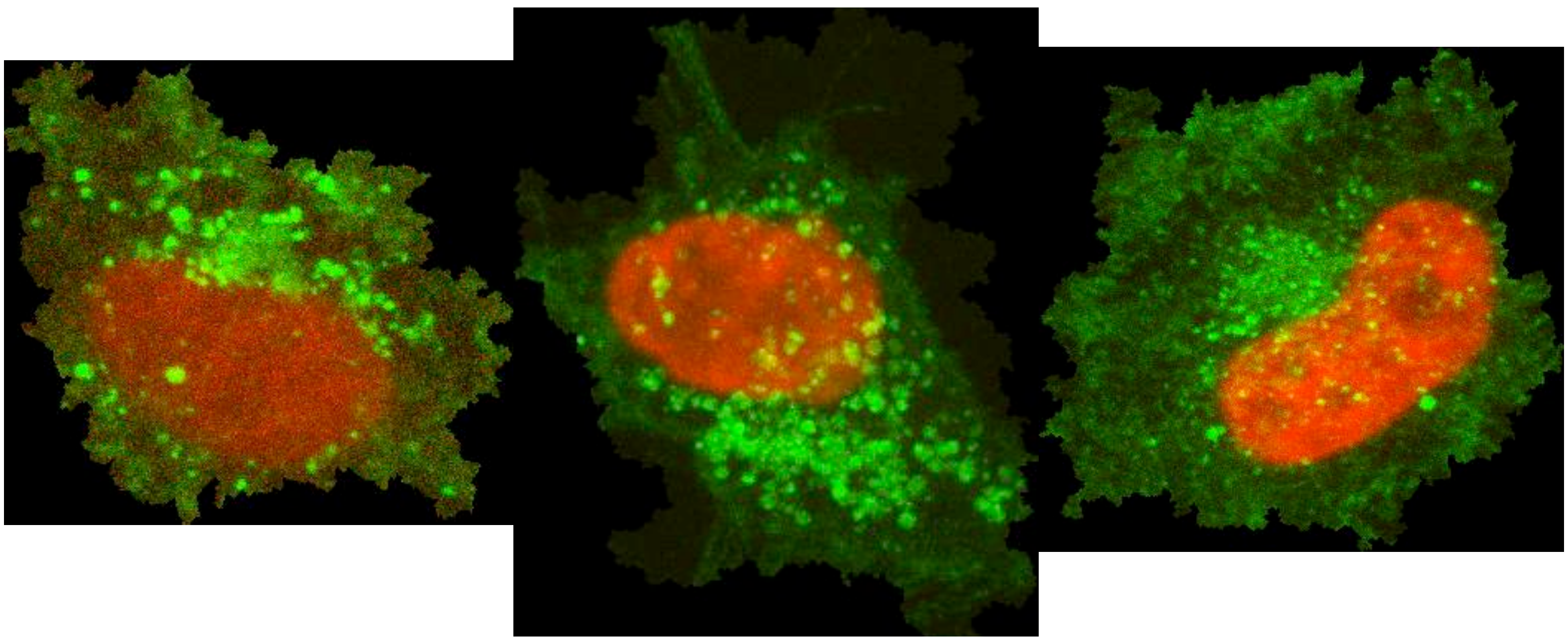
Classification



Assign proteins to *major* subcellular structures

Automated Analysis of Subcellular Location

- Problem is hard because different cells have different **shapes, sizes, orientations**
- Organelles **not found in fixed locations**
- Use **numerical features** to describe patterns



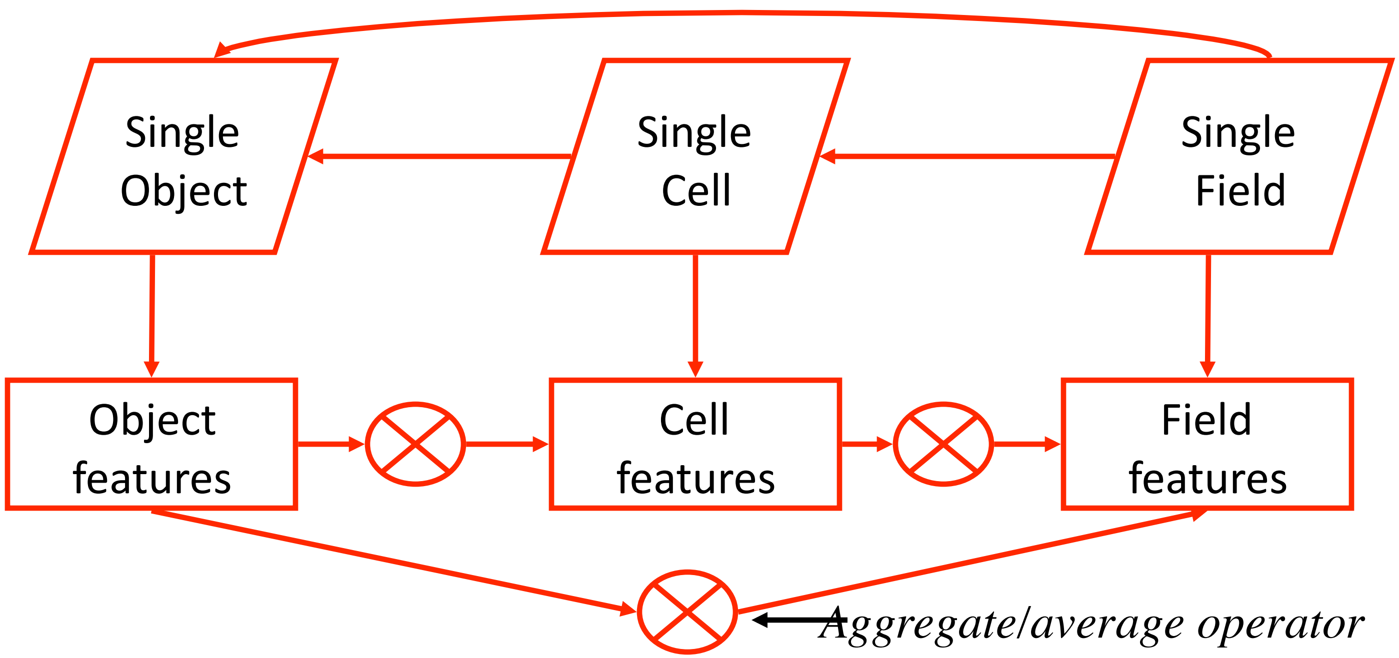
Feature-Based, Supervised Learning Approach

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 statistical **classification methods** to "learn" how to distinguish each class using the features

Preprocessing

- Correction for/Removal of camera defects
- Background correction
- Autofluorescence correction
- Illumination correction
- Deconvolution
- Registration
  - Not critical if only using DNA or membrane references
- Intensity scaling (constant scale or contrast stretched for each cell)

Feature levels and granularity



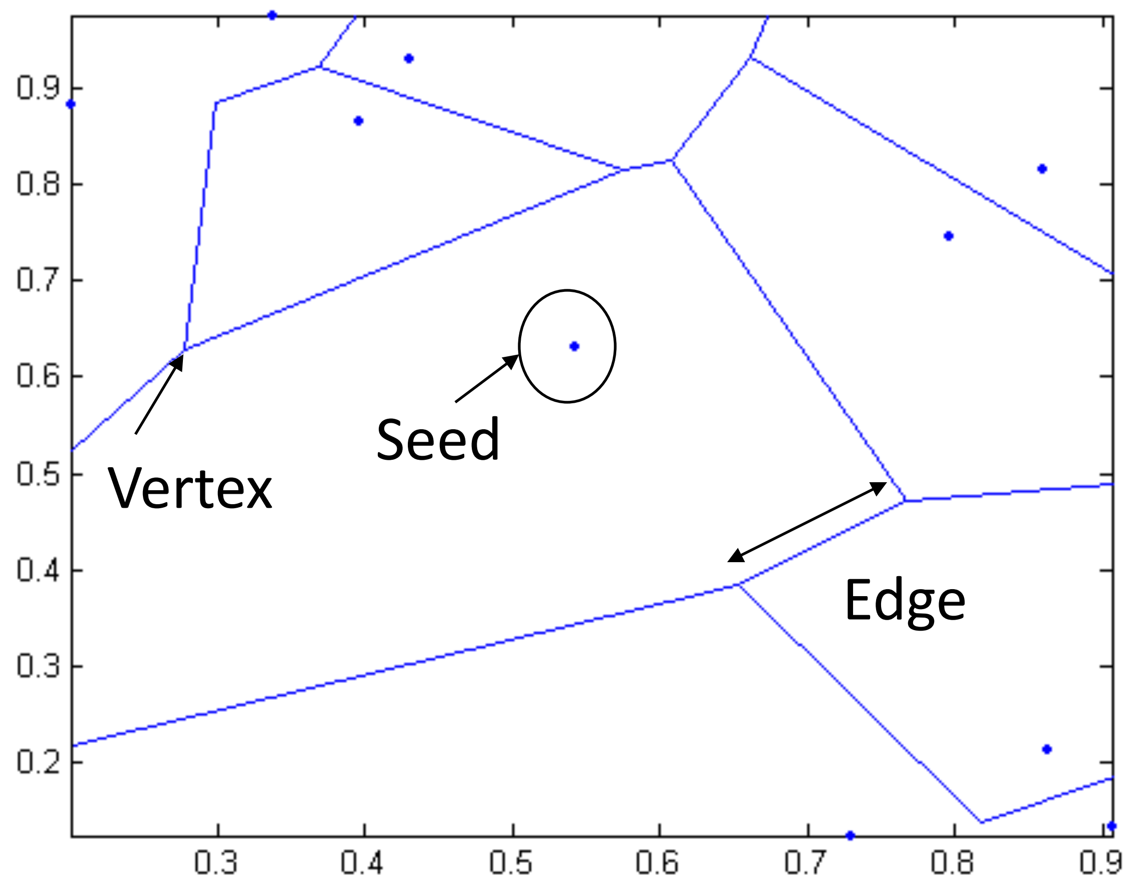
Granularity: 2D, 3D, 2Dt, 3Dt

Segmentation of Images into Single Cell Regions

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

Voronoi diagram

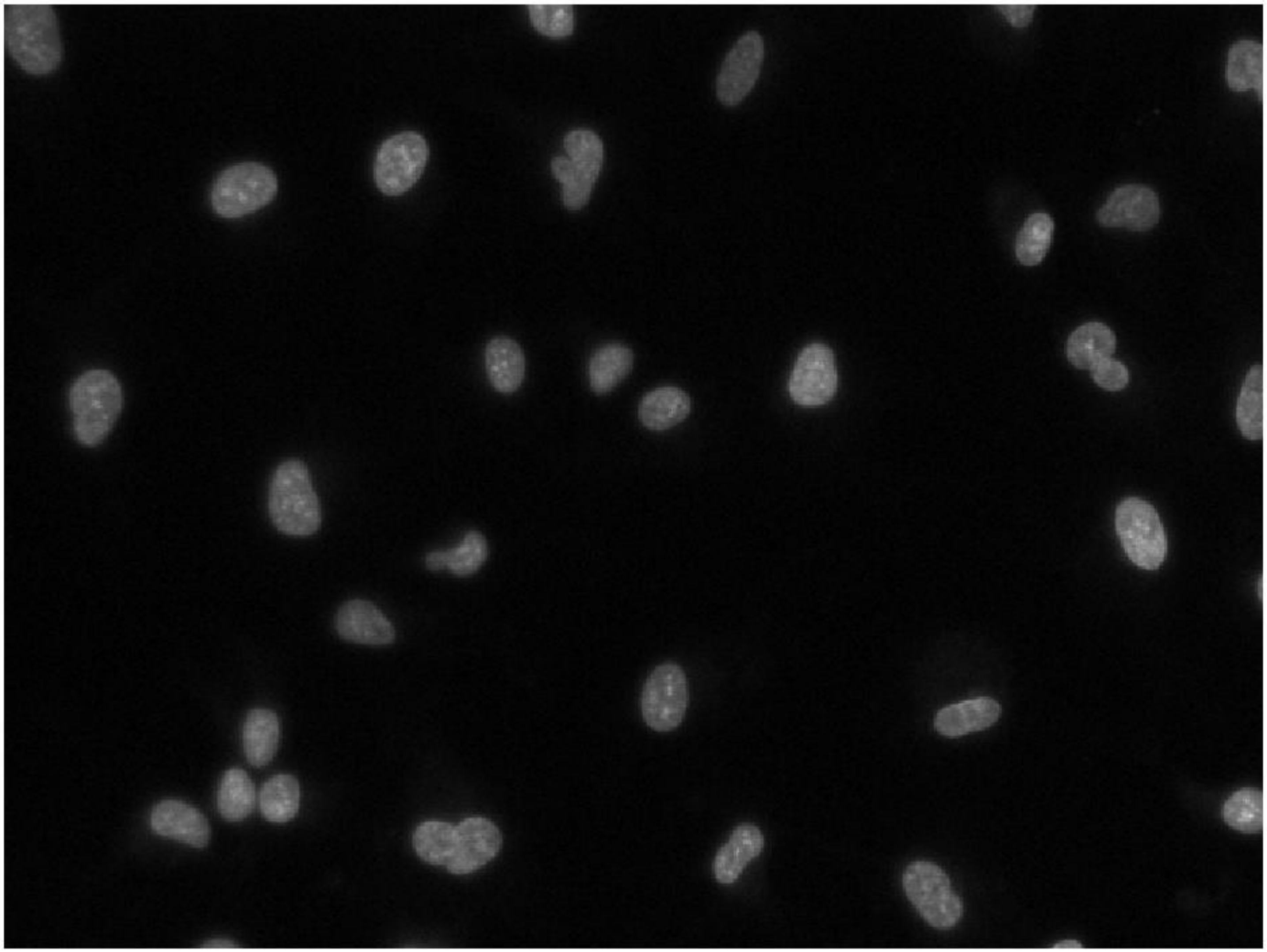
Given a set of seeds, draw vertices and edges such that each seed is enclosed in a single polygon where each edge is equidistant from the seeds on either side.



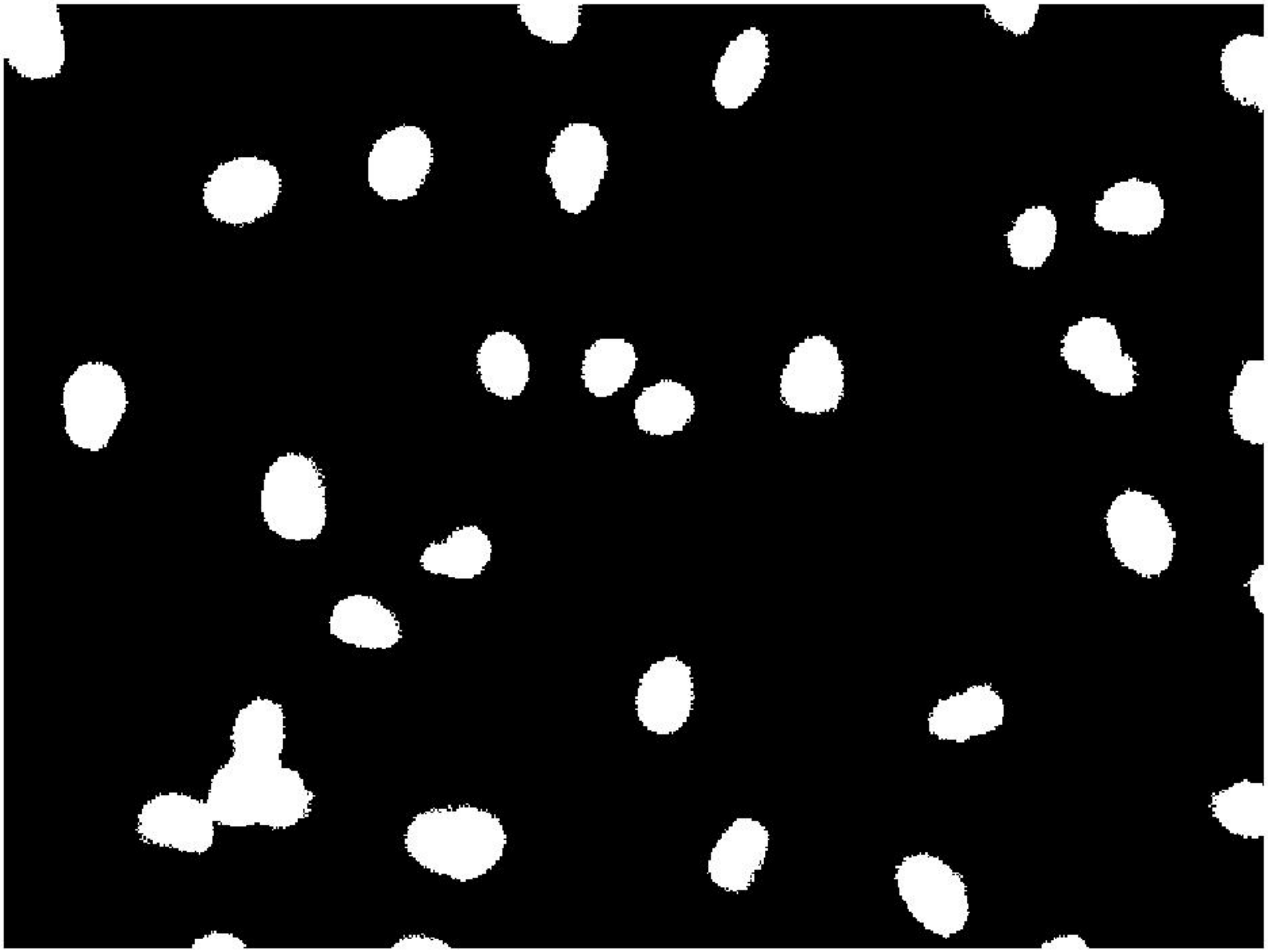
Voronoi Segmentation Process

- 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

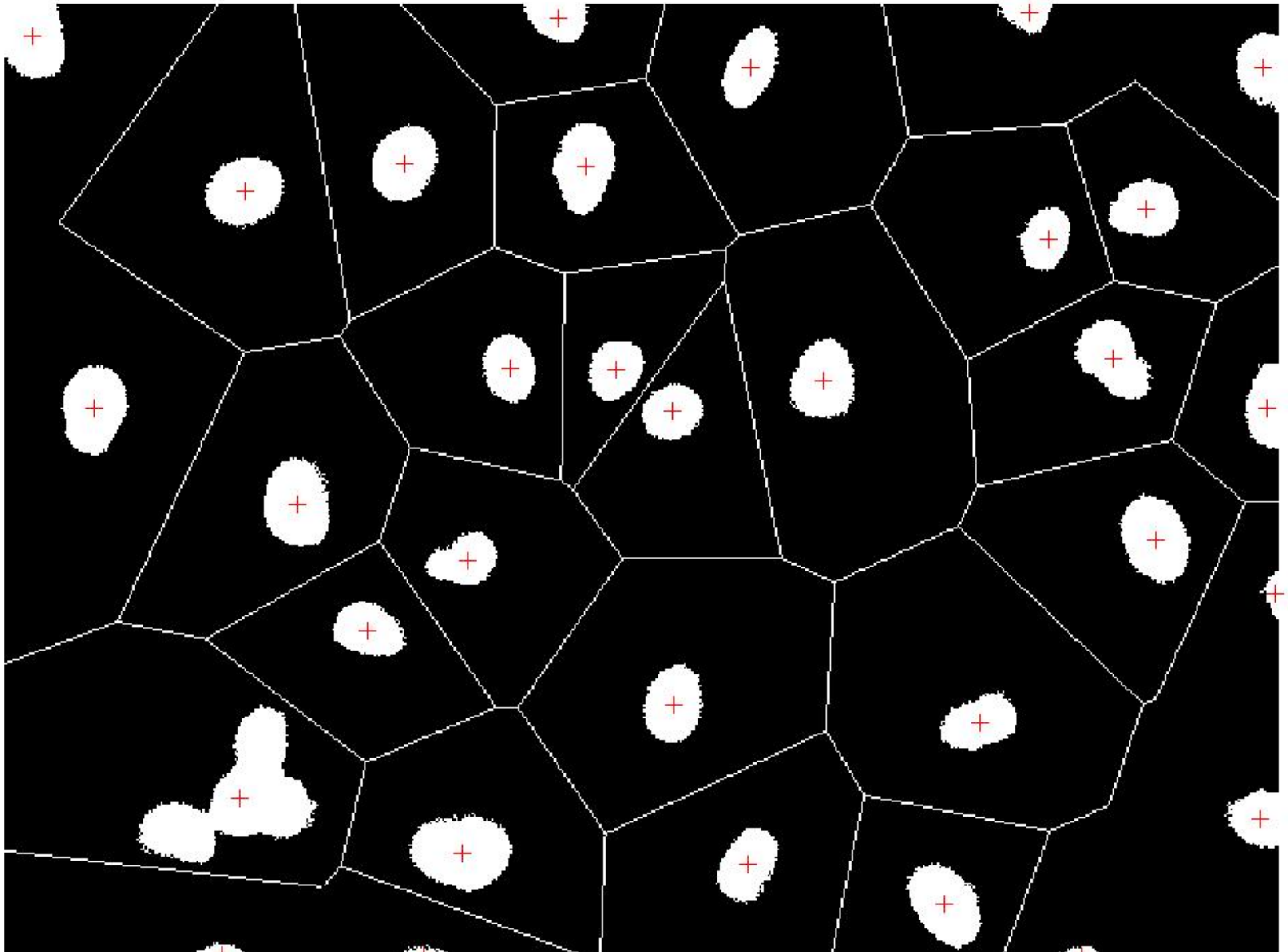
Original DNA image



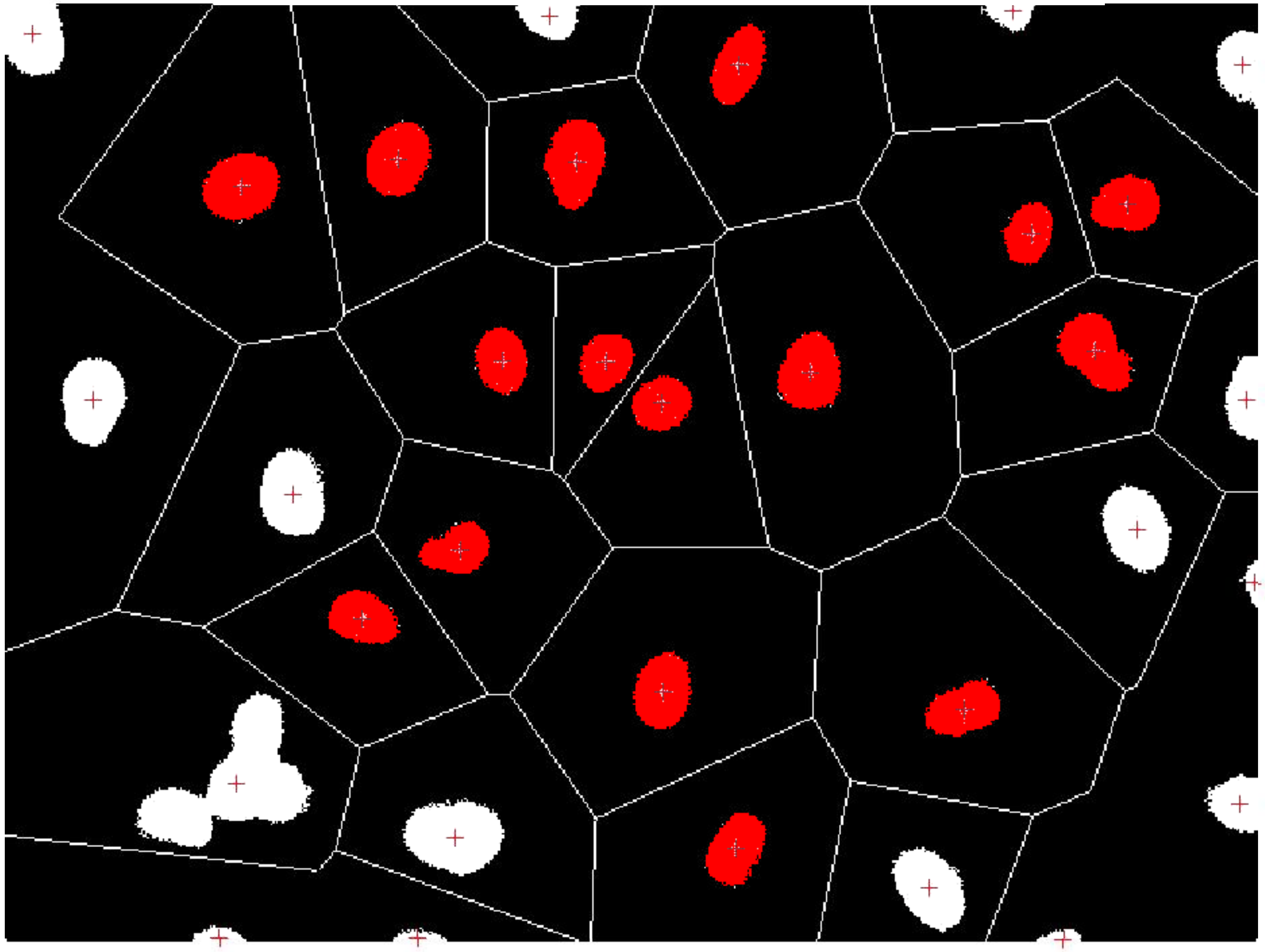
After thresholding and removing small objects



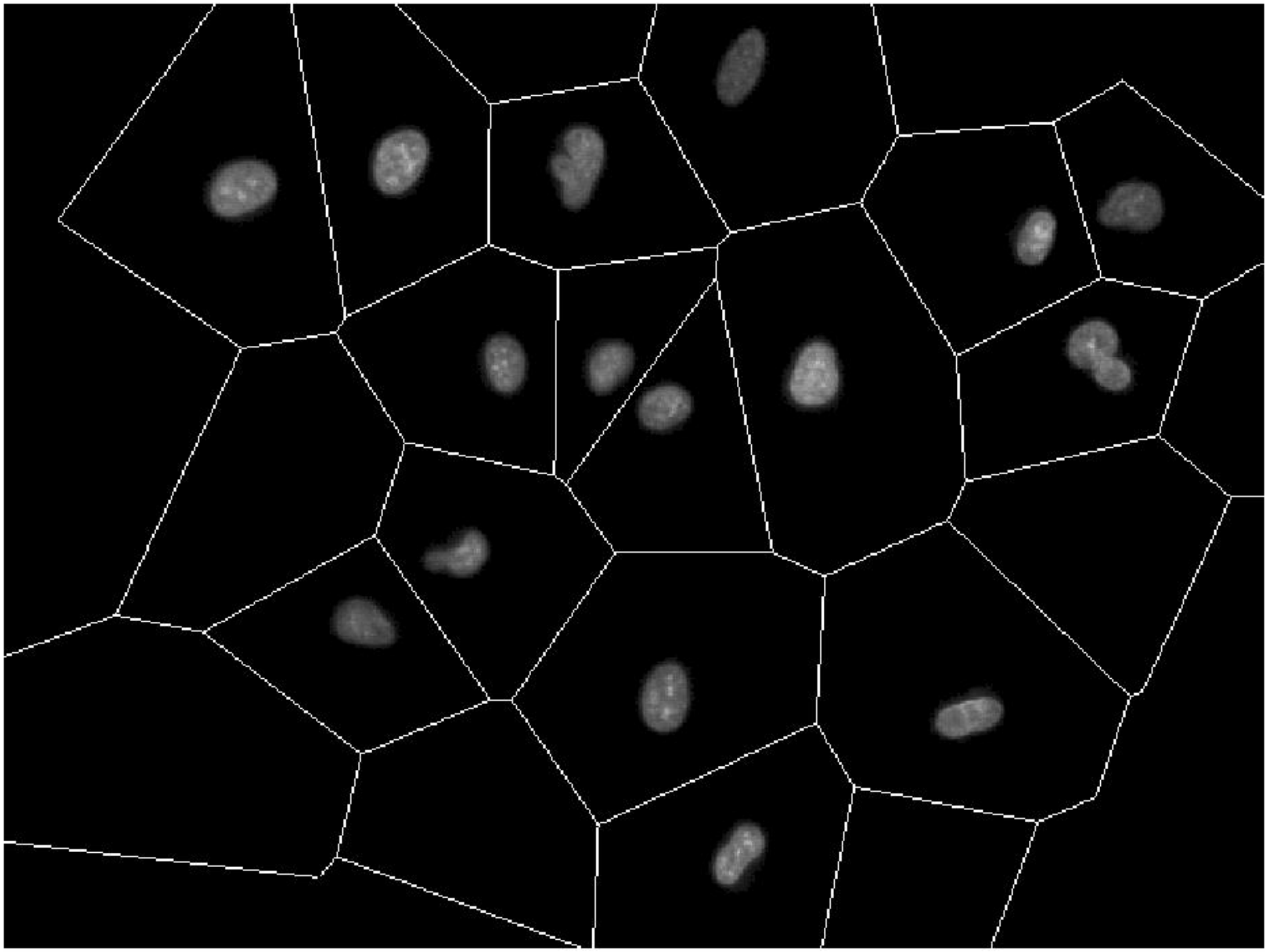
After triangulation



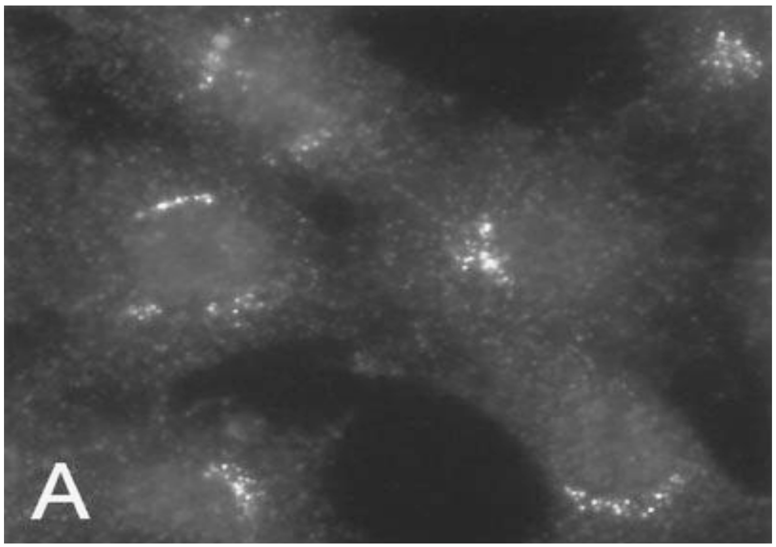
After removing edge cells and filtering



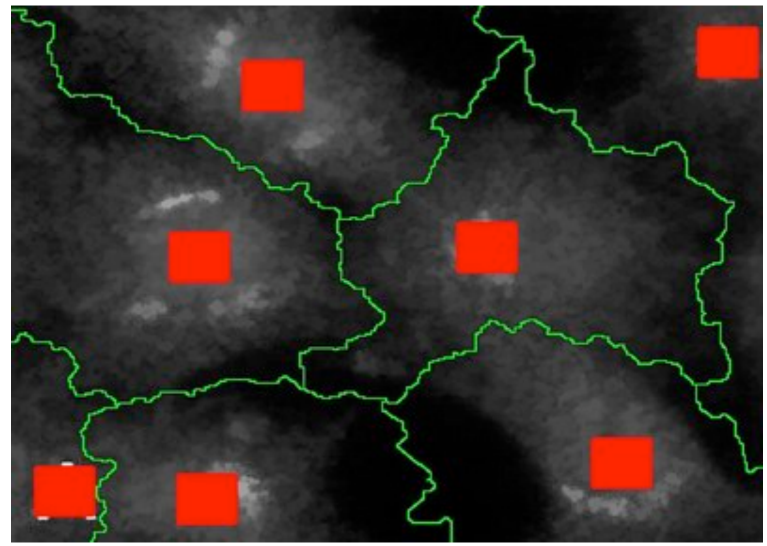
Final regions masked onto original image



Seeded Watershed Segmentation



Original image



Seeds and boundary

Applied directly to protein image (no DNA image)

Note non-linear boundaries

Subcellular Location Features (SLFs)

- Morphological
  - Protein only
  - Relative to reference
- Edge
- Zernike moment
- Texture

Thresholding

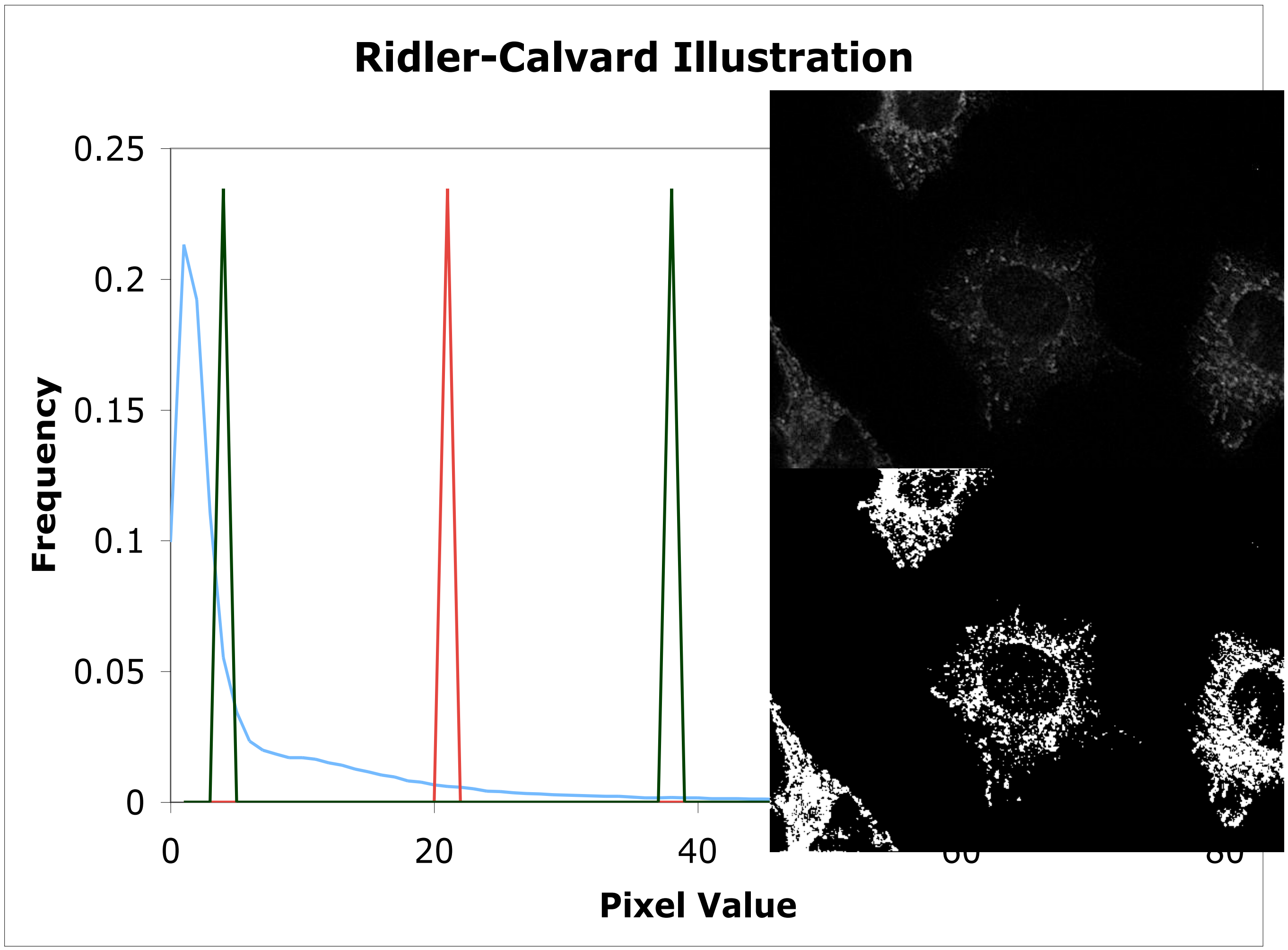
- Morphological features require some method for defining objects
- Most common approach is global thresholding
- Methods exist for automatically choosing a global threshold (e.g., Riddler-Calvard method)

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.

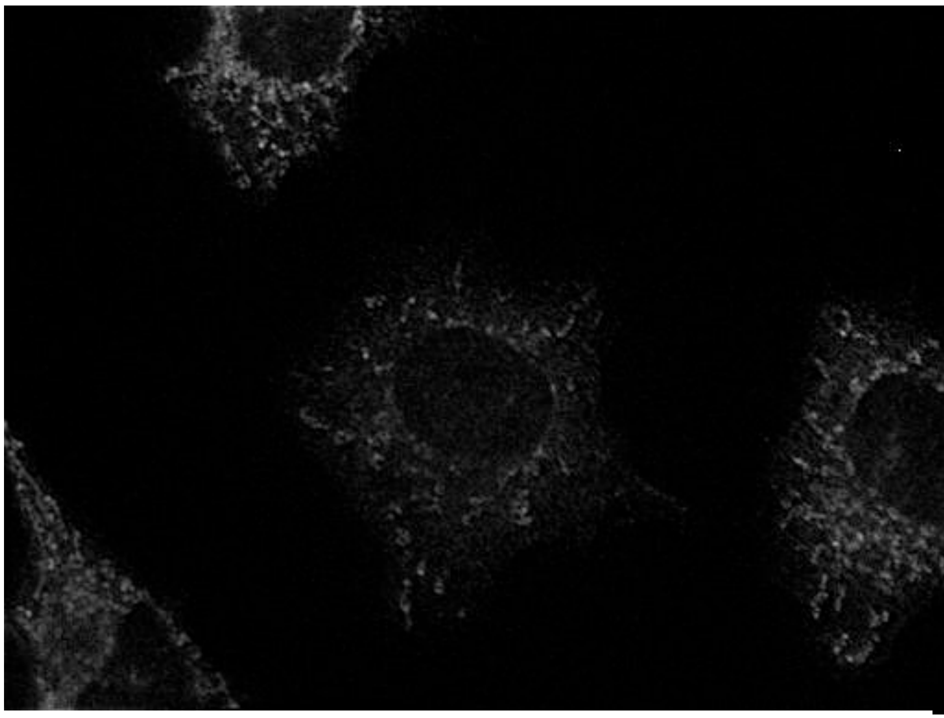
Ridler-Calvard Method

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

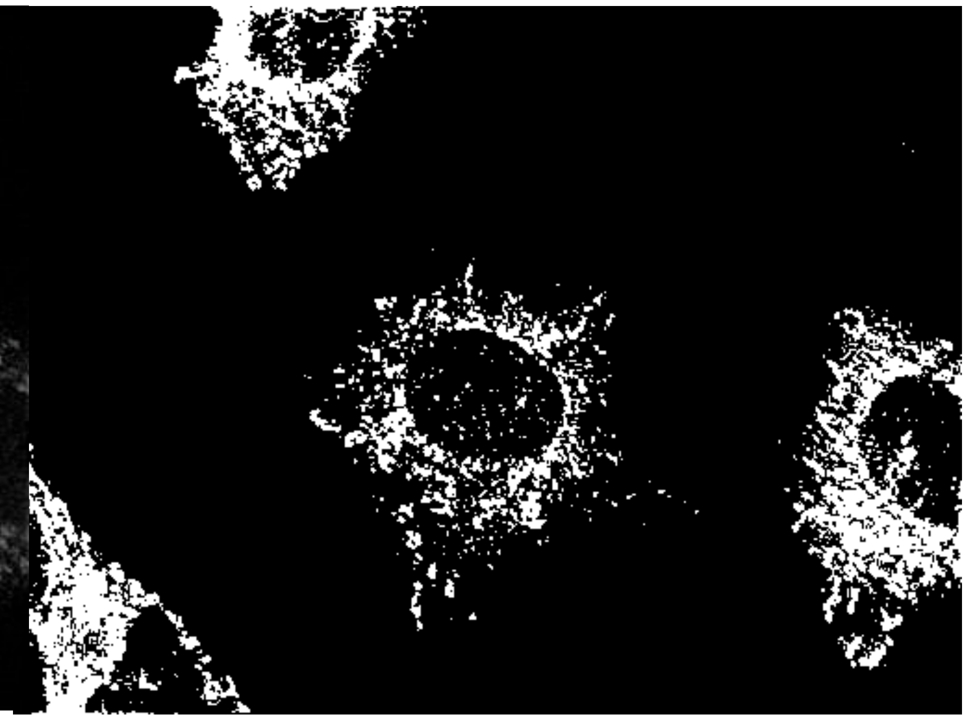


Ridler-Calvard Method

original



original



thresholded

Otsu Method

- Find threshold to minimize the variances of the pixels below and above it
- Otsu, N., (1979) A Threshold Selection Method from Gray-Level Histograms, *IEEE Transactions on Systems, Man, and Cybernetics*, 9:62-66.

Adaptive Thresholding

- Various approaches available
- Basic principle is use automated methods over small regions and then interpolate to form a smooth surface

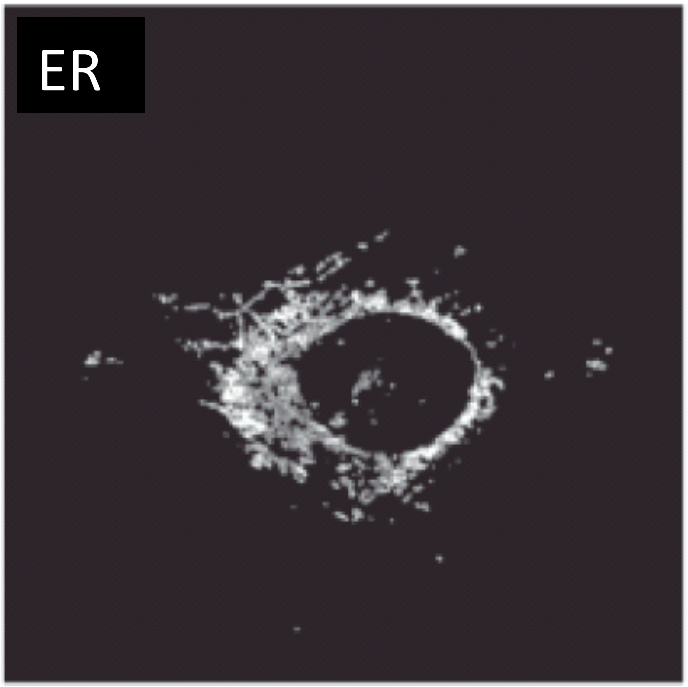
Object finding

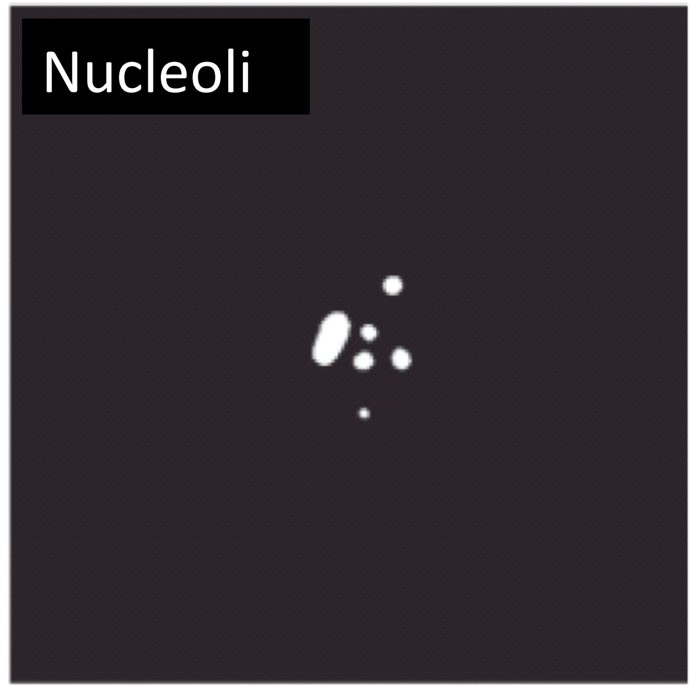
- After choice of threshold, define objects as sets of touching pixels that are above threshold

2D Morphological Features

- Number of objects
- Size and shape of objects
  - Average number of pixels
  - Holes, Ellipse parameters, Skeletons
- Position of objects relative to reference
  - Center of protein fluorescence
  - Center of DNA fluorescence
- Overlap of objects relative to reference
  - Overlap with DNA fluorescence

2D Morphological Features

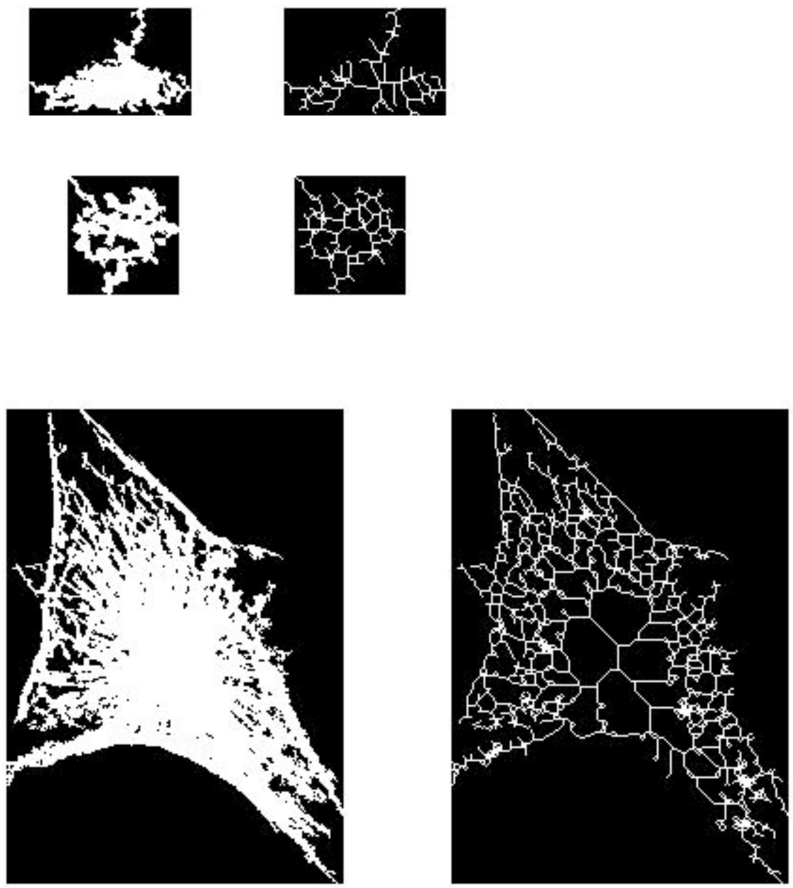




ER		Nucleoli
108	# of objects	6
83	Average size of objects	232
31	Average distance to COF	4

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

2D Skeleton Features



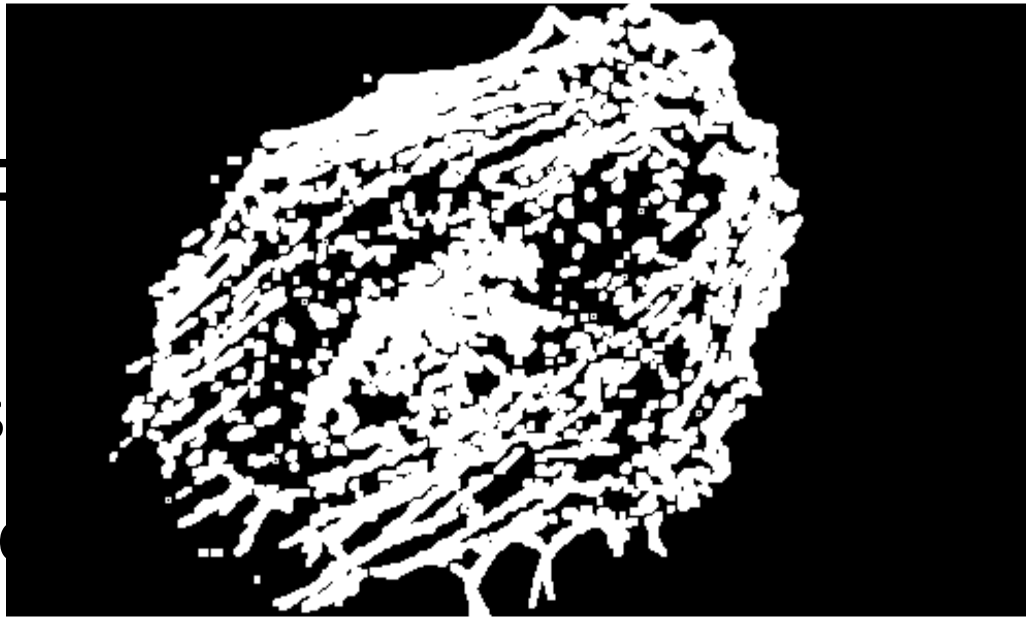
Features
average length of the skeleton
average ratio of skeleton length to area of the convex hull of the skeleton
fraction of object pixels contained within the skeleton
fraction of object fluorescence contained within the skeleton
ratio of the number of branch points in the skeleton to the length of skeleton

Suitability of Morphological Features for Classification

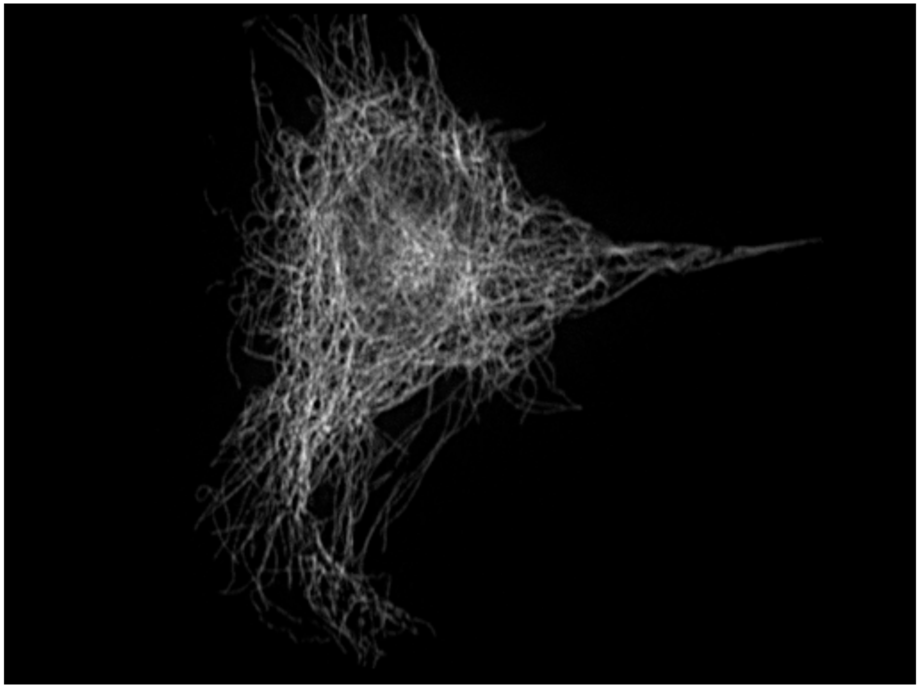
- Images for some subcellular patterns, such as those for cytoskeletal proteins, are not well-segmented by automated thresholding
- When combined with non-morphological features, classifiers can learn to “ignore” morphological features for those classes

2D Edge F

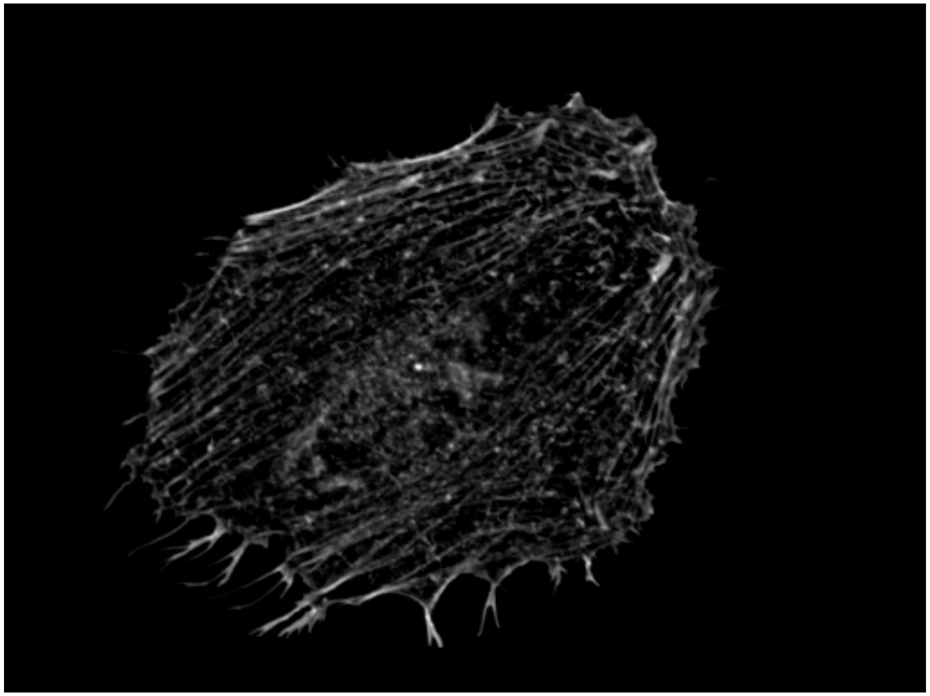
- Fraction of pixels
- Measures of edge



microtubules

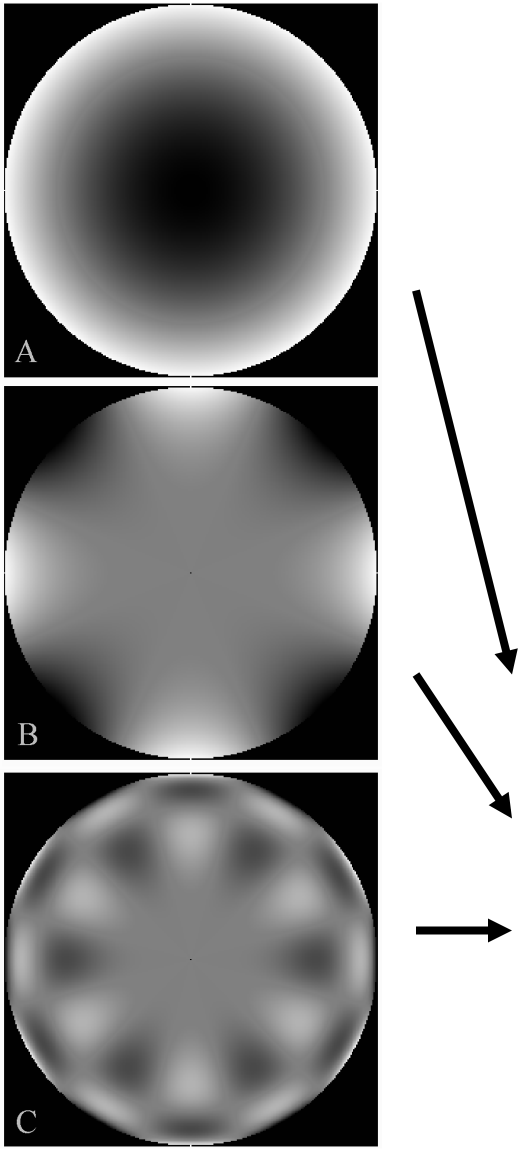


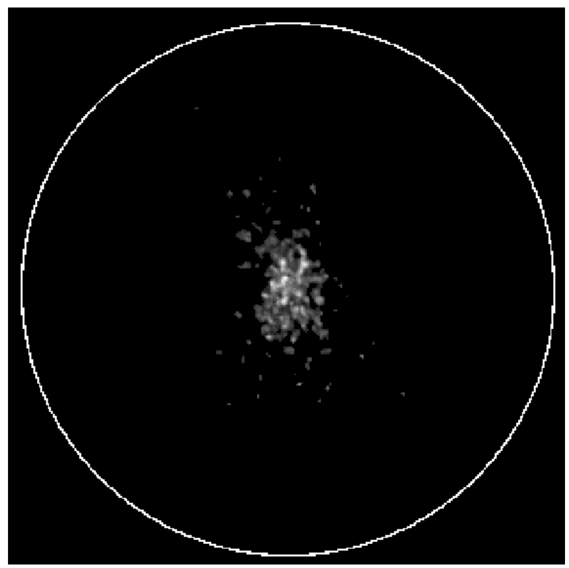
actin filaments



2D Zernike Moment Features

- Shape similarity of protein image to Zernike polynomials Z(n,l)
- 49 polynomials and 49 features





left: Zernike polynomials
A: Z(2,0)
B: Z(4,4)
C: Z(10,6)

right: lamp2 image

2D Haralick Texture Features

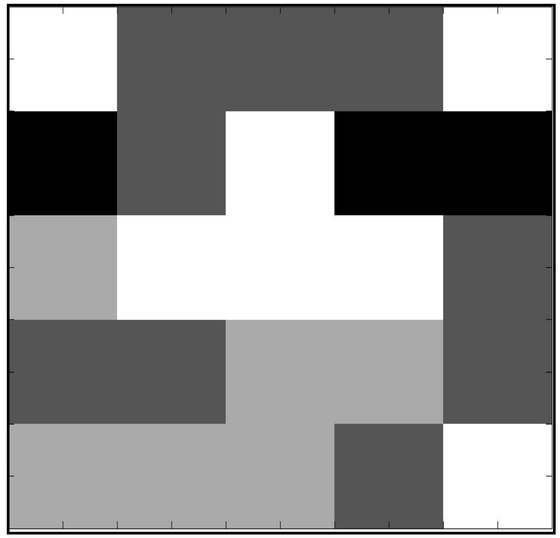
- Correlations of adjacent pixels in gray level images
- Start by calculating co-occurrence matrix P:

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

  Element P(i,j) is the probability of a pixel with value i being adjacent to a pixel with value j
- Four directions in which a pixel can be adjacent
- Each direction considered separately and then features averaged across all directions

Co-occurrence Matrices

Example image with 4 gray levels



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

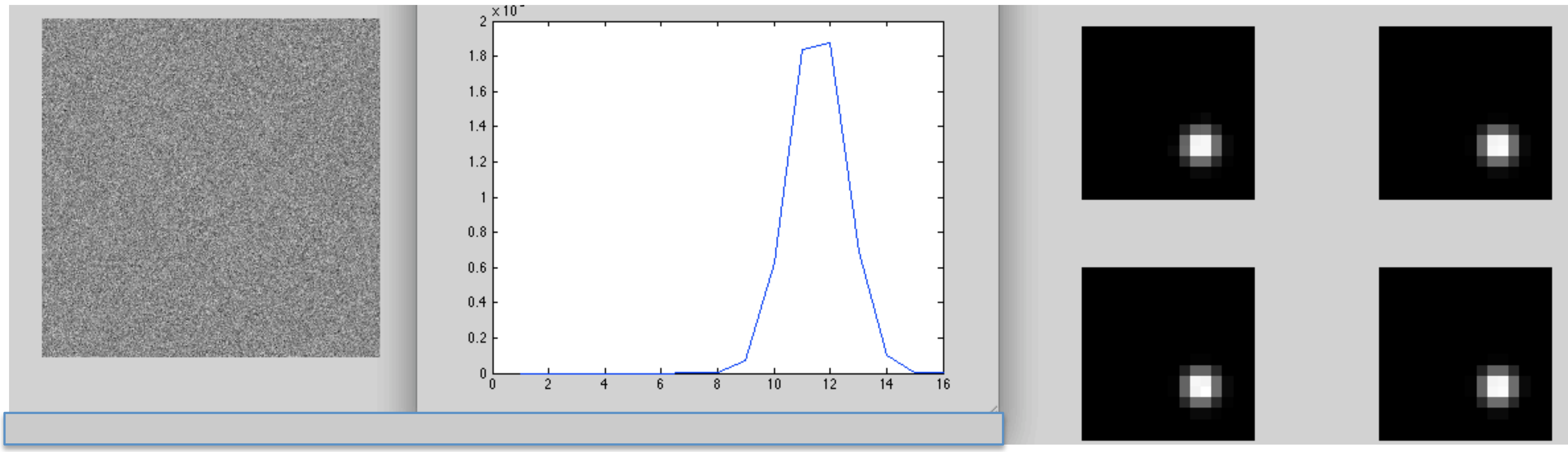
	1	2	3	4
1	0	2	1	3
2	2	4	4	4
3	1	4	2	2
4	2	3	2	2

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

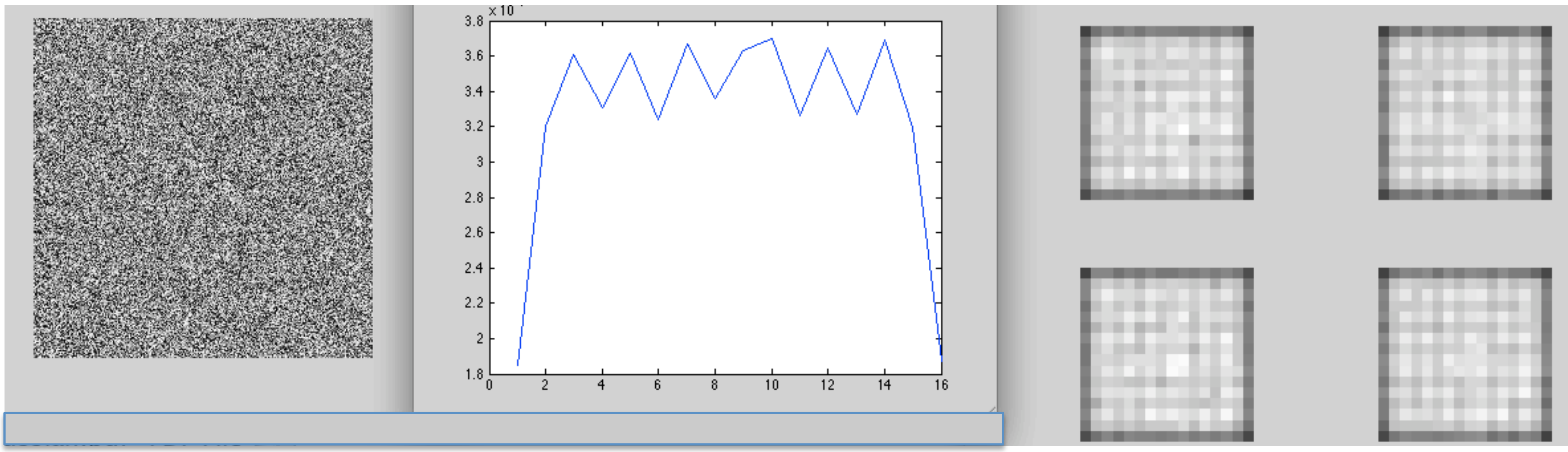
	1	2	3	4
1	0	1	0	3
2	1	4	3	3
3	0	3	4	1
4	3	3	1	2

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

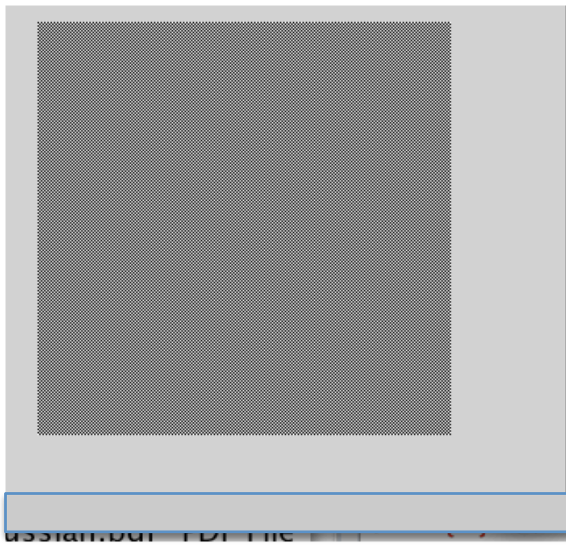
Solid plus some noise

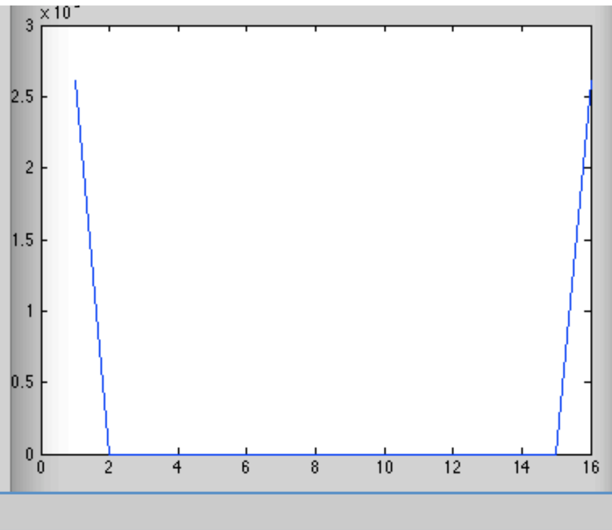


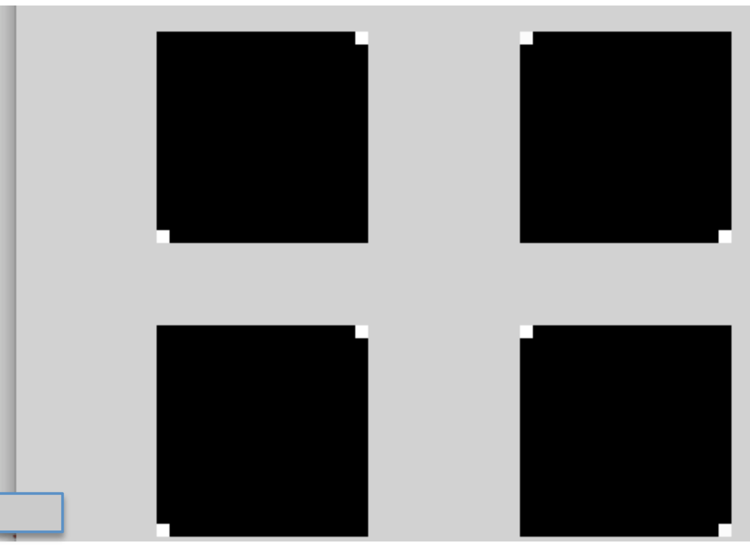
Random



Checkerboard







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

Other categories

- Other moments
- Wavelets
- Gabor

3D Features

- Morphological
  - straightforward extension
  - equivalence between voxel dimensions?
- Edge
  - 3D edges expensive
  - use stack of 2D edges
- Texture
  - straightforward extension
  - voxel dimensions isotropic?

Object level features (SOF)

- Subset of SLFs calculated on single objects

Index	Feature Description
SOF1.1	Number of pixels in object
SOF1.2	Distance between object Center of Fluorescence (COF) and DNA COF
SOF1.3	Fraction of object pixels overlapping with DNA
SOF1.4	A measure of eccentricity of the object
SOF1.5	Euler number of the object
SOF1.6	A measure of roundness of the object
SOF1.7	The length of the object's skeleton
SOF1.8	The ratio of skeleton length to the area of the convex hull of the skeleton
SOF1.9	The fraction of object pixels contained within the skeleton
SOF1.10	The fraction of object fluorescence contained within the skeleton
SOF1.11	The ratio of the number of branch points in skeleton to length of skeleton

Field level features

- Subset of SLFs that do not require segmentation into single cells
  - Average object features
  - Texture features (on whole field)
  - Edge features (on whole field)

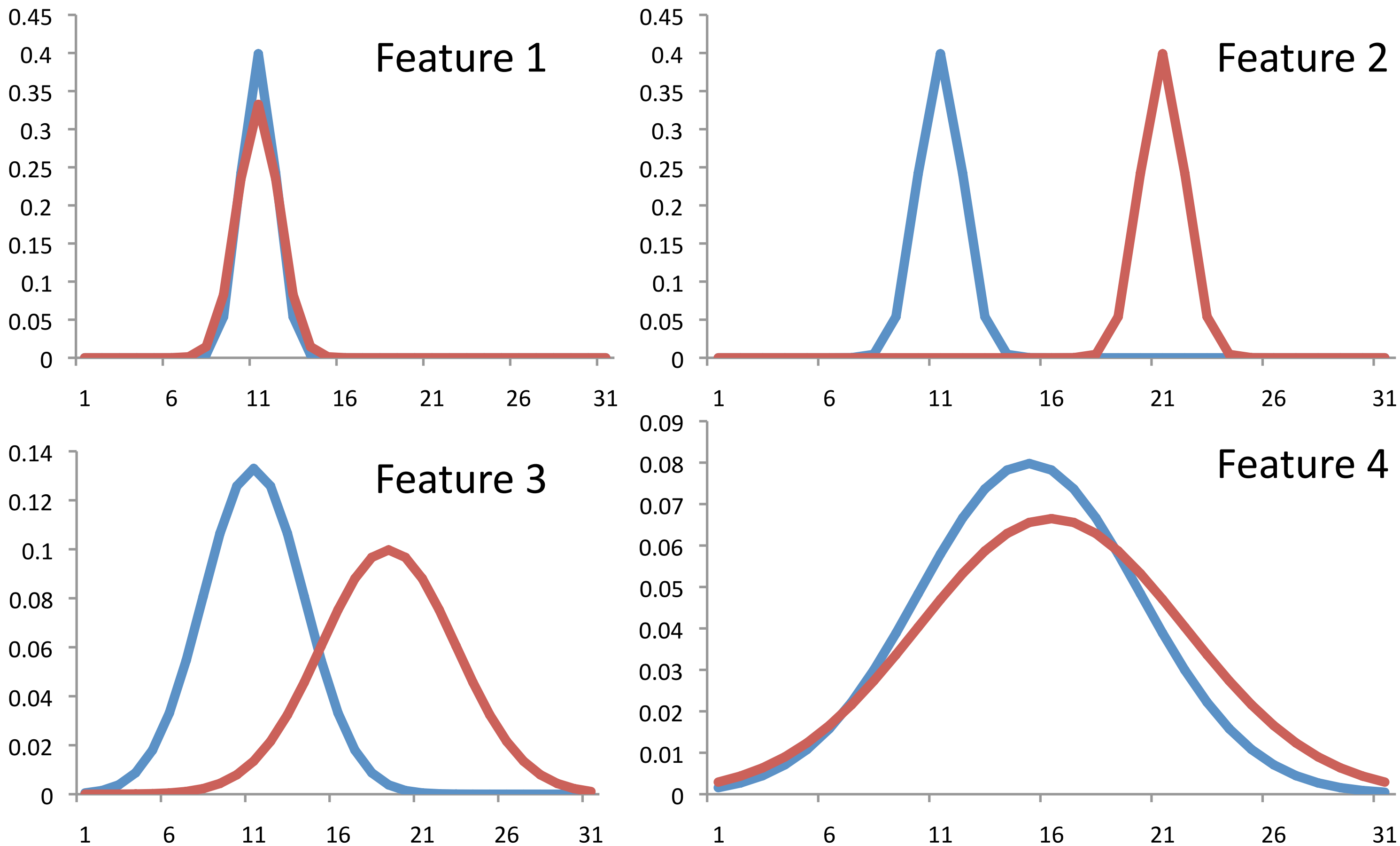
2Dt or 3Dt Features
Temporal Texture Features

- **Haralick texture features** describe the correlation in intensity of pixels that are next to each other in **space**.
  - These have been valuable for classifying static patterns.
- **Temporal texture features** describe the correlation in intensity of pixels in the same position in images next to each other over **time**.

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

Basic principle of feature selection



red=class 1, blue=class 2

Need to consider multivariate distance

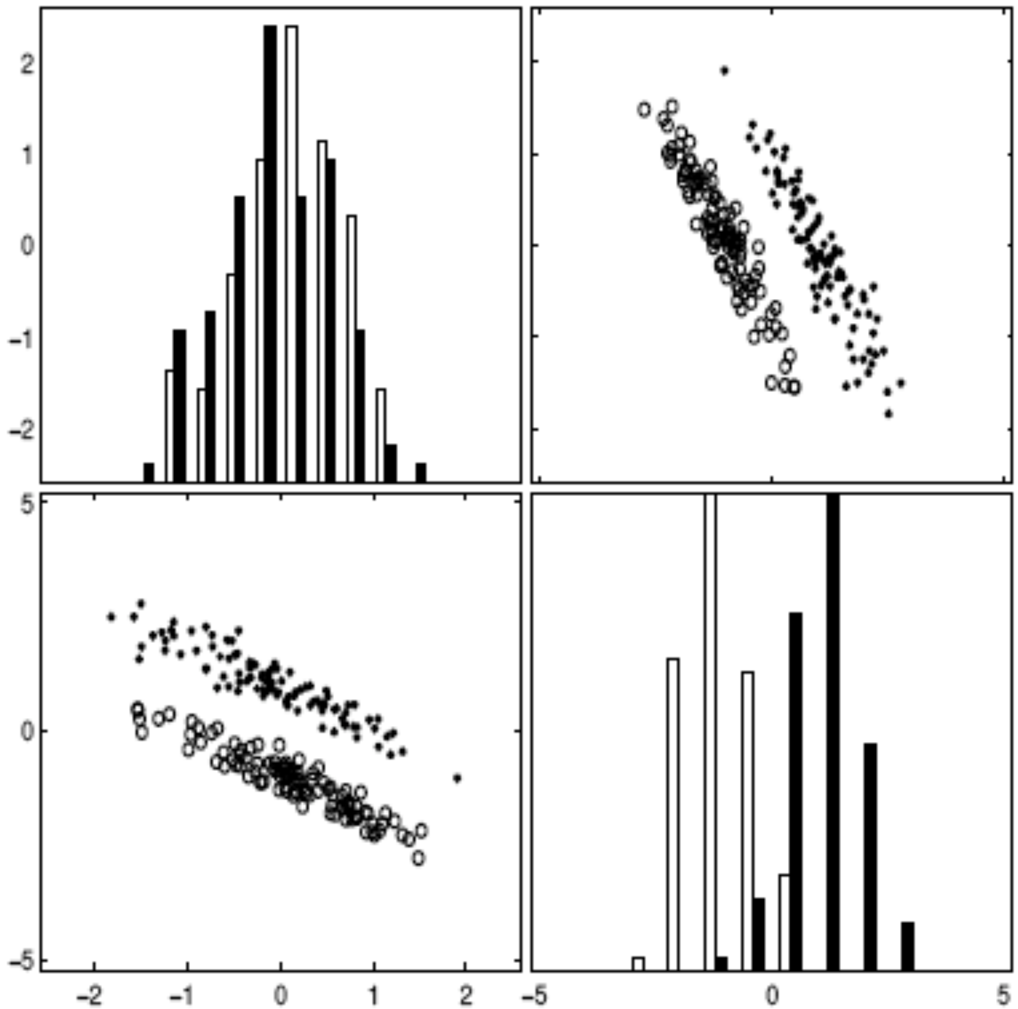


Figure from Guyon & Elisseeff

Bad and Good Covariance

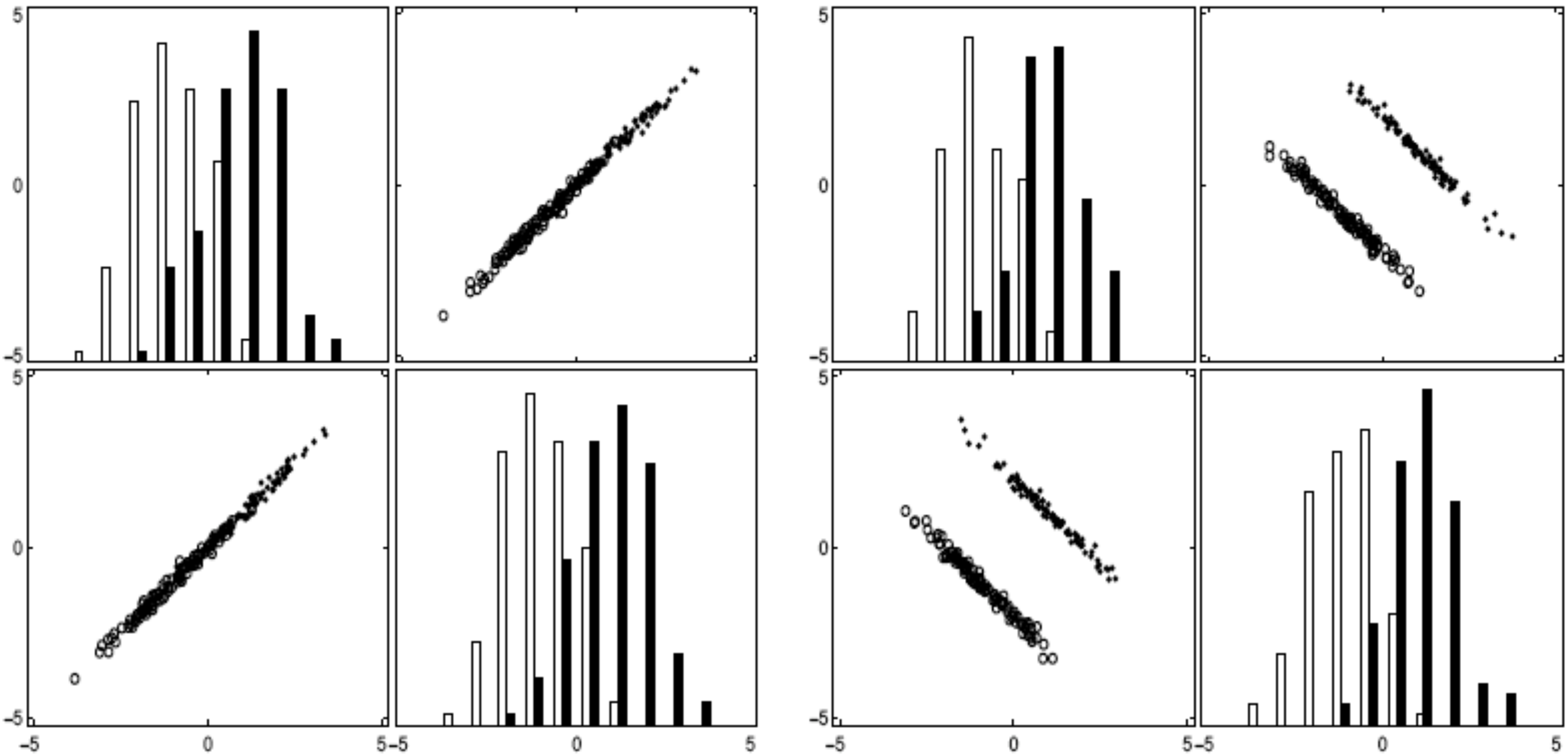
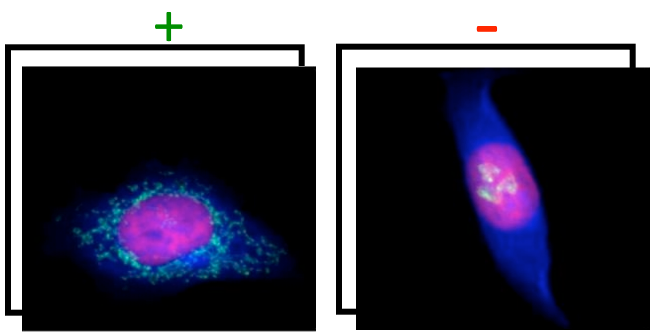


Figure from Guyon & Elisseeff

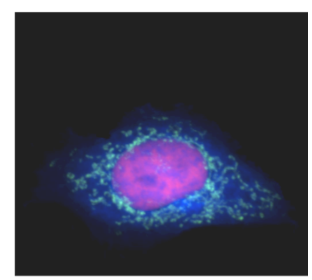
Feature Selection Methods

- Principal Components Analysis
- Non-Linear Principal Components Analysis
- Independent Components Analysis
- Information Gain
- Stepwise Discriminant Analysis
- Genetic Algorithms
- Max-Relevance, Min-Redundancy

Classification: Simple two class problem



???

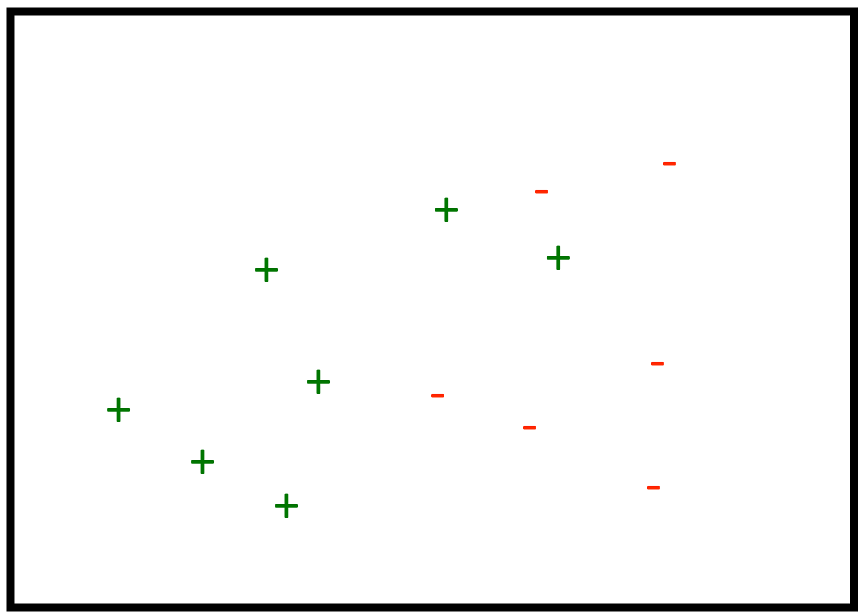


Describe each image by features
Train classifier

k-Nearest Neighbor (kNN)

- In feature space, training examples are

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

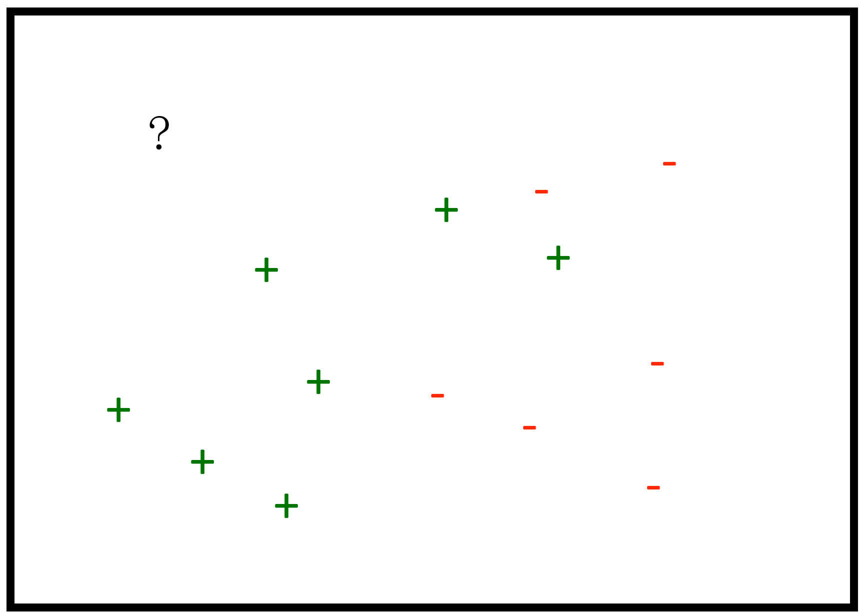


Feature #1 (e.g.., ‘area’)

k-Nearest Neighbor (kNN)

- We want to label ‘**?**’

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

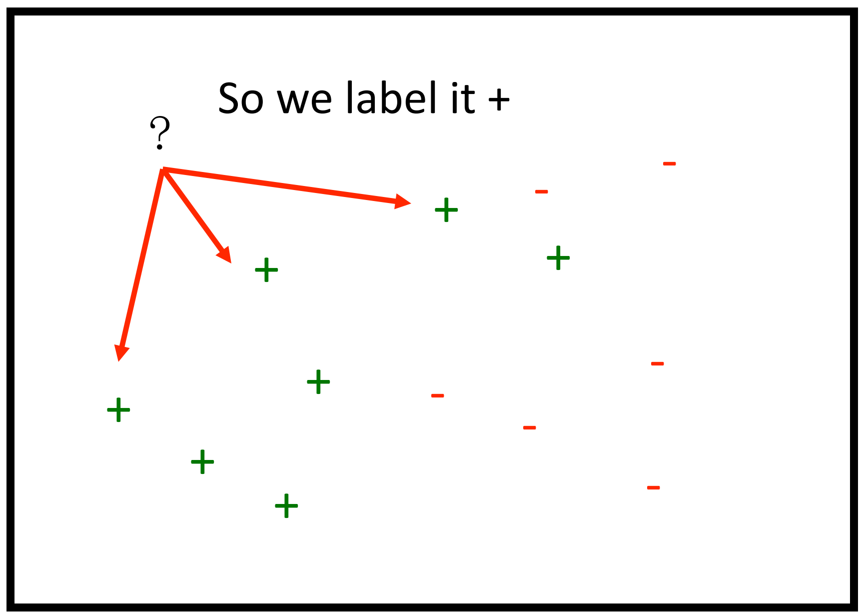


Feature #1 (e.g.., ‘area’)

k-Nearest Neighbor (kNN)

- Find k nearest neighbors and vote

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



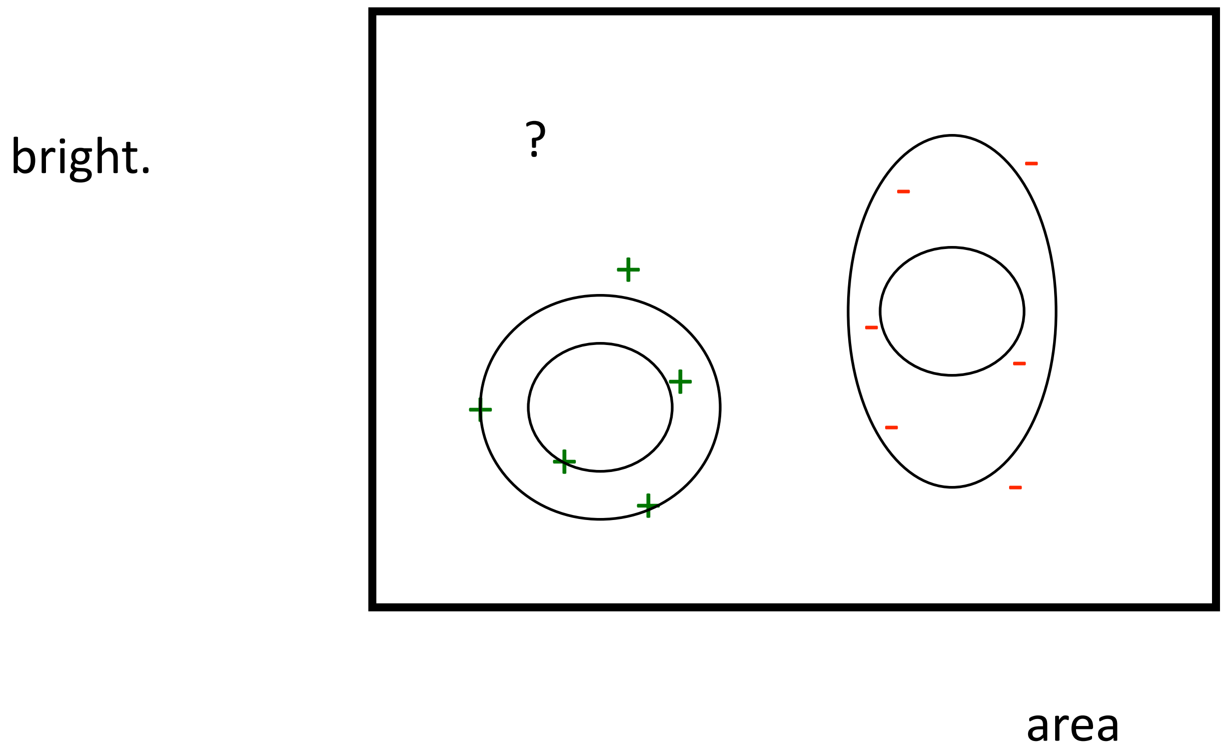
for k=3,

nearest
neighbors
are

Feature #1 (e.g.., ‘area’)

Linear Discriminants

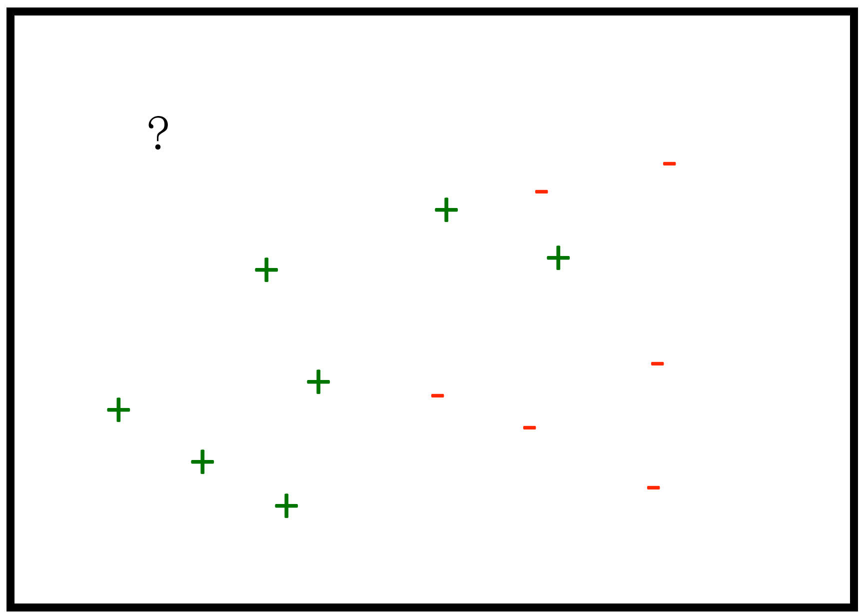
- Fit multivariate Gaussian to each class
- Measure distance from ? to each Gaussian



Decision trees

- Again we want to label ‘**?**’

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

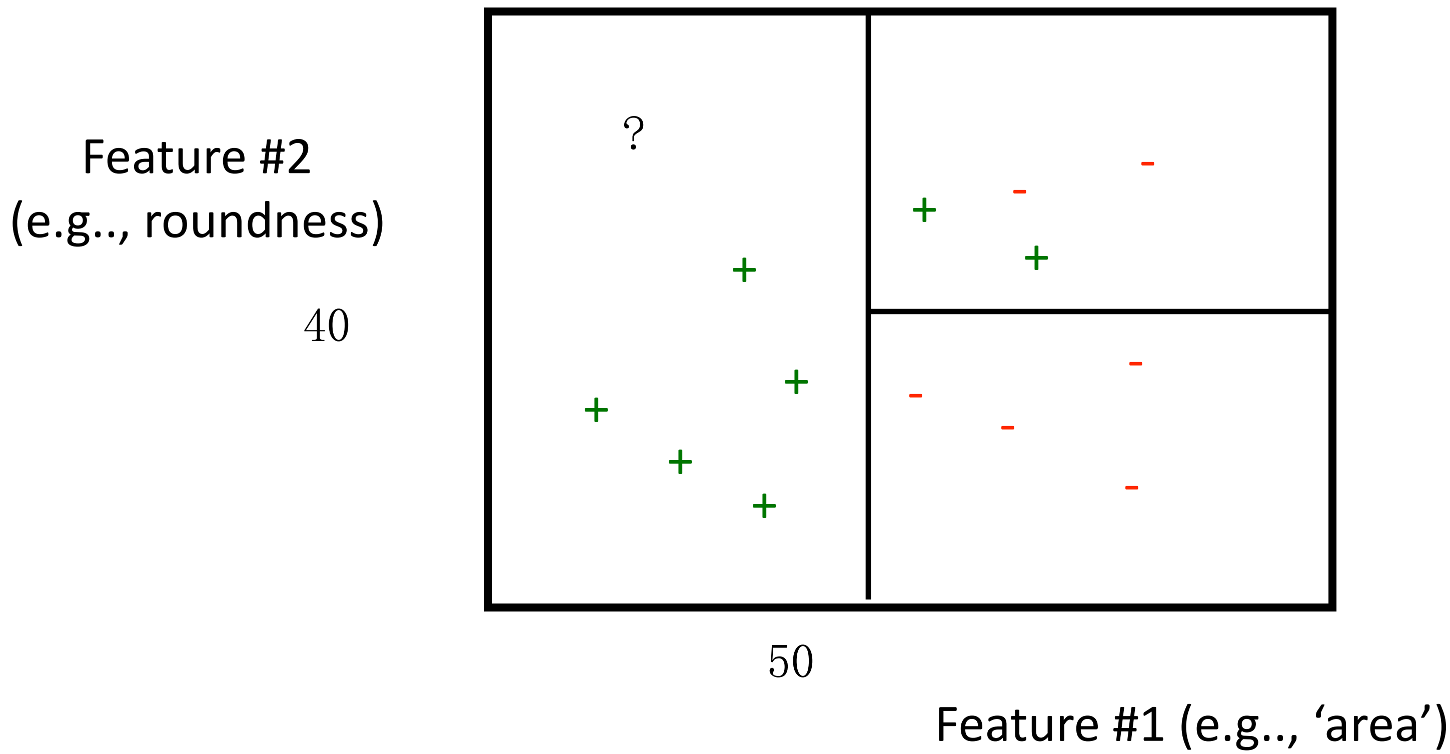


Feature #1 (e.g.., ‘area’)

Slide courtesy of Christos Faloutsos

Decision trees

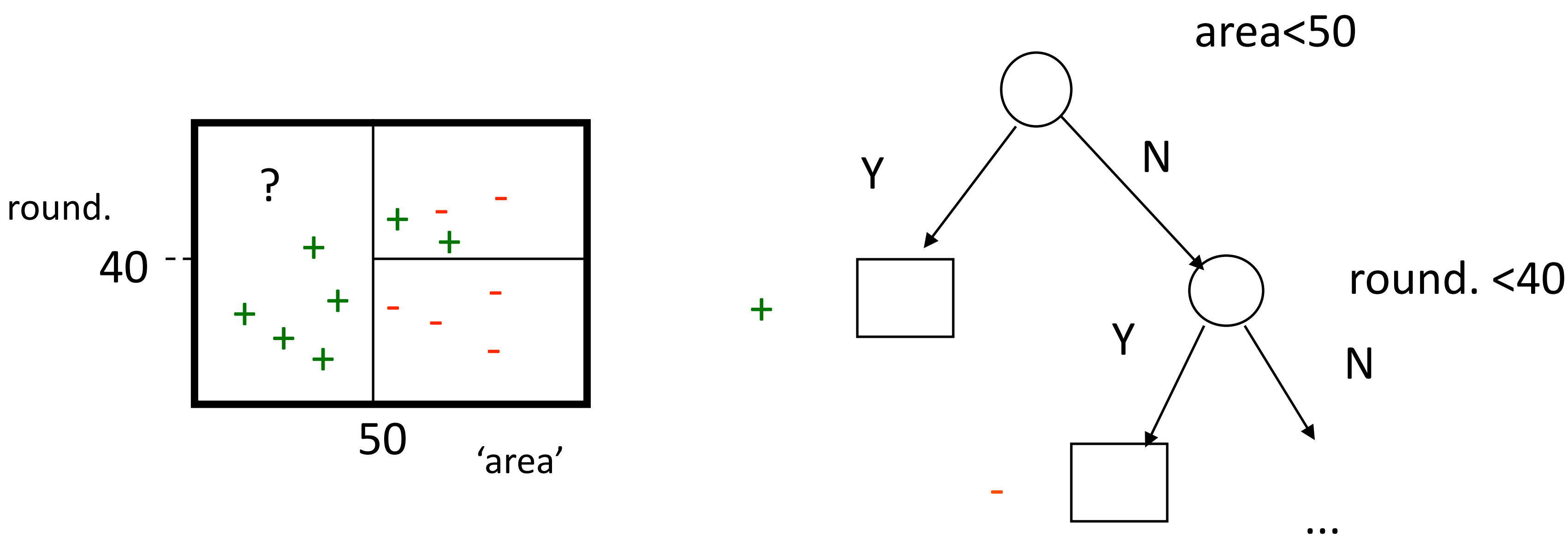
- so we build a decision tree:



Slide courtesy of Christos Faloutsos

Decision trees

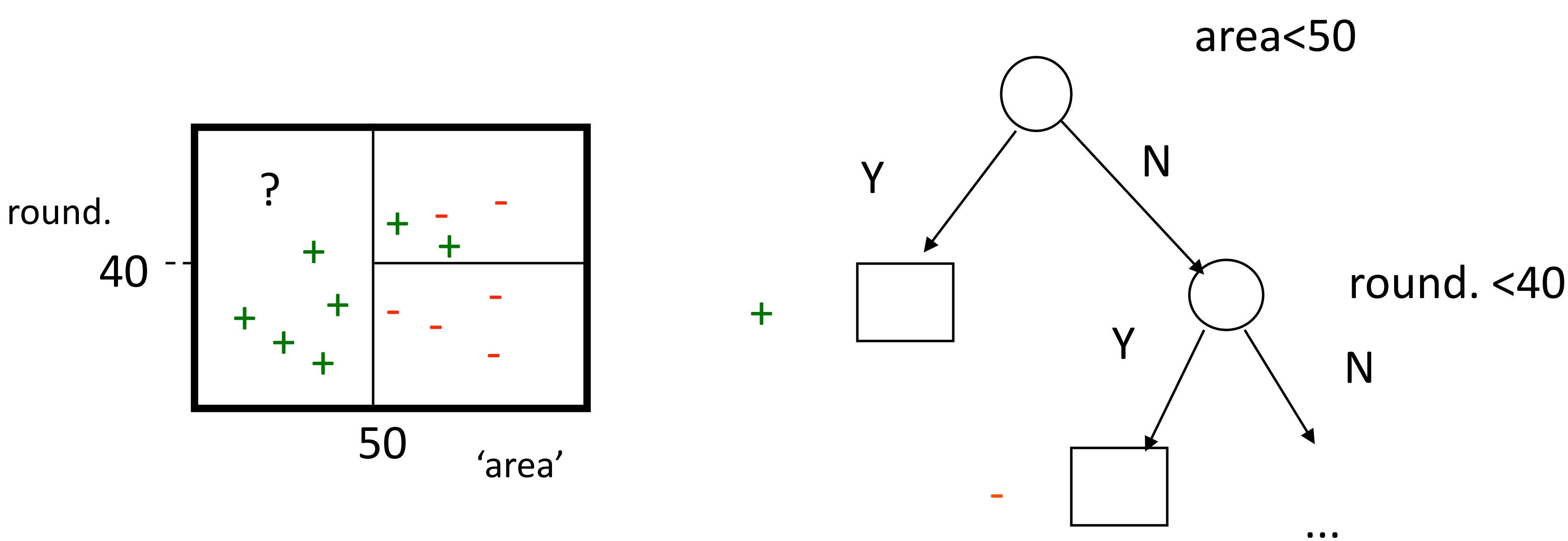
- so we build a decision tree:



Slide courtesy of Christos Faloutsos

Decision trees

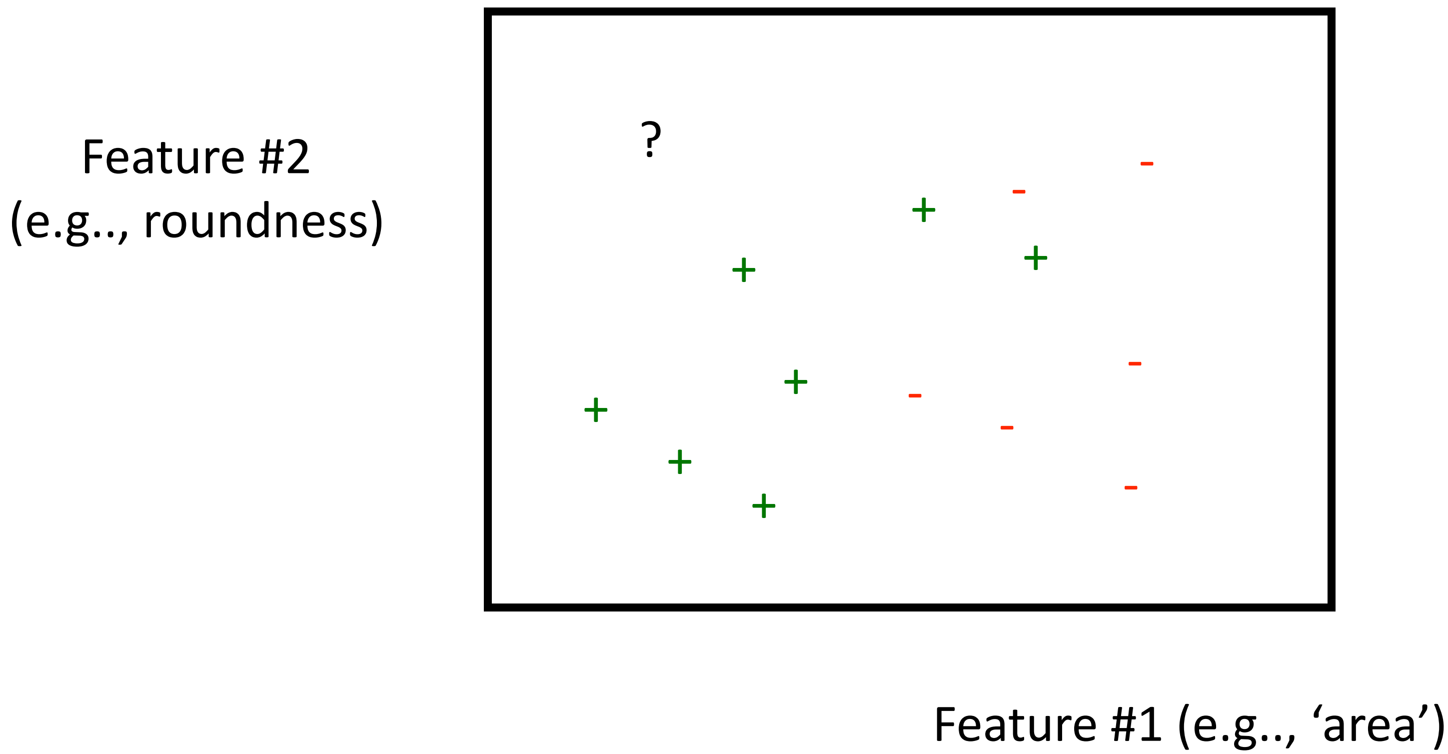
- Goal: split address space in (almost) homogeneous regions



Slide courtesy of Christos Faloutsos

Support vector machines

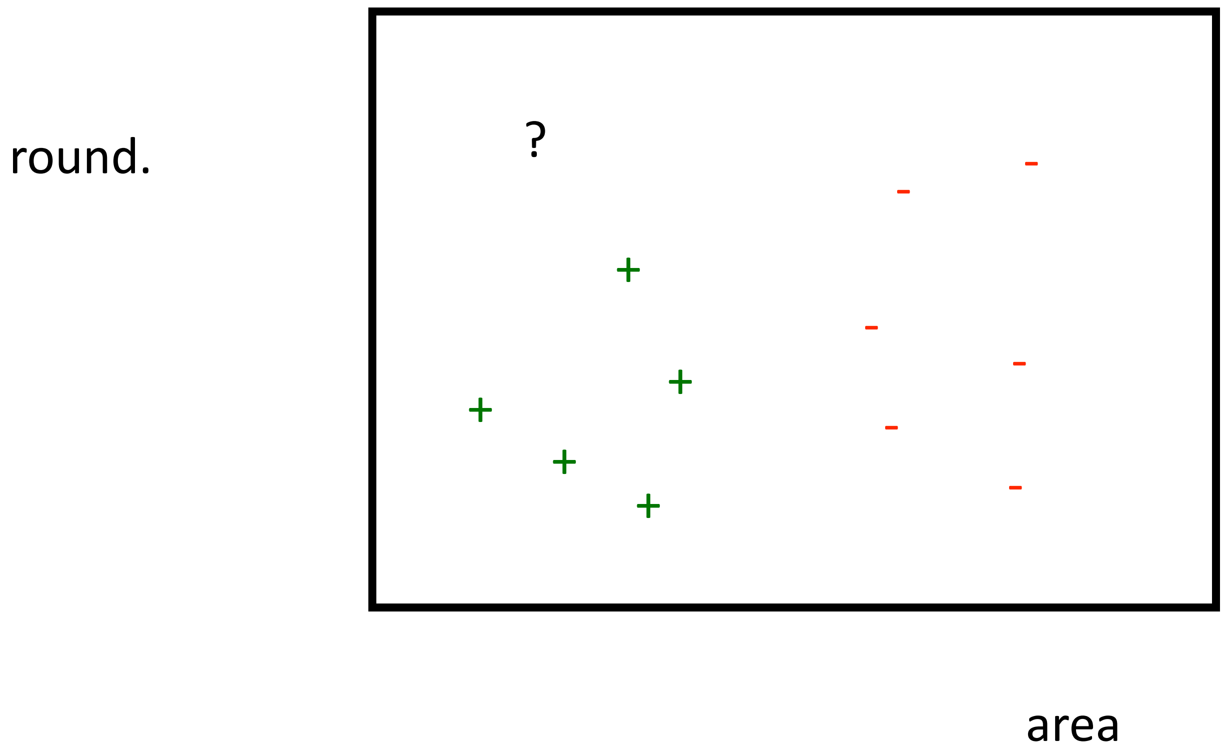
- Again we want to label '**?**'



Slide courtesy of Christos Faloutsos

Support Vector Machines (SVMs)

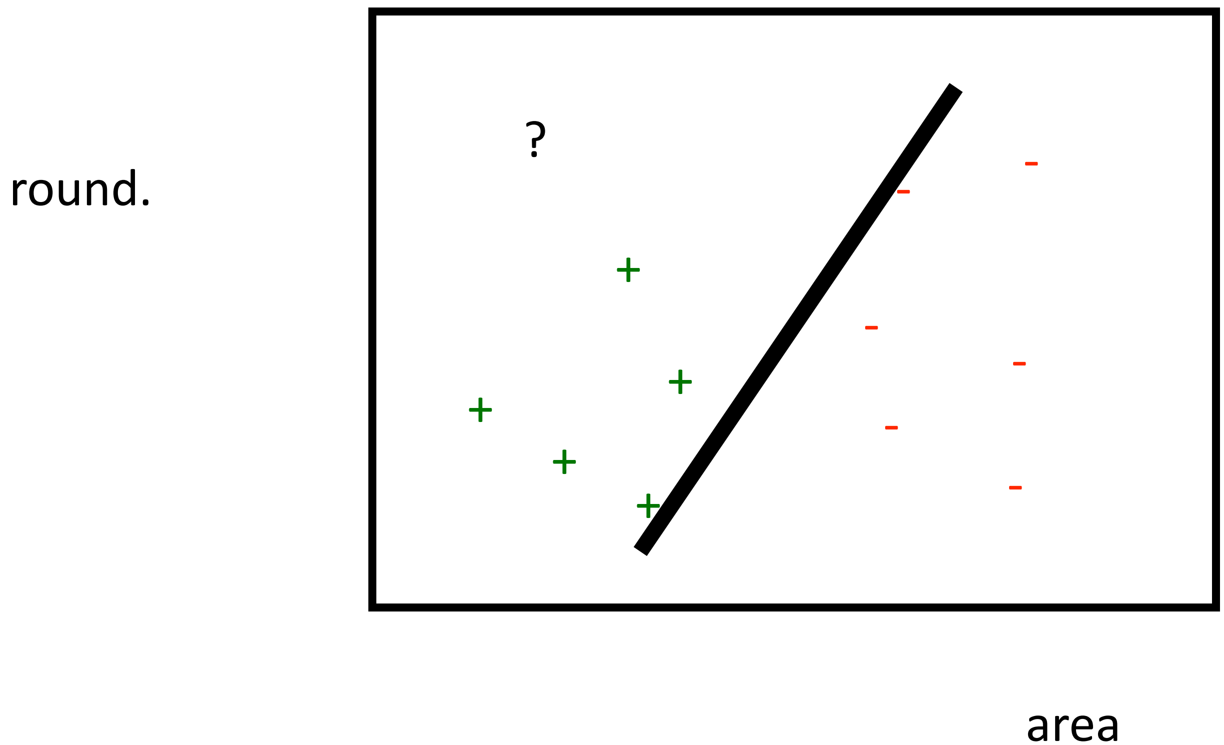
- Use single linear separator??



Slide courtesy of Christos Faloutsos

Support Vector Machines (SVMs)

- Use single linear separator??

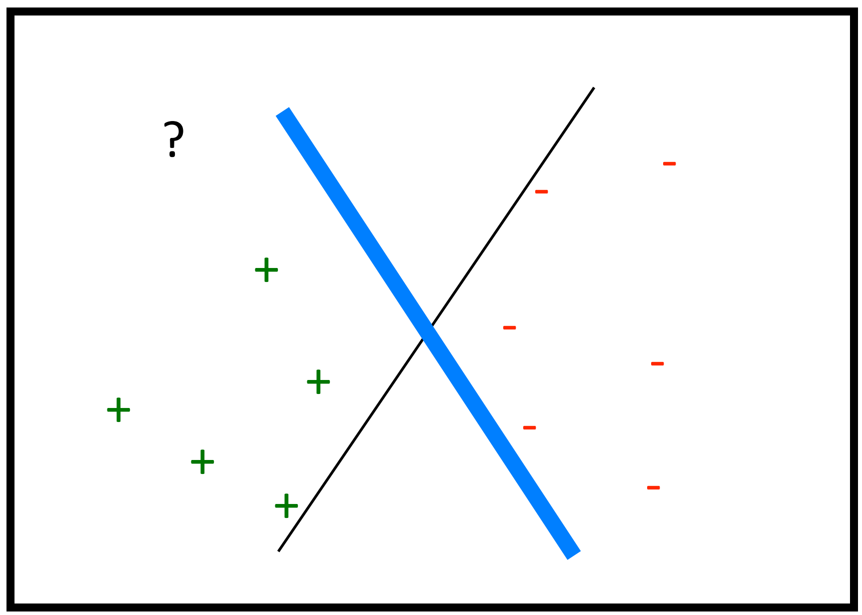


Slide courtesy of Christos Faloutsos

Support Vector Machines (SVMs)

- Use single linear separator??

round.

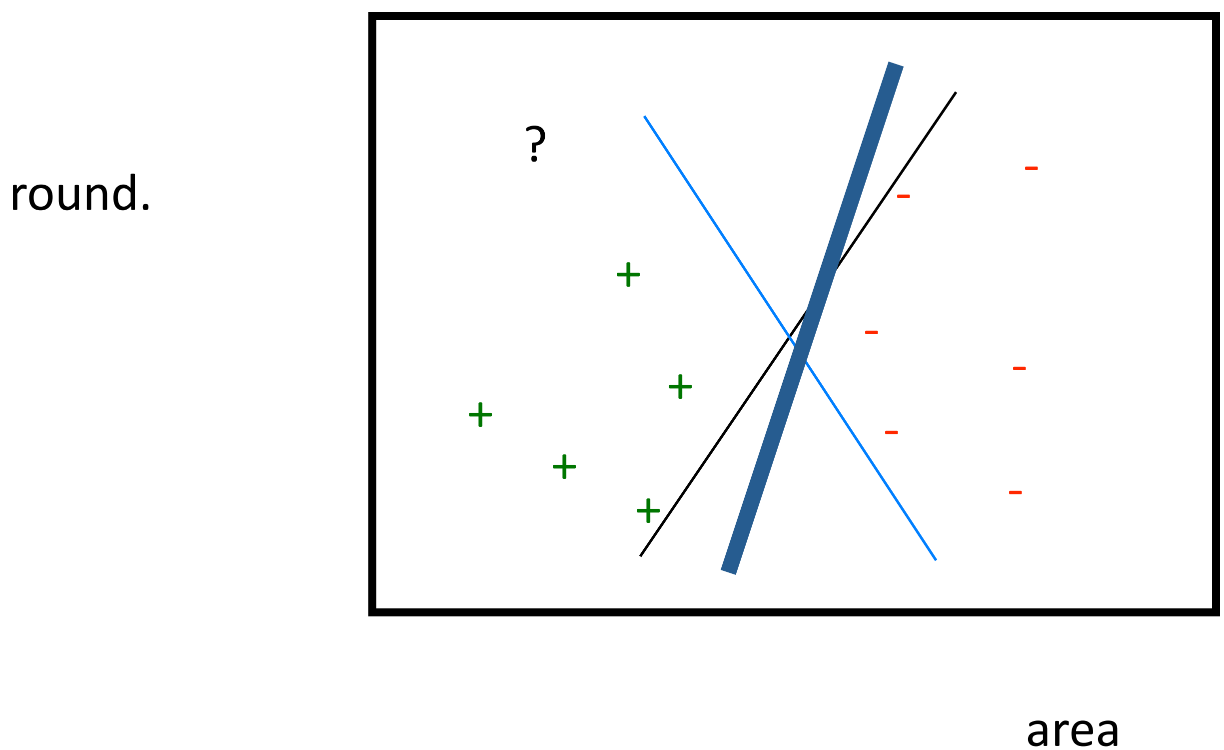


area

Slide courtesy of Christos Faloutsos

Support Vector Machines (SVMs)

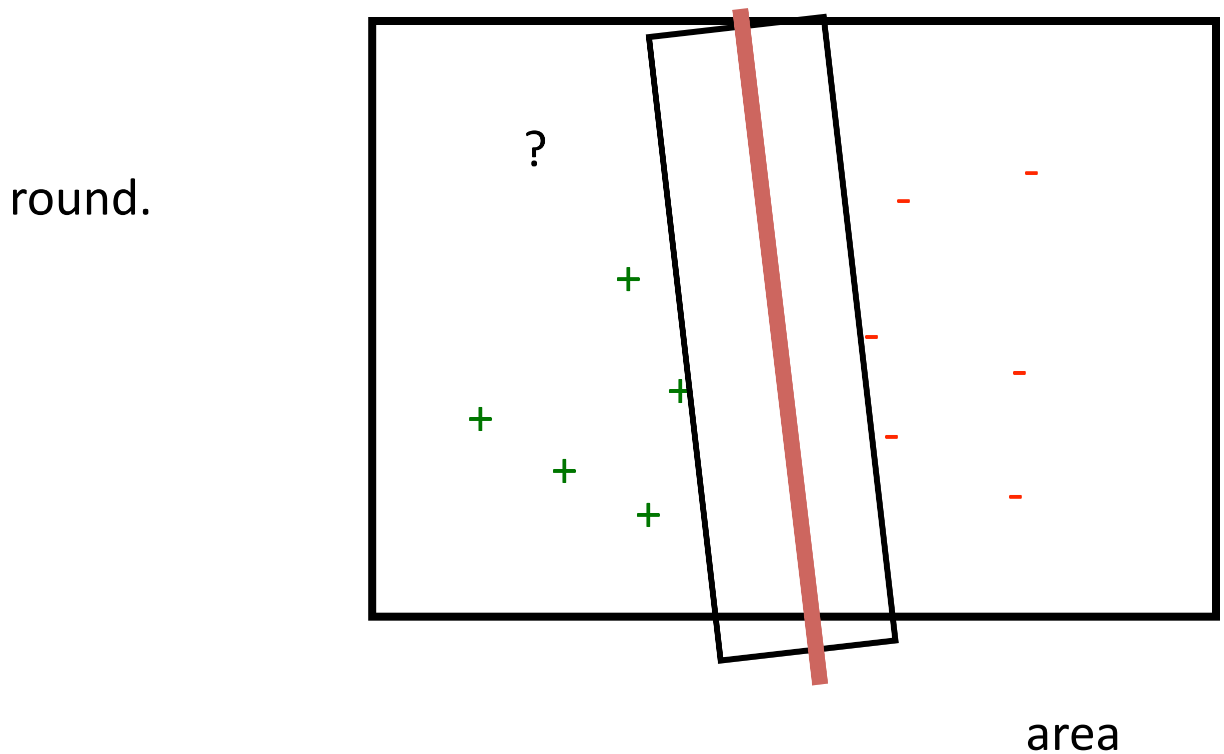
- Use single linear separator??



Slide courtesy of Christos Faloutsos

Support Vector Machines (SVMs)

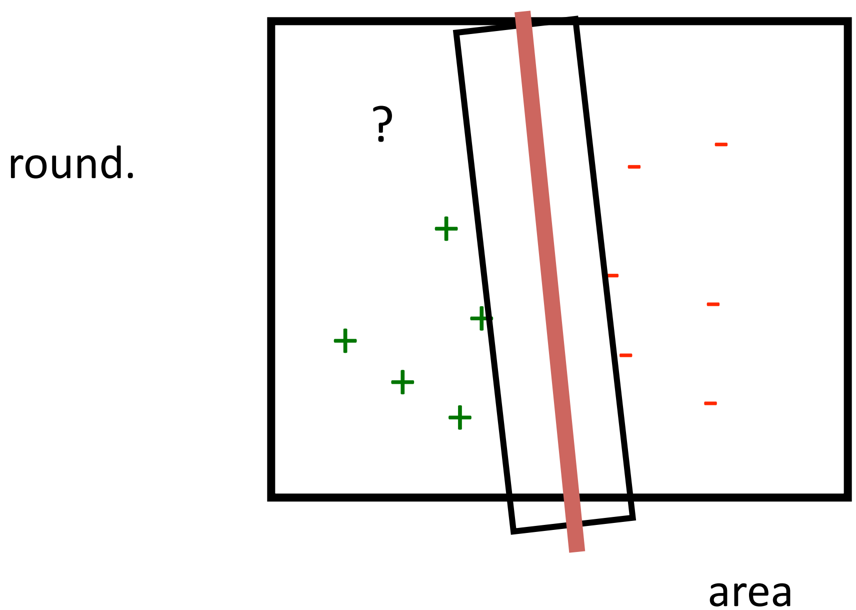
- Use single linear separator??



Slide courtesy of Christos Faloutsos

Support Vector Machines (SVMs)

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



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

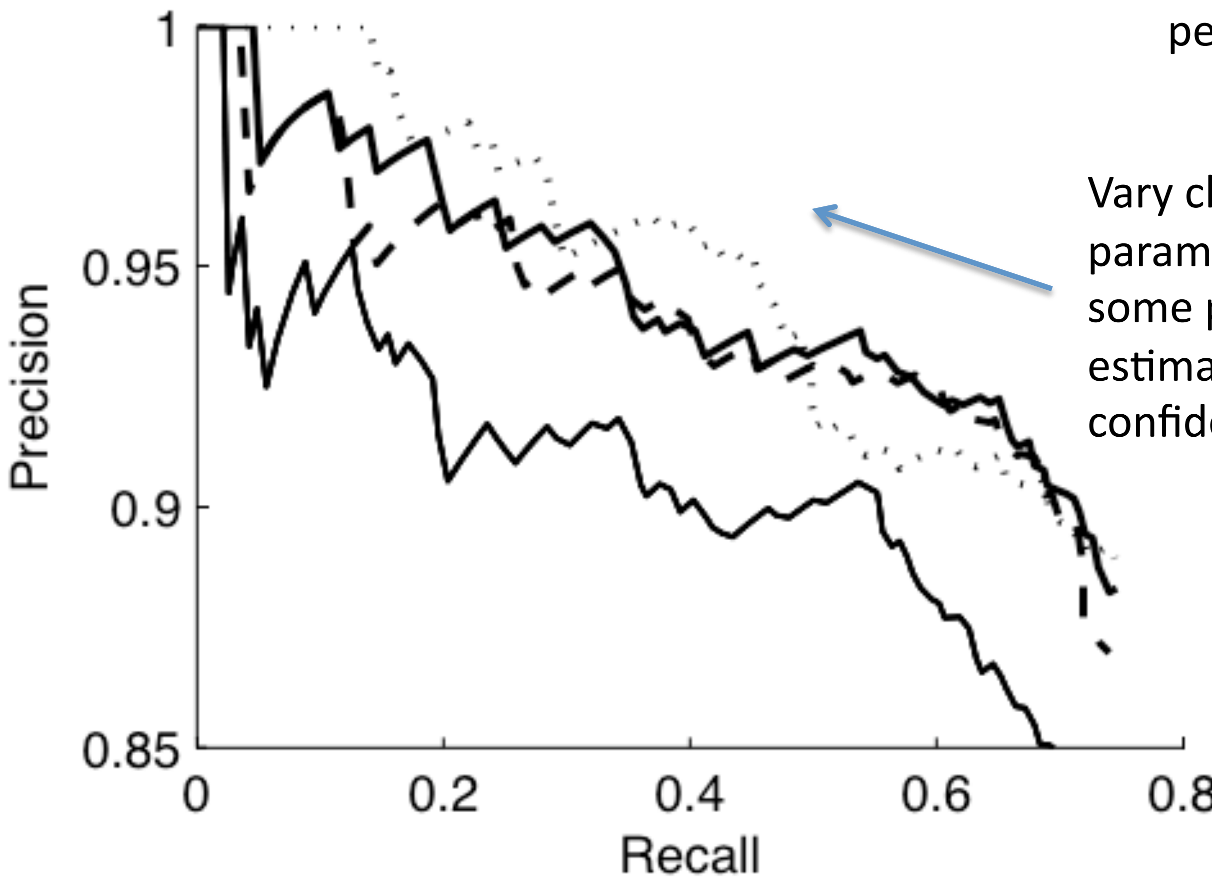
- If we train a classifier to minimize error on a set of data, have no ability to estimate (generalize) error that will be seen on new dataset
- To calculate *generalizable* accuracy, we use ***n*-fold cross-validation**
- Divide images into *n* sets, train using *n*-1 of them and test on the remaining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

Describing classifier errors

- For multiclass classifiers, define
  - Recall = #correct /#samples
  - Precision = #correct/#predictions
    - if prediction not made for all samples
  - F-measure= 2*Recall*Precision/(Recall + Precision)

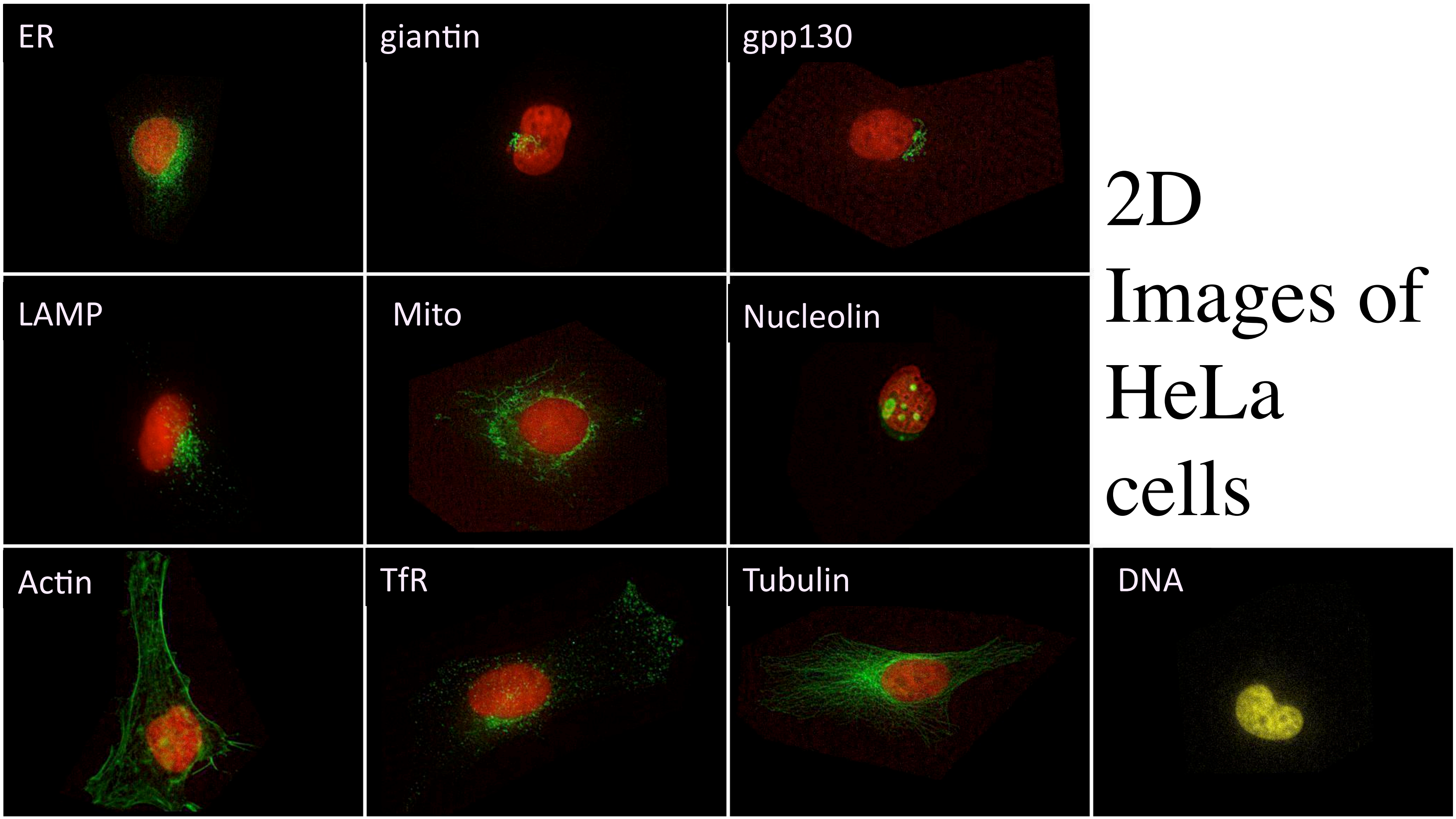
Precision-recall analysis

Ideal performance



Vary classifier parameter to "loosen" some performance estimate: i.e., confidence

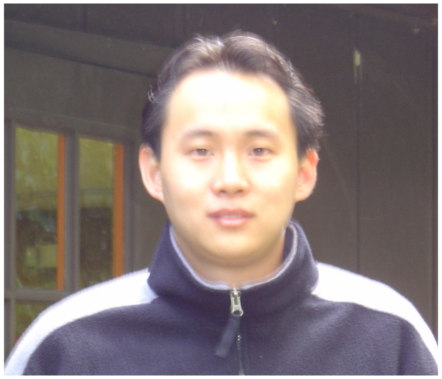
Goal: Learn to recognize all major subcellular patterns



2D Images of HeLa cells

Murphy et al 2000; Boland & Murphy 2001; Huang & Murphy 2004

Kai Huang



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%

Murphy et al 2003

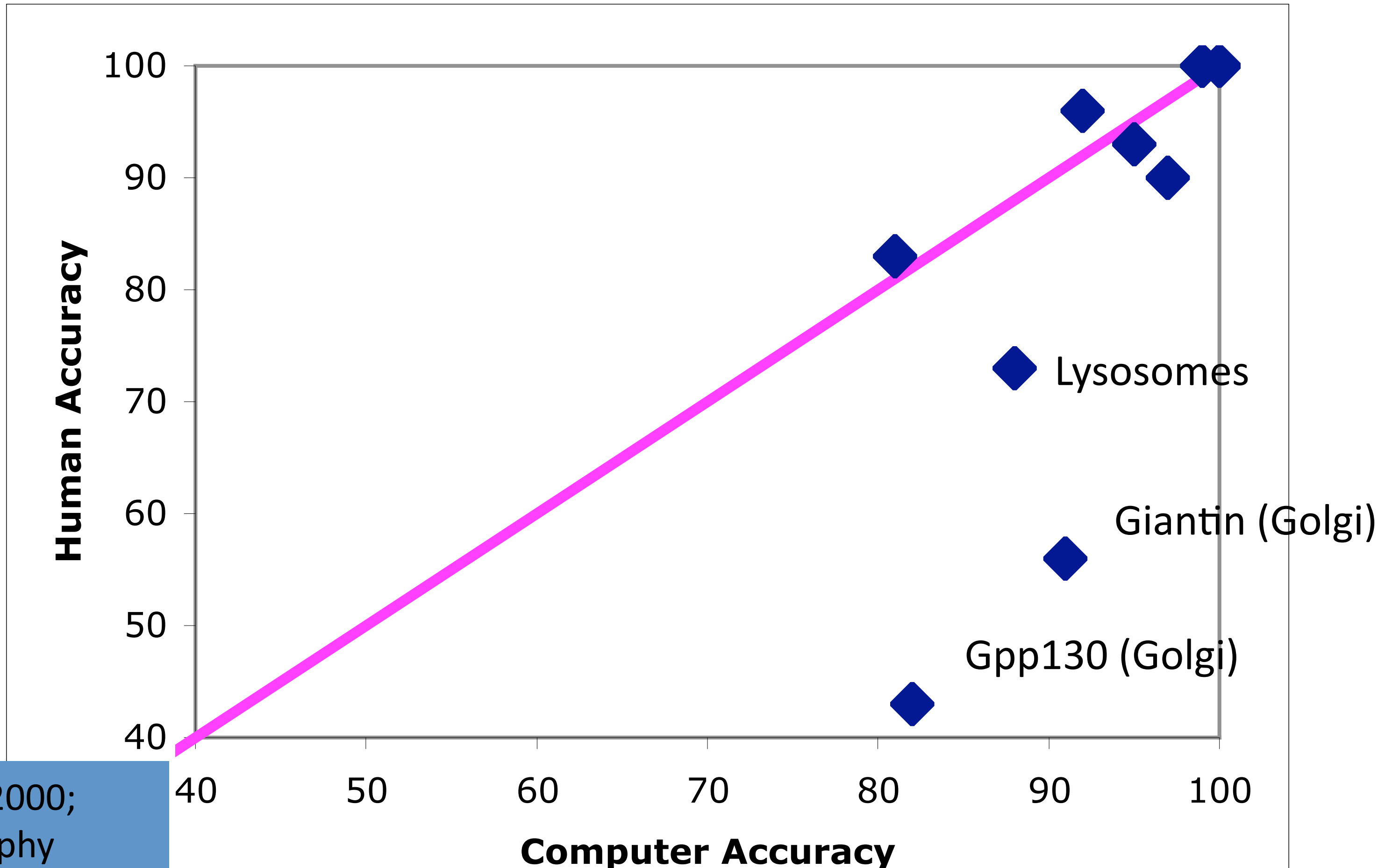
Human Classification Results

Greg Porreca & Meel Velliste

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%

Classification Results: Computer vs. Human



Murphy et al 2000; Boland & Murphy 2001; Murphy et al 2003; Huang & Murphy 2004

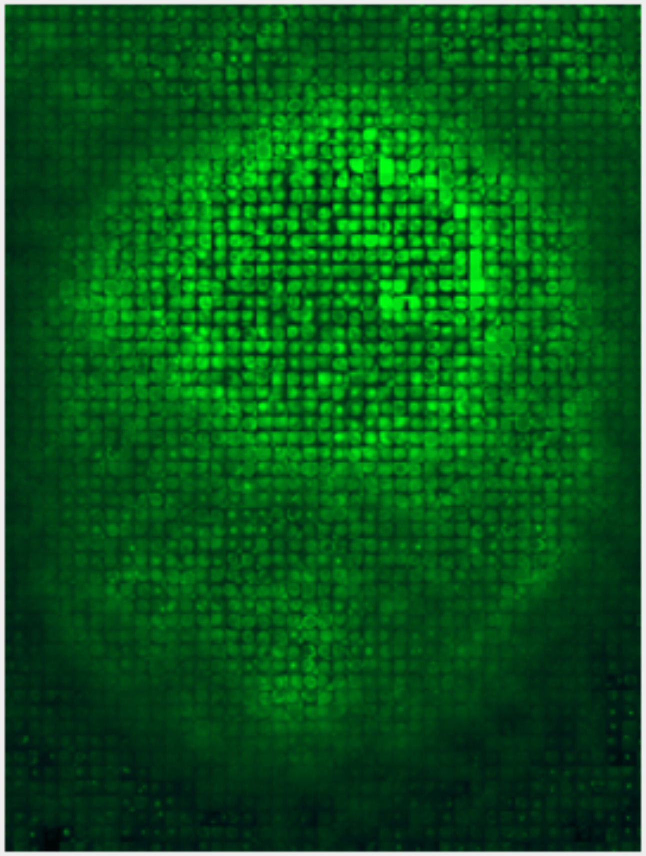
Notes: Even better results using MR methods by Kovacevic group
Even better results for 3D images

yeastgfp.ucsf.edu
YEAST GFP FUSION LOCALIZATION DATABASE

>> Advanced Query < Go >

>> Quick Search yal001c go clr

>> info
>> faq
>> help



Welcome to yeastgfp.ucsf.edu

The database of our global analysis of protein localization studies in the budding yeast, *S. cerevisiae*.

> quick case-insensitive searches of the database may be performed on yeast orf names (yal001c) or gene names (TFC3)

> separate multiple orfs/genes with a space (e.g. yal001c zwf1 bud2 etc.)

> more advanced searching and downloading can be done in Advanced Query

> GFP-tagged strains can be obtained from Invitrogen.

> TAP-tagged strains can be obtained from Open Biosystems.

> more details available in >> info >> faq >> help

This web site supports Huh, *et al.*, *Nature* **425**, 686-691 (2003). <pdf>

The quantitation data presented here is published in Ghaemmaghami, *et al.*, *Nature* **425**, 737-741 (2003). <pdf>

Detailed collection construction methods can be found in Howson *et al.*, *Comp Funct Genom* **6**, 2-16 (2005). <pdf>

This research is the work of the laboratories of Erin O'Shea and Jonathan Weissman at the University of California San Francisco.
Please direct comments, concerns, and questions to <jan.ihmels@gmail.com>

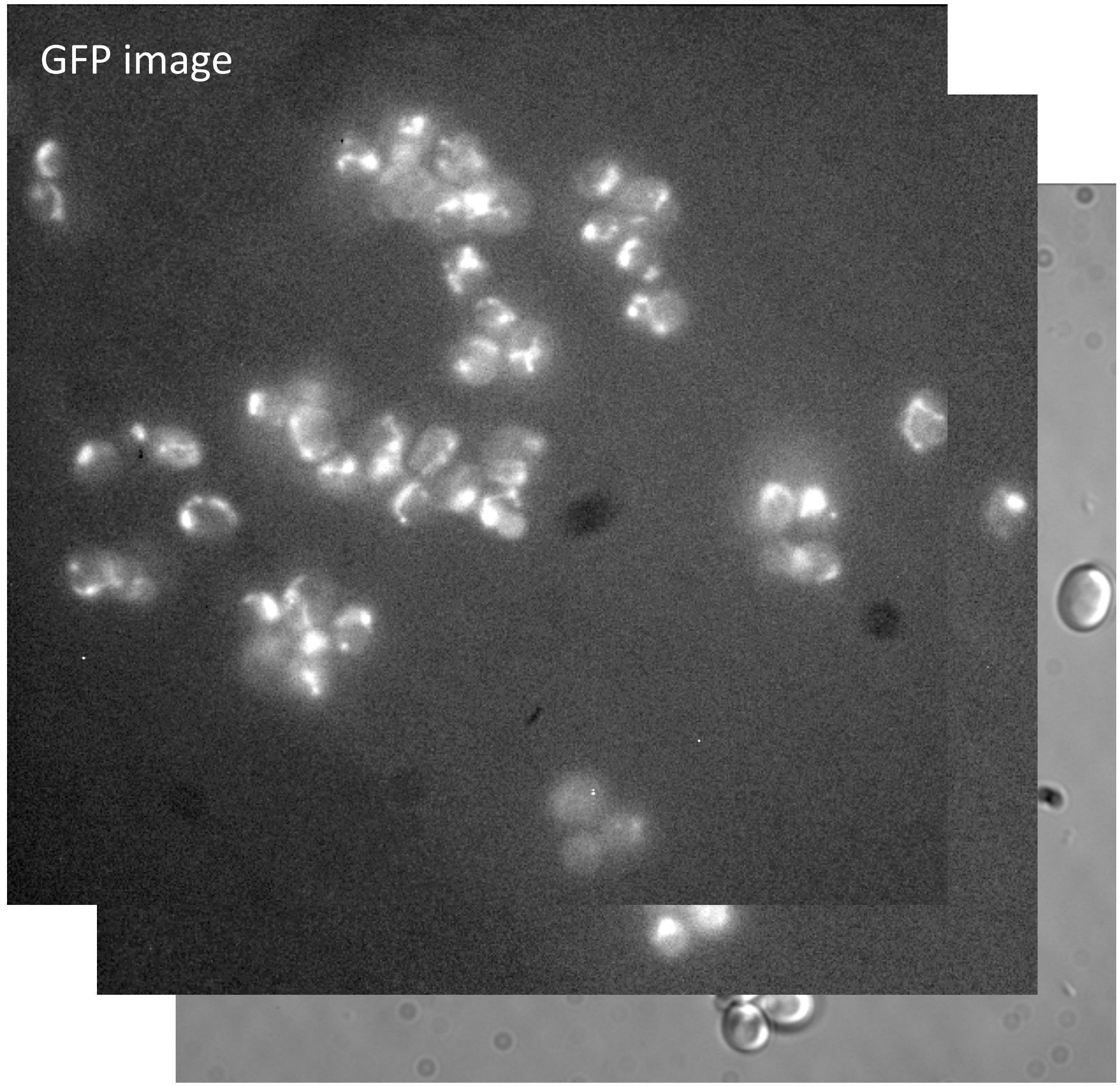
Yeast GFP Fusion Localization Database

- Contains images of 4156 proteins (out of 6234 ORFs in all 16 yeast chromosomes).
- GFP tagged immediately before the stop codon of each ORF to minimize perturbation of protein expression.
- Annotations were done manually by two scorers and co-localization experiments were done for some cases using mRFP.
- Each protein is assigned one or more of 22 location categories.

Content in the Yeast GFP Database

Example ORF

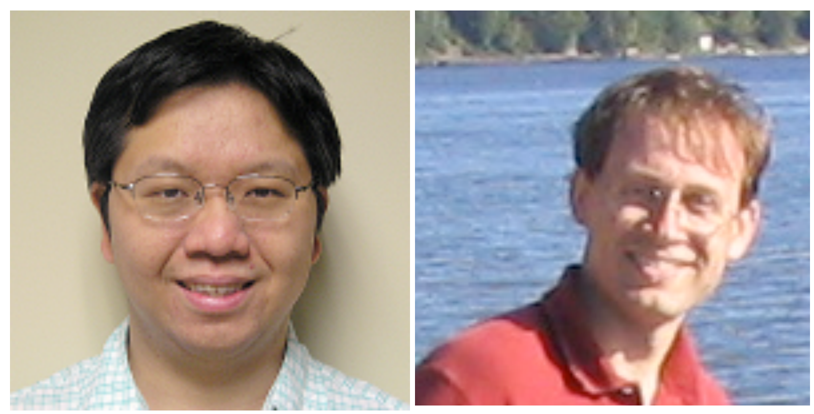
Name: YBR282W
Location: *mitochondrion*



Chen et al 2007

Classification of Yeast Subcellular Patterns

- Selected only those assigned to single unambiguous location class (21 classes)
- Trained classifier to recognize those classes
- **81% agreement with human classification**
- **94.5% agreement for high confidence assignments (without using colocalization!)**
- Examination of proteins for which methods disagree suggests machine classifier is correct in at least some cases



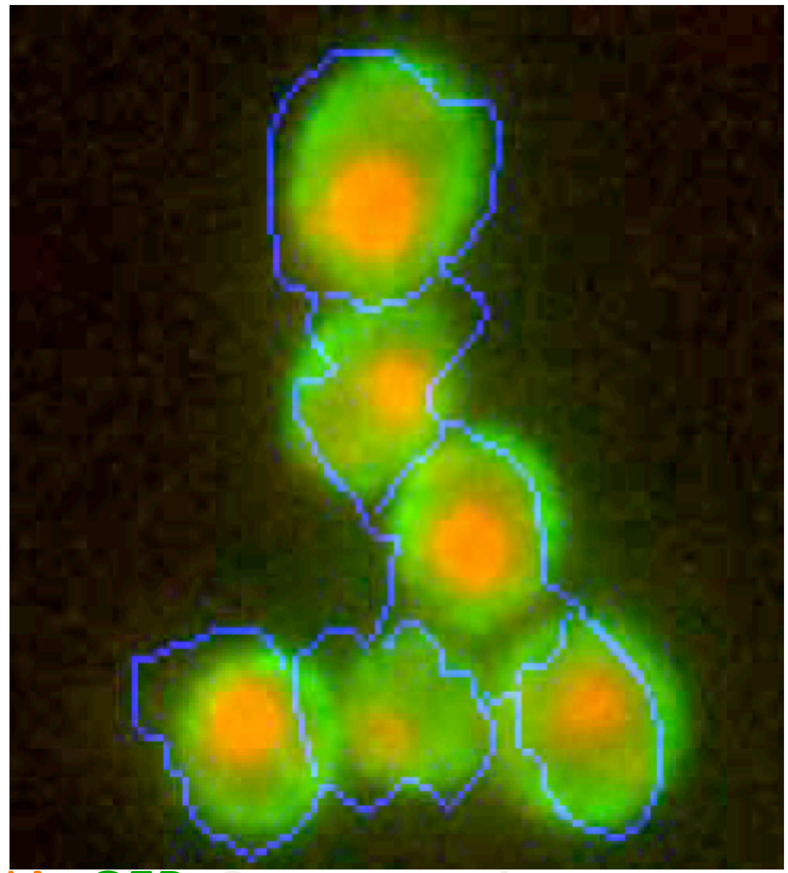
Shann-Ching (Sam) Chen & Geoff Gordon

Example of Potentially Incorrect Label

ORF Name
YGR130C

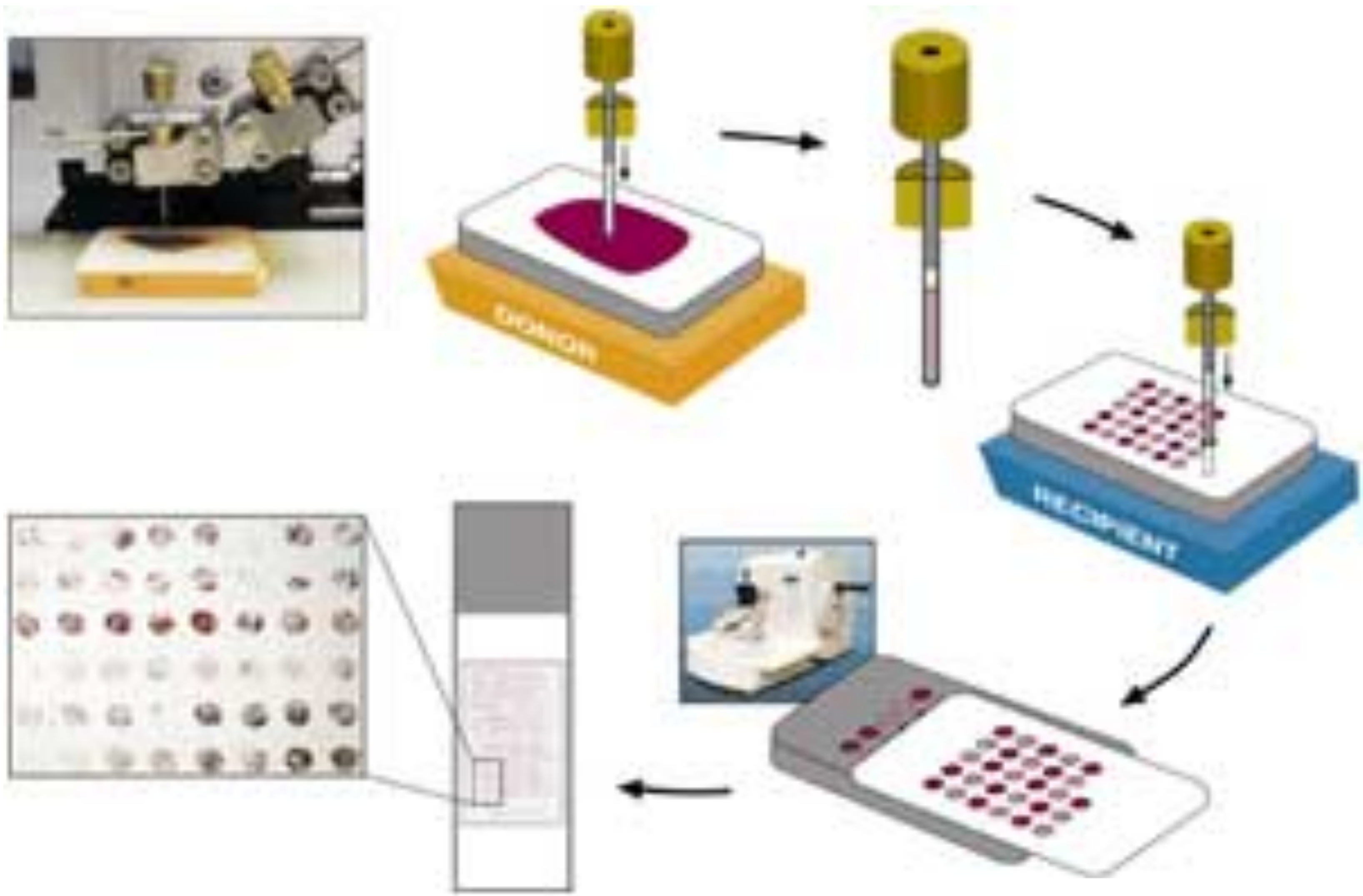
UCSF Location
punctate_composite

Automated Prediction
cell_periphery (60.67%)
cytoplasm (30%)
ER (9.33%)



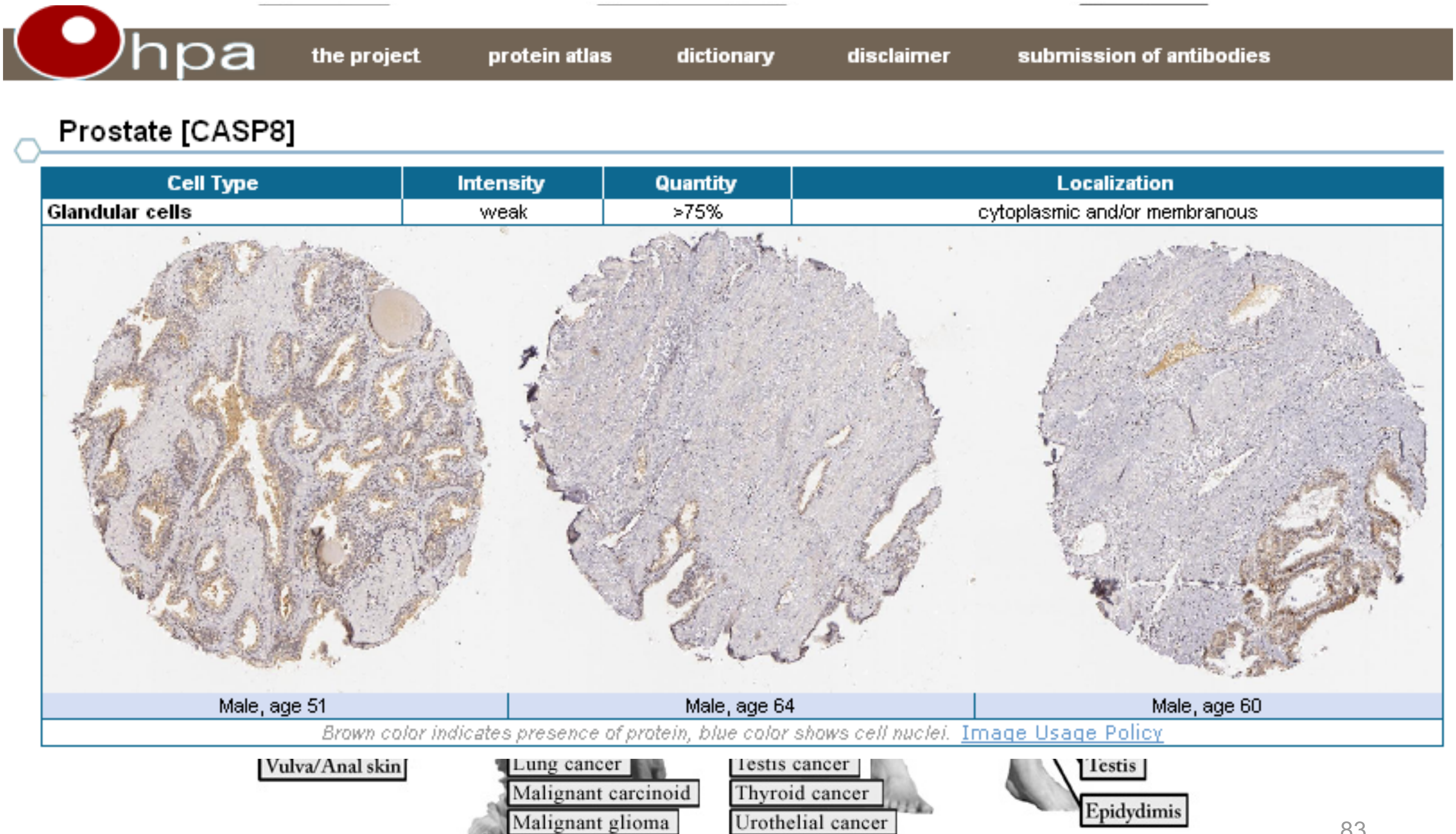
DNA GFP Segmentation

Tissue Microarrays

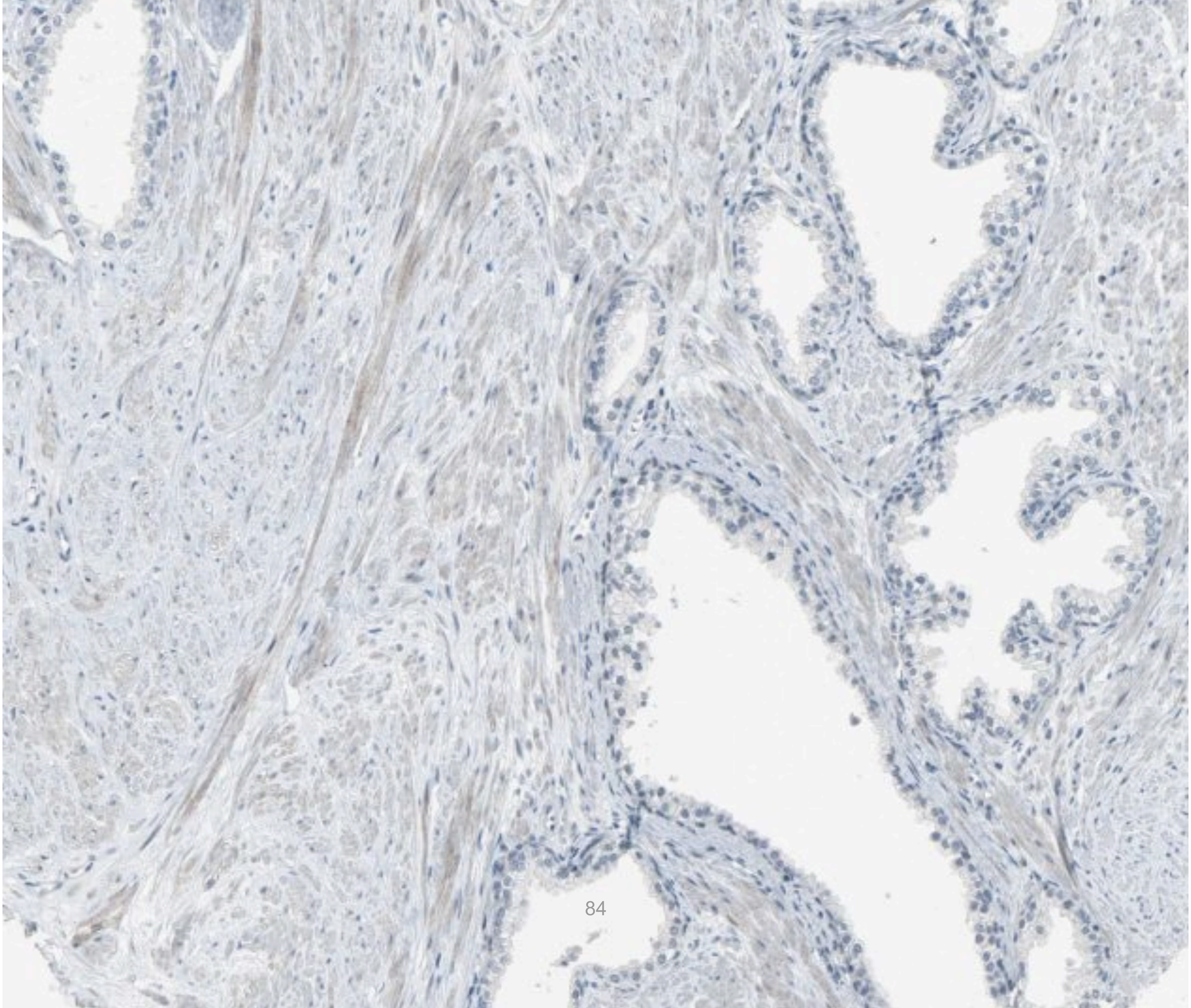


Courtesy www.microarraystation.com

Human Protein Atlas



Courtesy www.proteinatlas.org (Uhlén, Pontén et al)

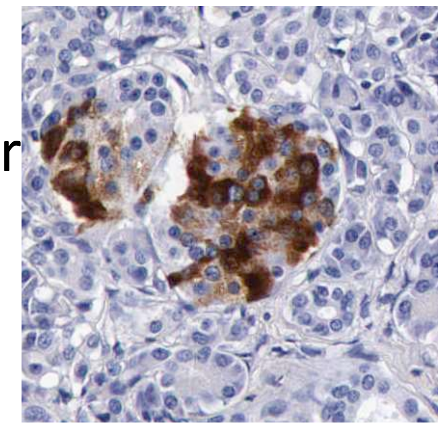


Immunocytochemistry Signal

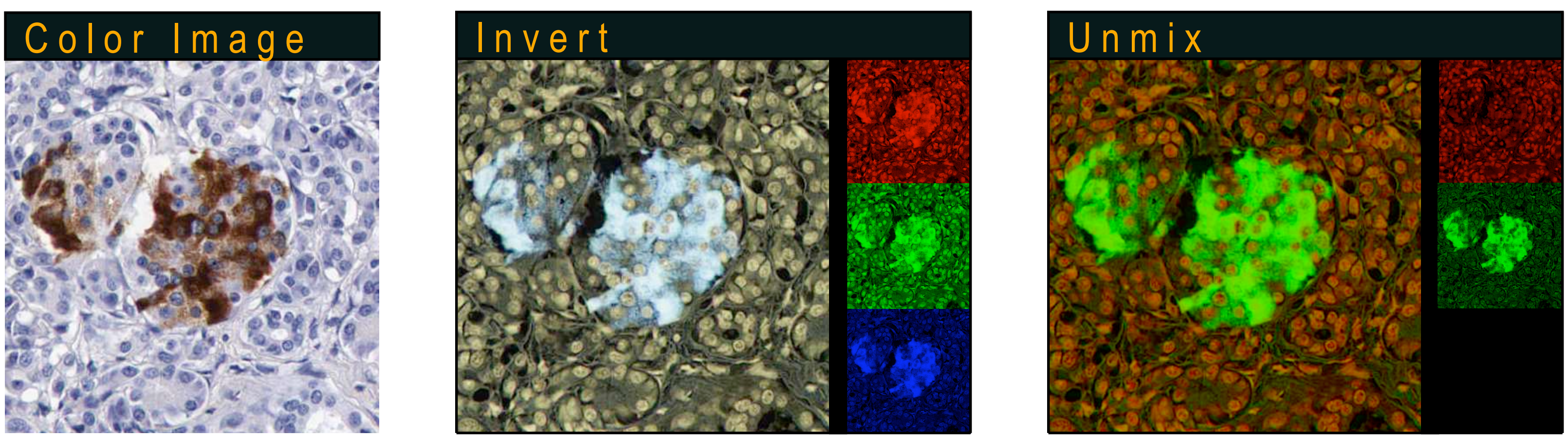
- The Haematoxylin and DAB stains are imaged together

- Each stain contains multiple color sour
  - Haem. = $n_1R + n_2G + n_3B$
  - DAB = $n_1R + n_2G + n_3B$

- Use linear unmixing to find w’s

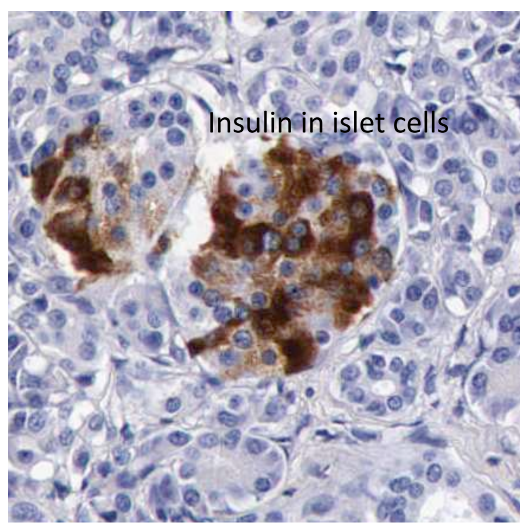


$$
\begin{array}{ccc}
R & G & B \\
\end{array}
$$
$$
\begin{pmatrix}
28 & 22 & 36 \\
28 & 23 & 29 \\
28 & 23 & 55 \\
28 & 24 & 21 \\
28 & 24 & 23 \\
28 & 24 & 25 \\
28 & 25 & 18 \\
28 & 25 & 20 \\
28 & 34 & 66 \\
28 & 44 & 34 \\
28 & 48 & 75
\end{pmatrix}
$$



Test Dataset from Human Protein Atlas

- Selected 16 proteins from the Atlas
- Two each from all major organelles (class)
- ~45 tissue types for each class (e.g. liver, skin)
- Goal: Train classifier to recognize each subcellular pattern across all tissue types



Justin Newberg

Newberg & Murphy, 2008

Subcellular Pattern Classification over 45 tissues

Prediction

	ER	Cyto	Endo	Golgi	Lyso	Mito	Nucleolus	Nucleus
ER (53)	**100**	0	0	0	0	0	0	0
Cyto (21)	4.8	**76.2**	0	0	14.3	4.8	0	0
Endo (2)	0	0	**100**	0	0	0	0	0
Golgi (88)	1.1	0	0	**98.9**	0	0	0	0
Lyso (52)	0	1.9	0	0	**96.2**	0	1.9	0
Mito (64)	0	0	0	0	0	**98.4**	1.6	0
Nucleolus (94)	0	0	0	2.1	2.1	1.1	**94.7**	0
Nucleus (78)	0	0	0	0	0	0	0	**100**

Accuracy for 50% of images with highest confidence: 97%

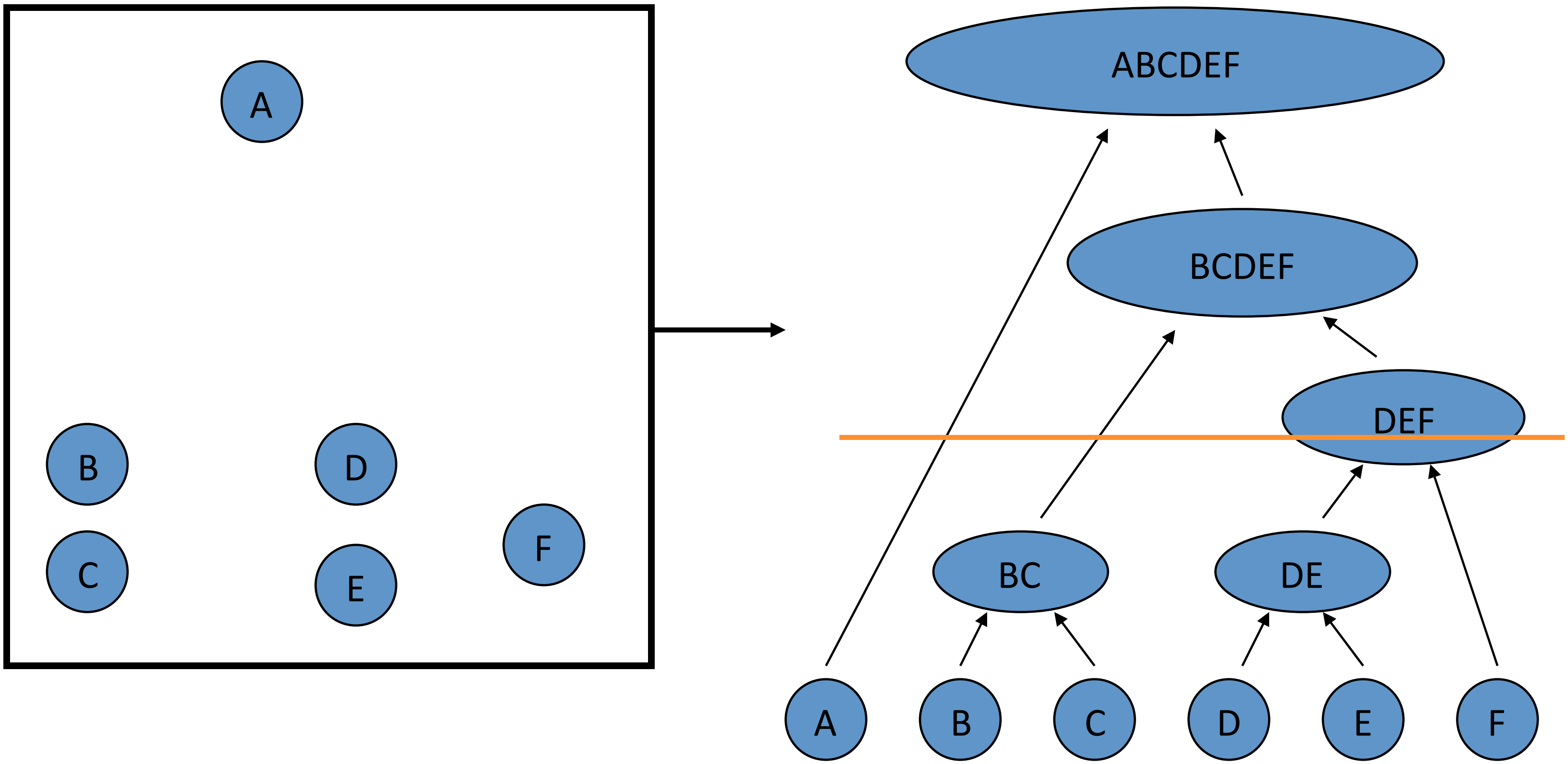
Supervised vs. Unsupervised Learning

- Feasibility of using classification methods to assign all proteins to known major classes well demonstrated
- Do we know all locations? Are assignments to major classes enough?
- Need approach to discover classes – cluster proteins into subcellular location families

Hierarchical vs. *k*-means clustering

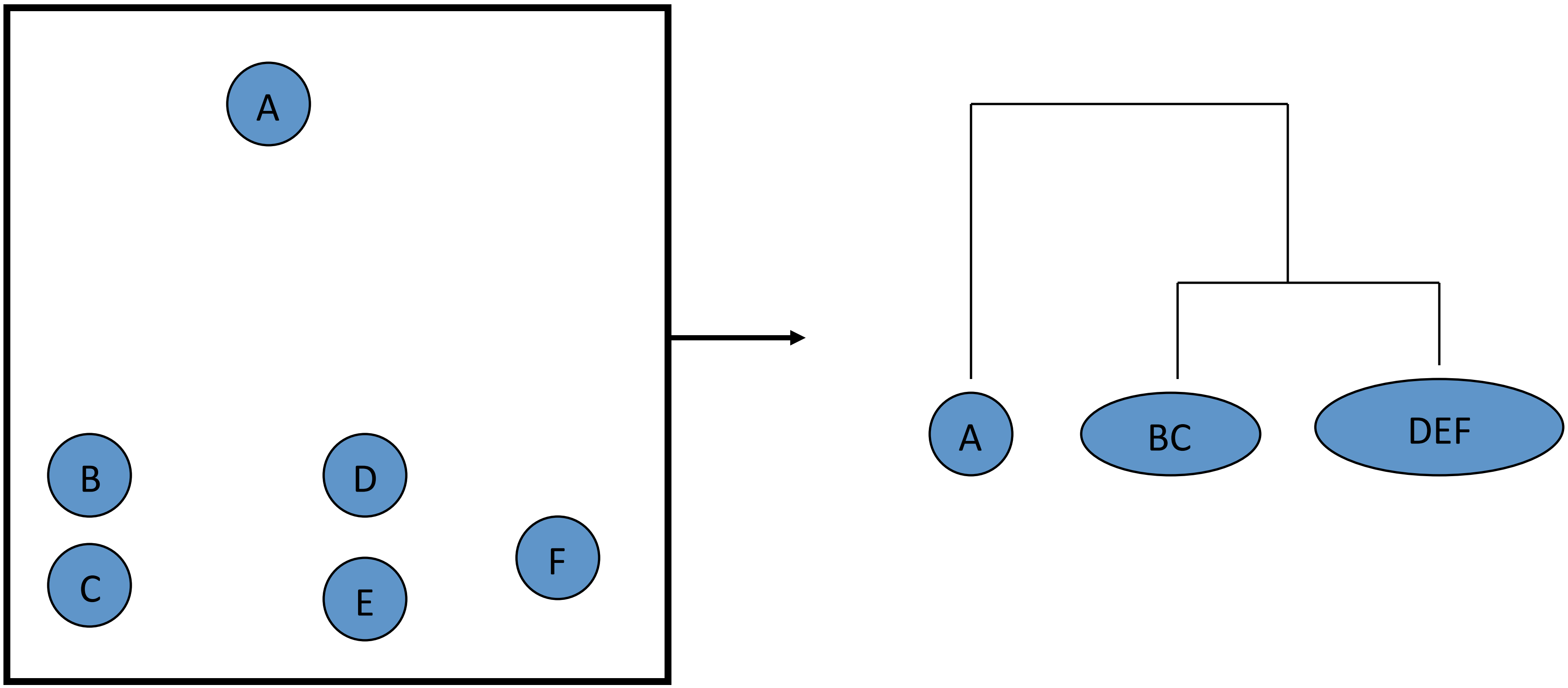
- Two most popular clustering algorithms
- Hierarchical builds tree sequentially from the closest pair of points (wells/cells/probes/conditions)
- *k*-means starts with *k* randomly chosen seed points, assigns each remaining point to the nearest seed, and repeats this until no point moves

Hierarchical Clustering



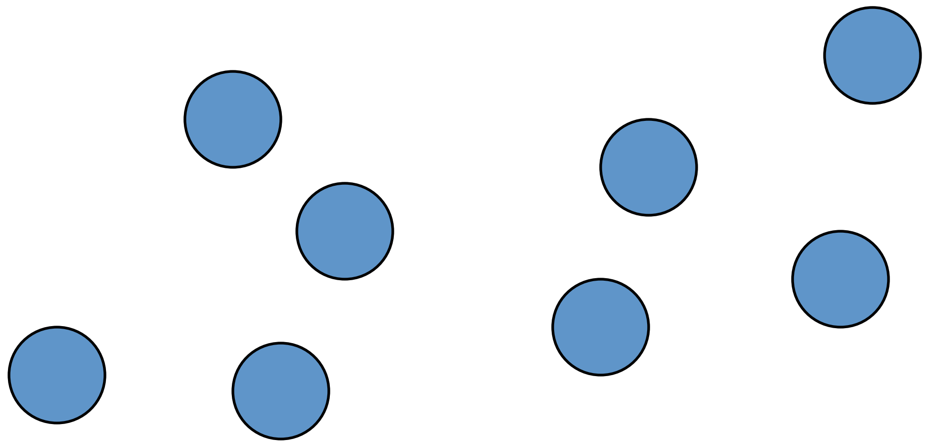
Slide courtesy of Elvira Garcia Osuna

Hierarchical Clustering



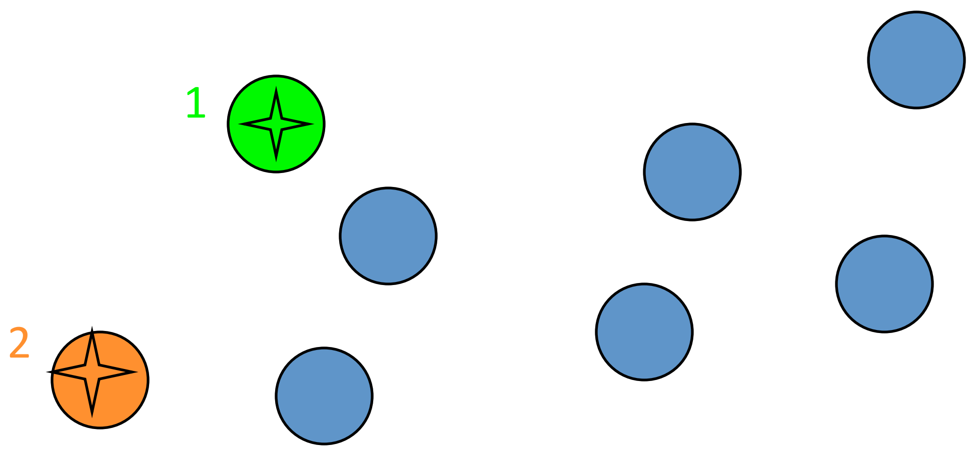
Slide courtesy of Elvira Garcia Osuna

K-means



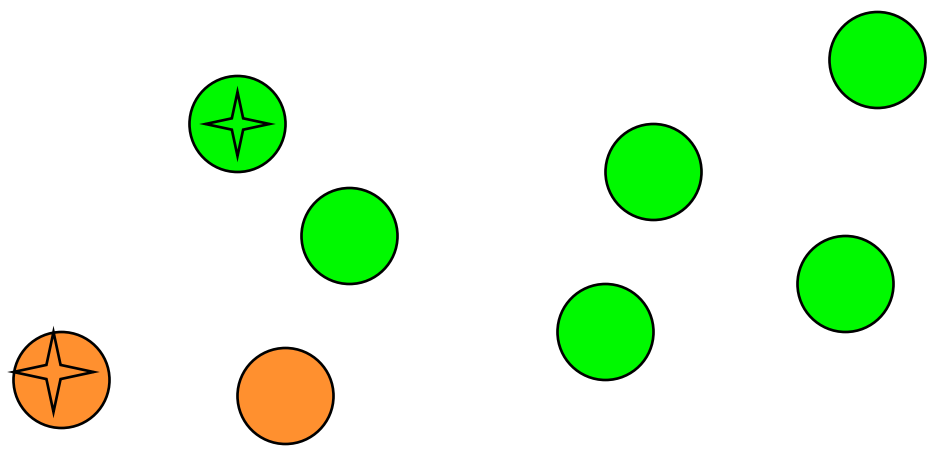
Slide courtesy of Elvira Garcia Osuna

K-means



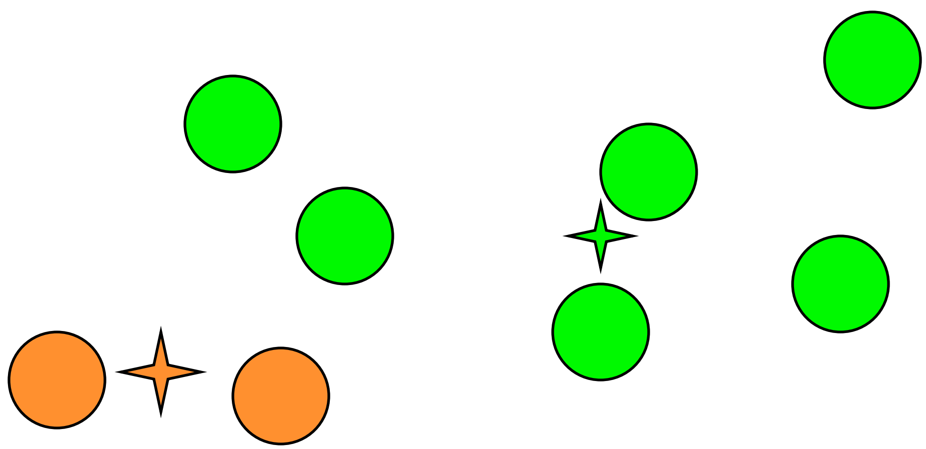
Slide courtesy of Elvira Garcia Osuna

K-means



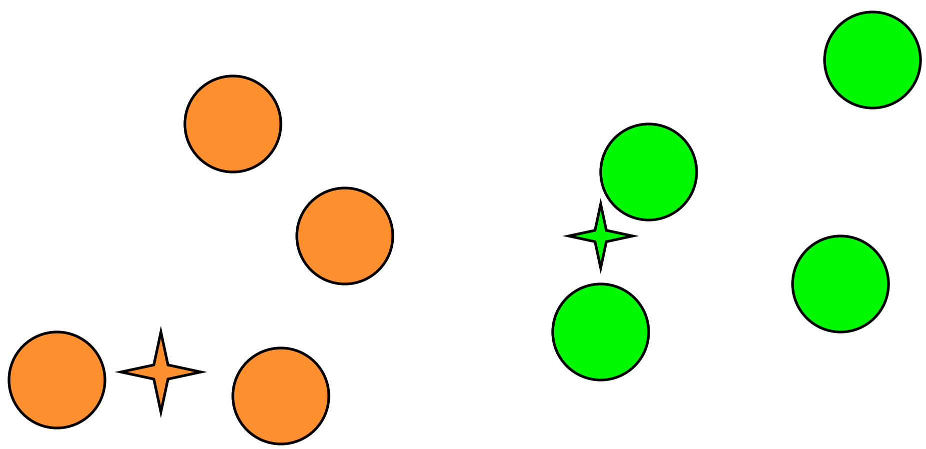
Slide courtesy of Elvira Garcia Osuna

K-means



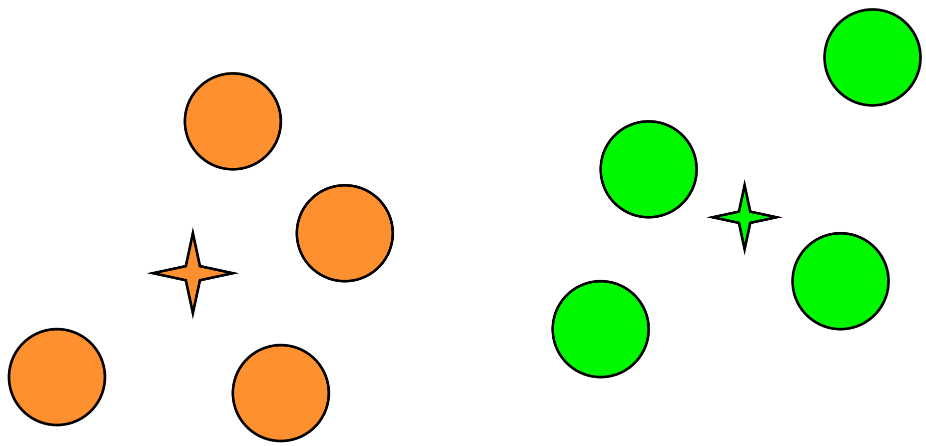
Slide courtesy of Elvira Garcia Osuna

K-means



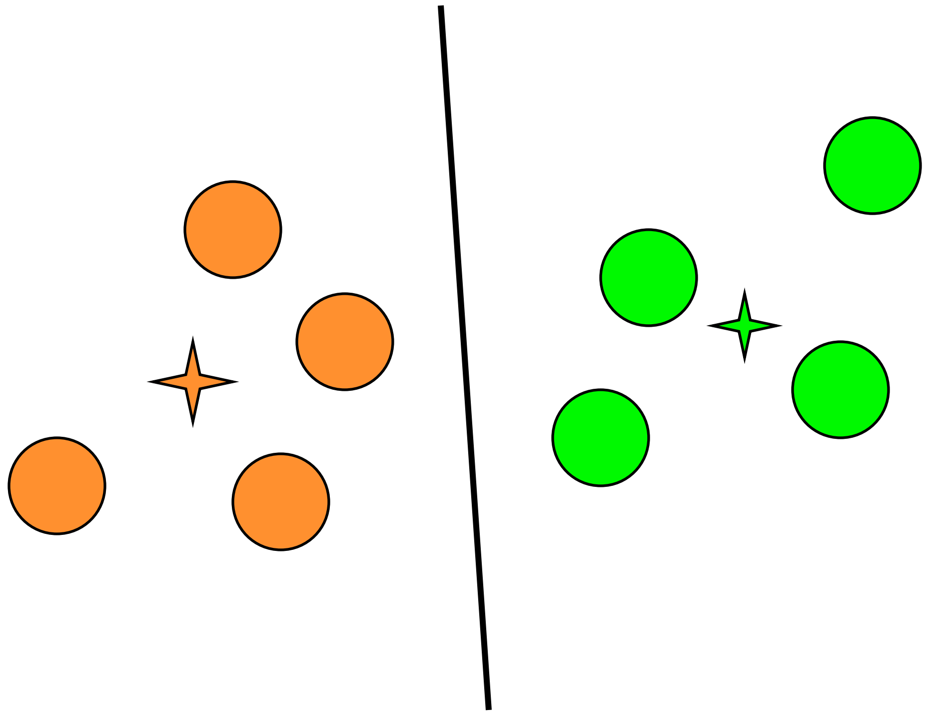
Slide courtesy of Elvira Garcia Osuna

K-means



Slide courtesy of Elvira Garcia Osuna

K-means



Slide courtesy of Elvira Garcia Osuna

Choosing the number of Clusters

- A difficult problem
- Most common approach is to try to find the solution that minimizes the Bayesian Information Criterion

$$BIC = -2\ln L + k\ln(n)$$

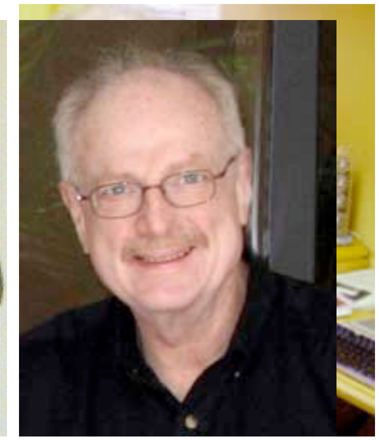
L = the likelihood function for the estimated model

K = # of parameters

n = # of samples

Location Proteomics: Randtag project

- **Tag** many proteins
  - use **CD-tagging**
  (developed by Jonathan Jarvik and Peter Berget): Infect population of cells with a retrovirus carrying DNA sequence that will "tag" in a random gene in e



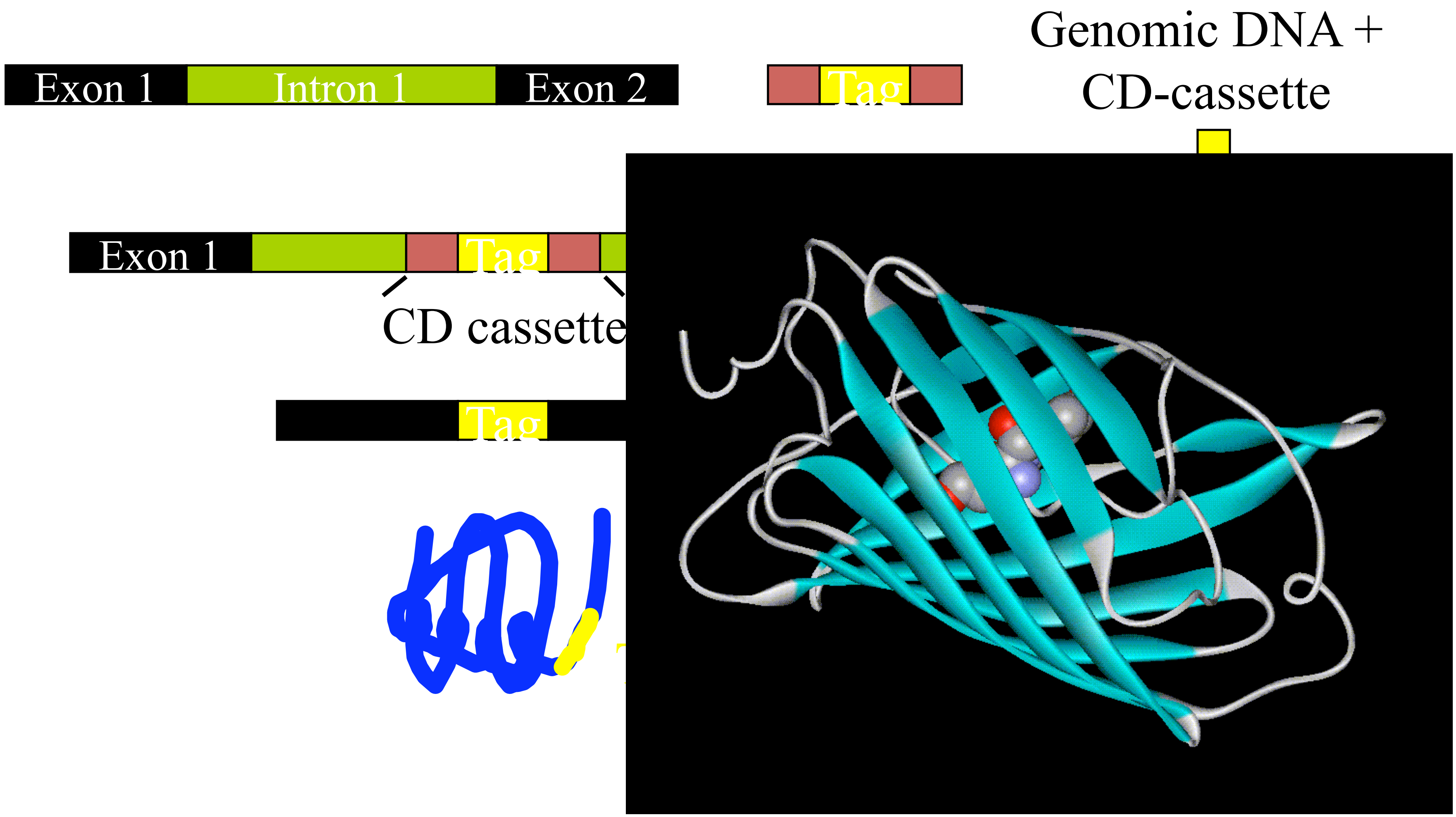
Jarvik et al 2002

Isolate separate **clones**, each of which produces express one tagged protein

Use RT-PCR to **identify tagged gene** in each clone

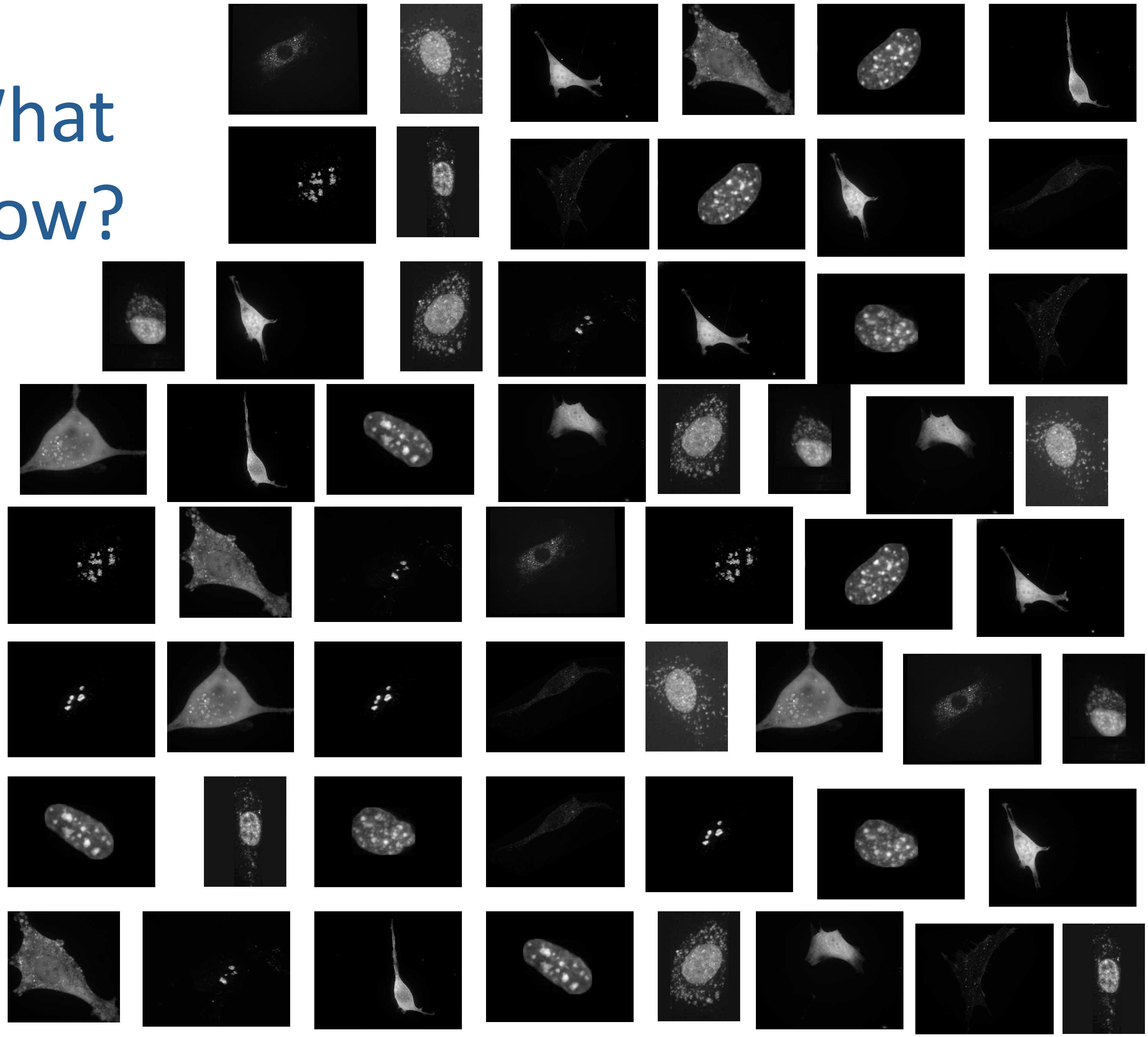
- Collect **many live cell images** for each clone using spinning disk confocal fluorescence microscopy

CD-Tagging Principle (CD=central dogma)

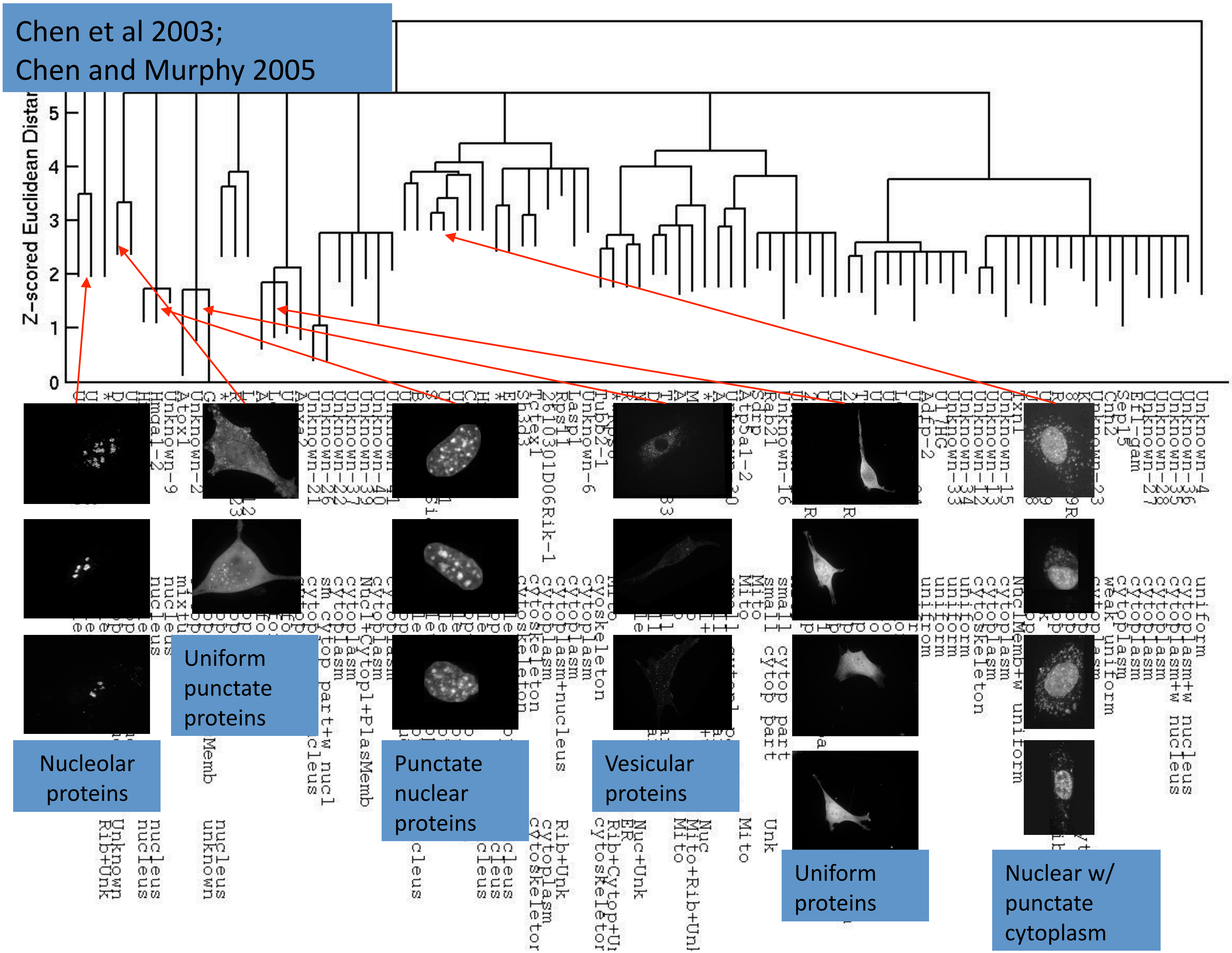


What Now?

Group ~90 tagged clones by pattern



Chen et al 2003;
Chen and Murphy 2005



Pattern unmixing

- Some proteins may be found in more than one organelle
- Clustering sees each combination of organelles as a new pattern
- Can we "unmix" such mixed patterns?

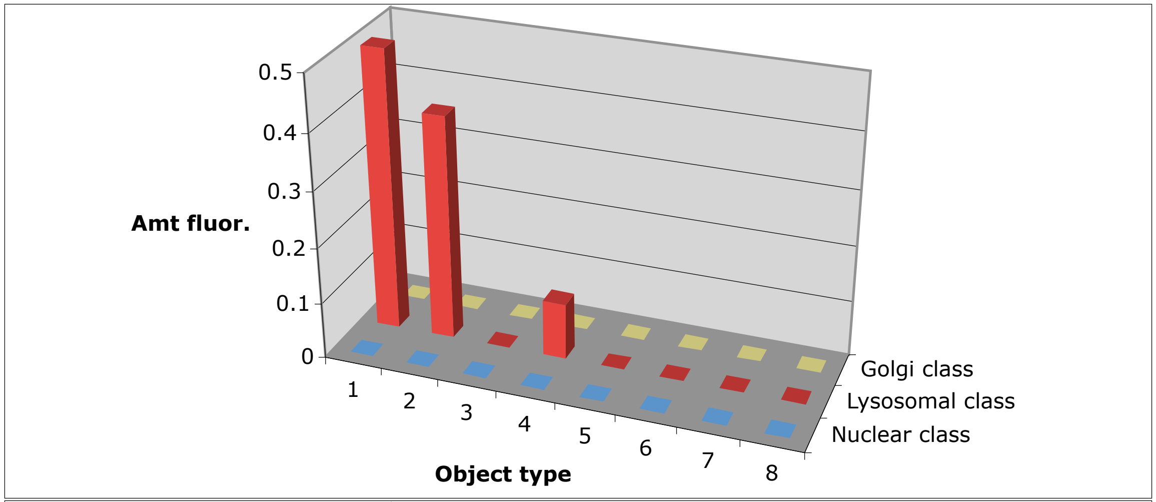
Unmixing approach

- Assume that each fundamental subcellular pattern can be represented by some combination of distinct object types (10% small round objects and 90% long skinny objects)
- Assume that a mixed pattern is formed by adding together the objects from two or more fundamental patterns *and that no new object types are created*

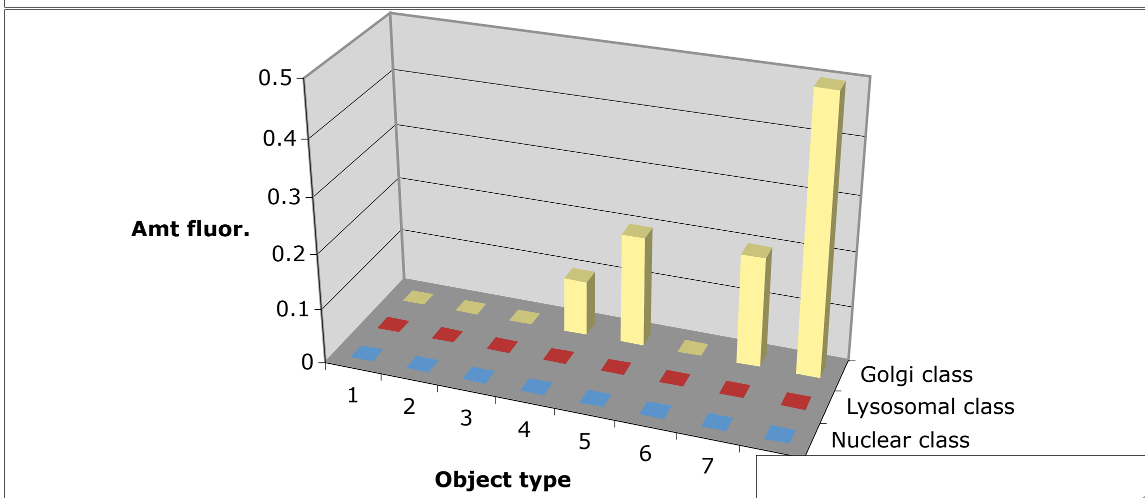
Zhao et al 2005

Learning object types

- Find all objects in all images of fundamental types
- Describe each object by features such as size, ellipticity, distance from nucleus
- Cluster objects to find types
- Represent each fundamental pattern as probabilities of observing each object type

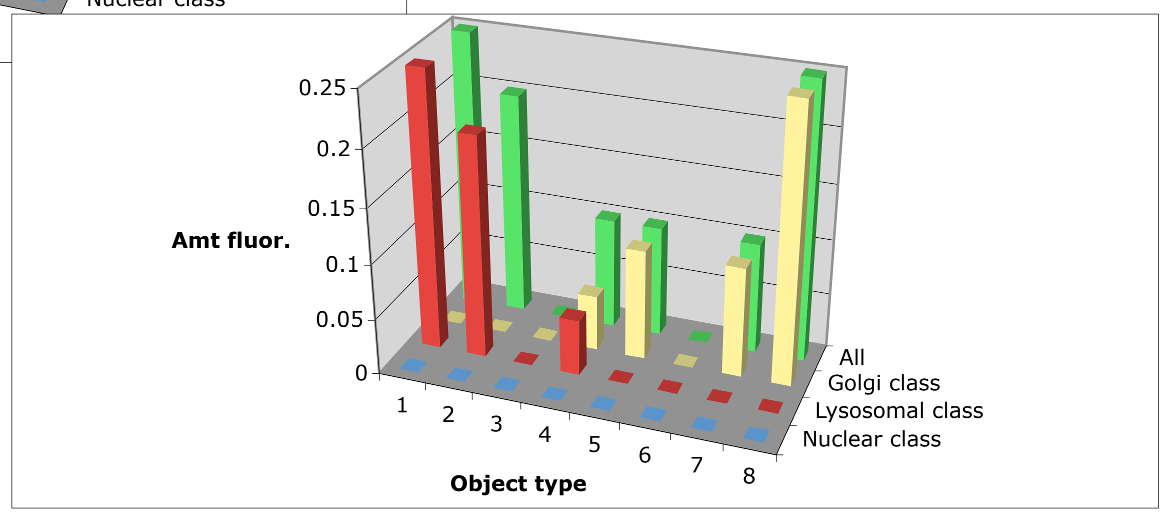


Pure Lysosomal Pattern



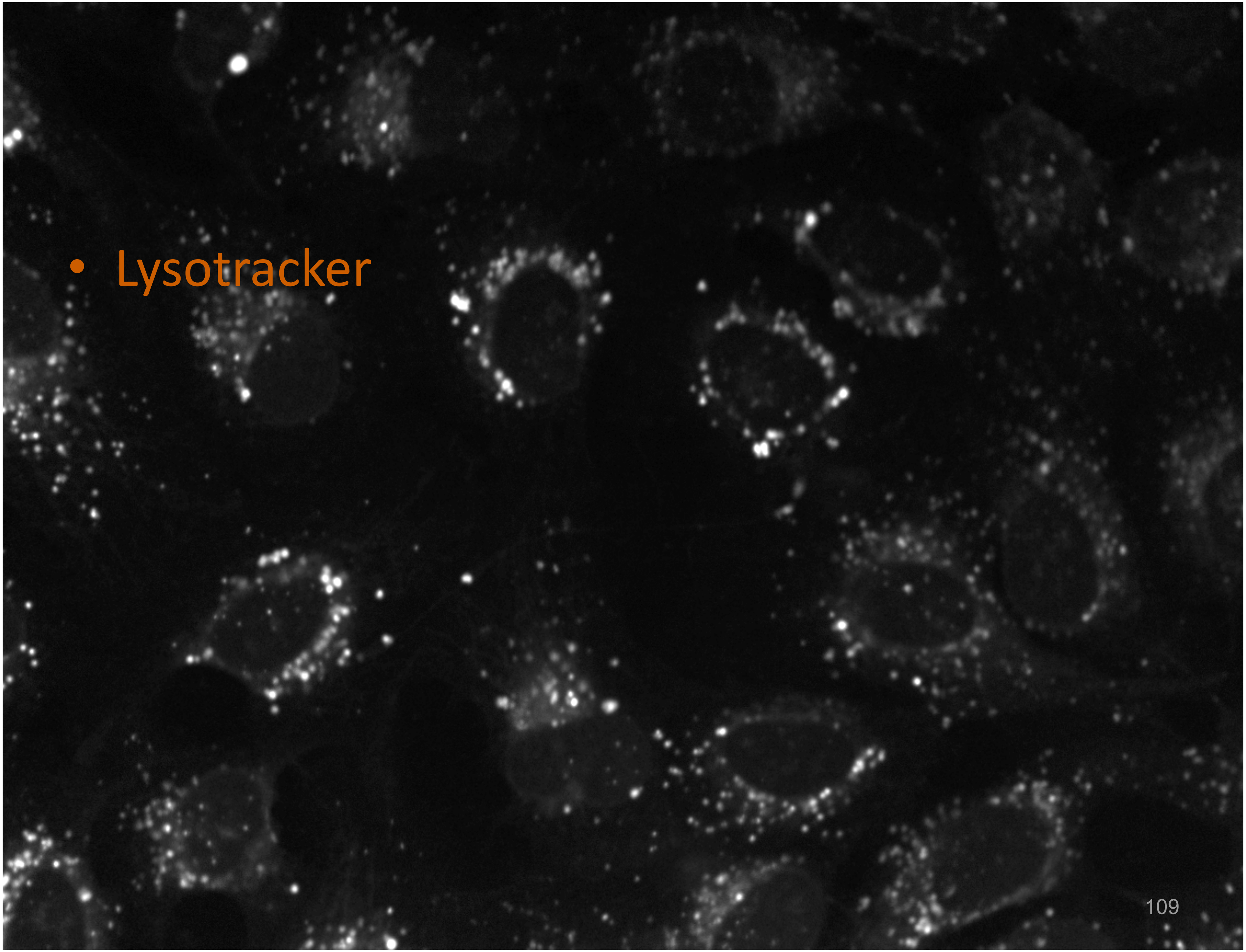
Pure Golgi Pattern

50% mix of each

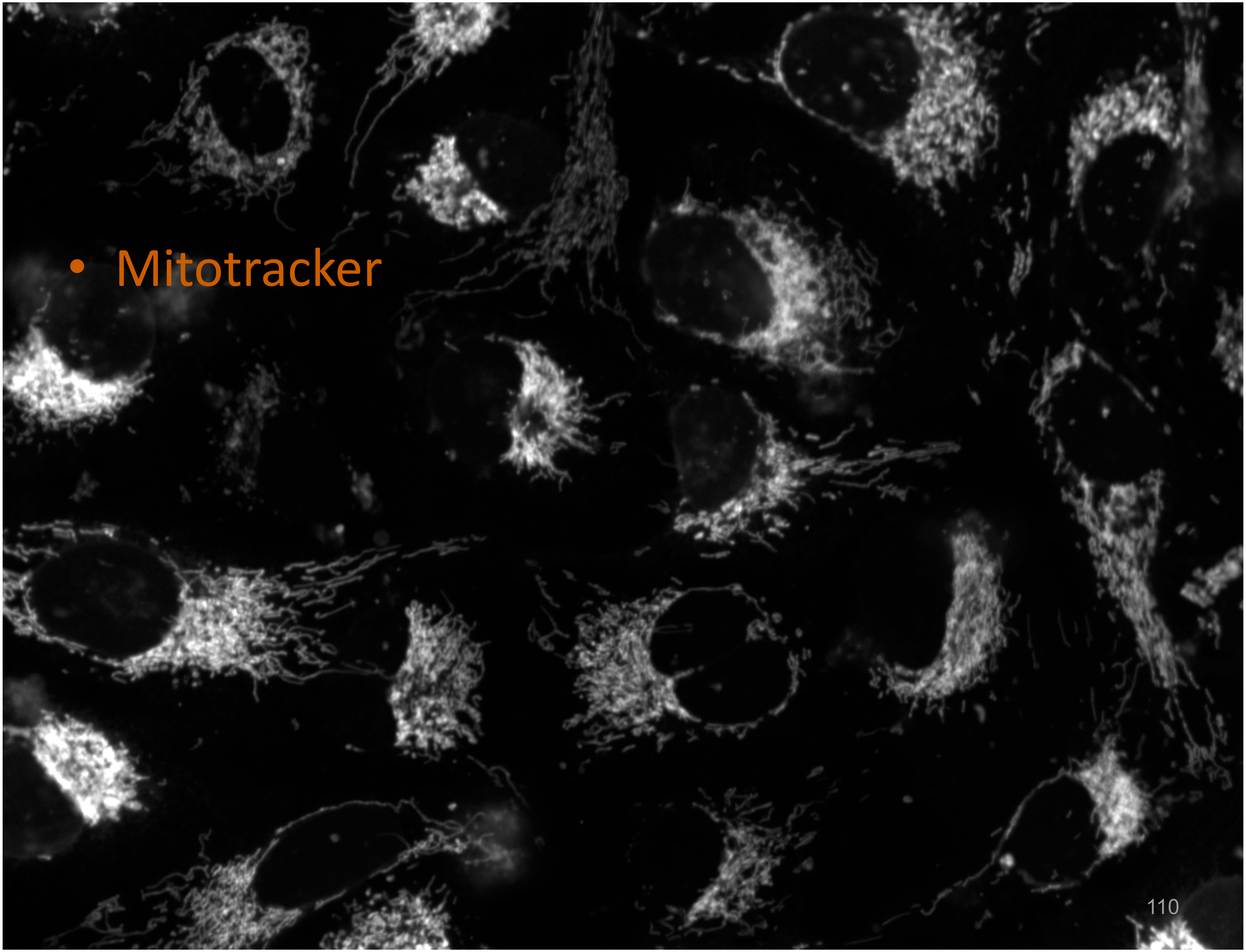


Test samples

- 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)

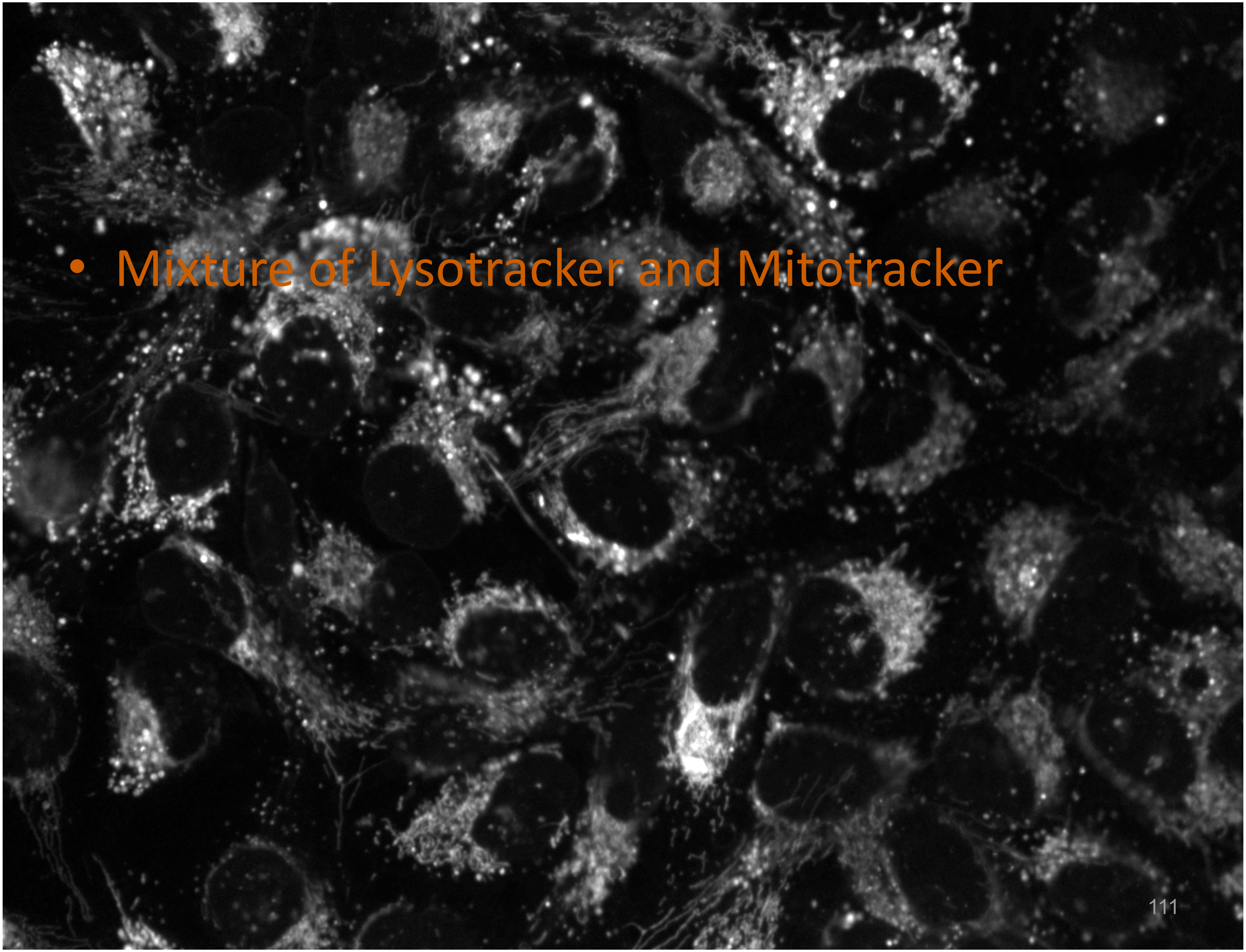


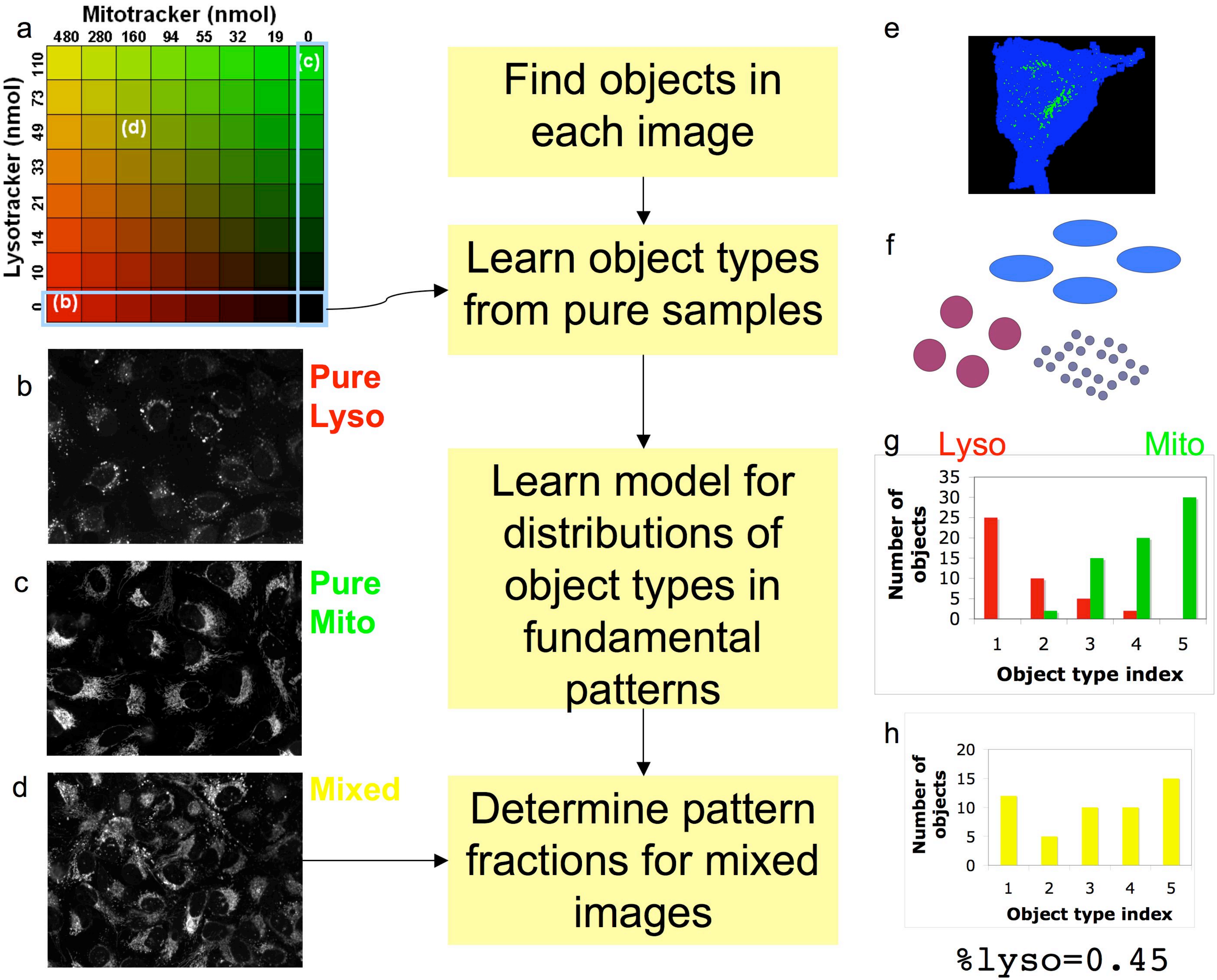
- Lysotracker



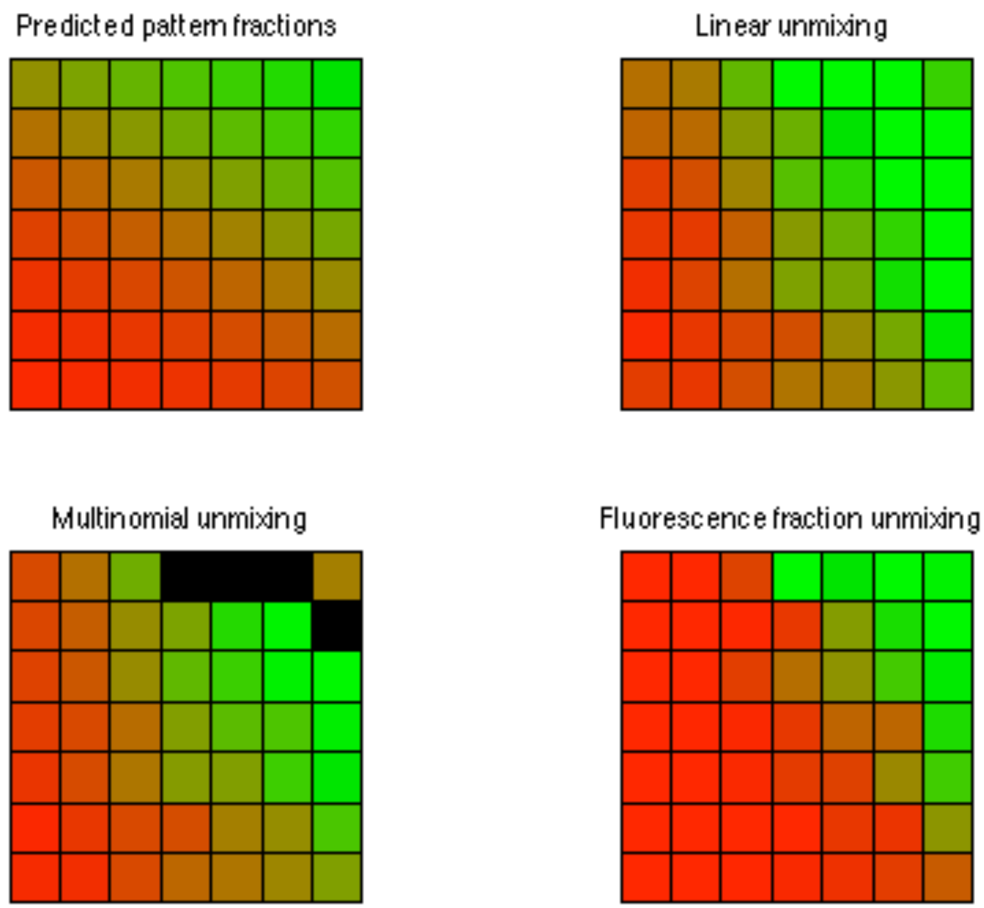
- Mitotracker

- Mixture of Lysotracker and Mitotracker




a
Mitotracker (nmol)
480 280 160 94 55 32 19 0
Lysotracker (nmol)
110 73 49 33 21 14 10 0
(c)
(d)
(b)
b
Pure Lyso
c
Pure Mito
d
Mixed
Find objects in each image
Learn object types from pure samples
Learn model for distributions of object types in fundamental patterns
Determine pattern fractions for mixed images
e
f
g
Lyso
Mito
Number of objects
35 30 25 20 15 10 5 0
1 2 3 4 5
Object type index
h
Number of objects
20 15 10 5 0
1 2 3 4 5
Object type index
%lyso=0.45

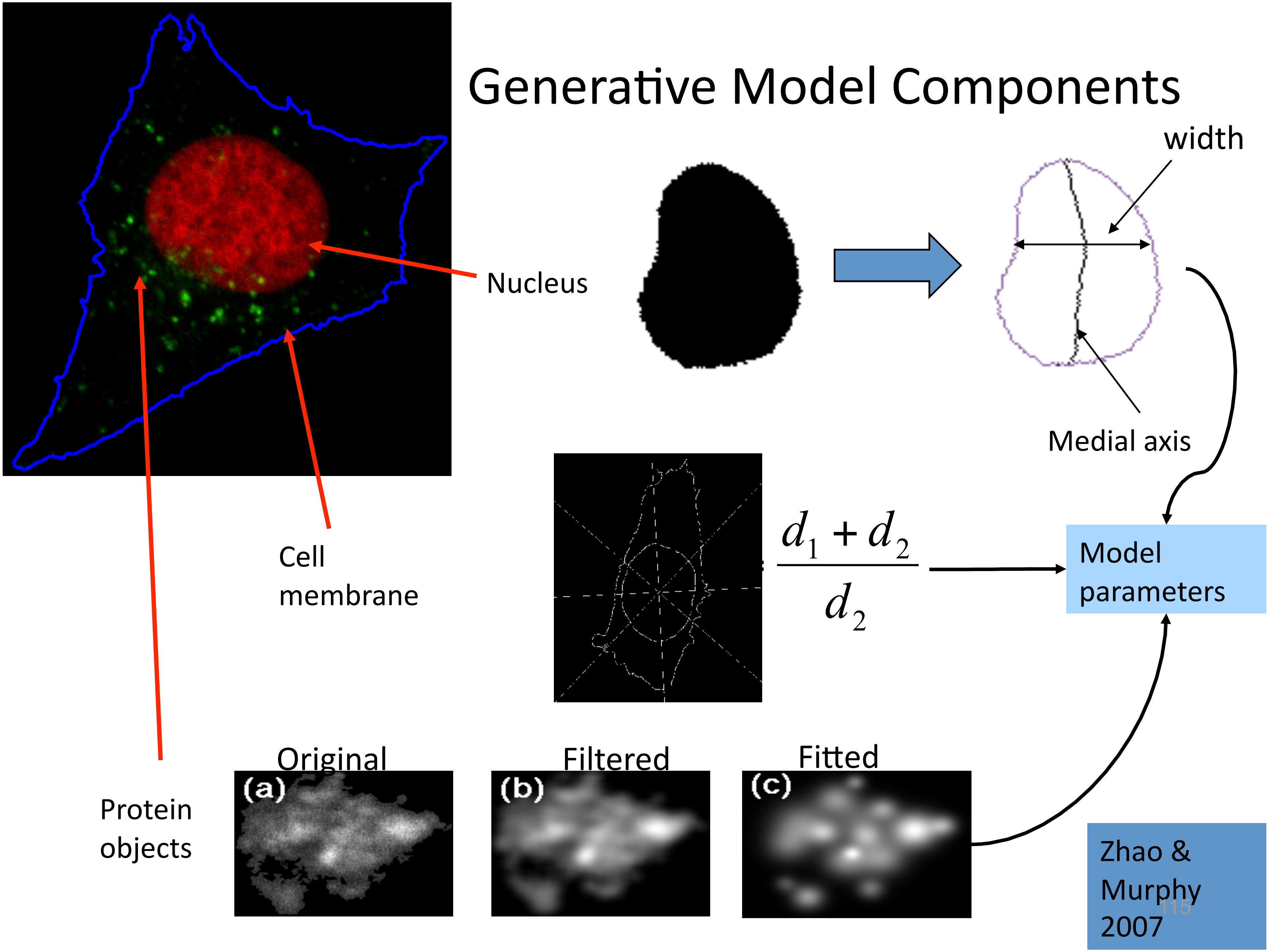
Pattern unmixing results



Communicating patterns

- How do we communicate results learned about subcellular patterns?
- Proposal: Use generative models learned from images to capture pattern and *variation* in pattern

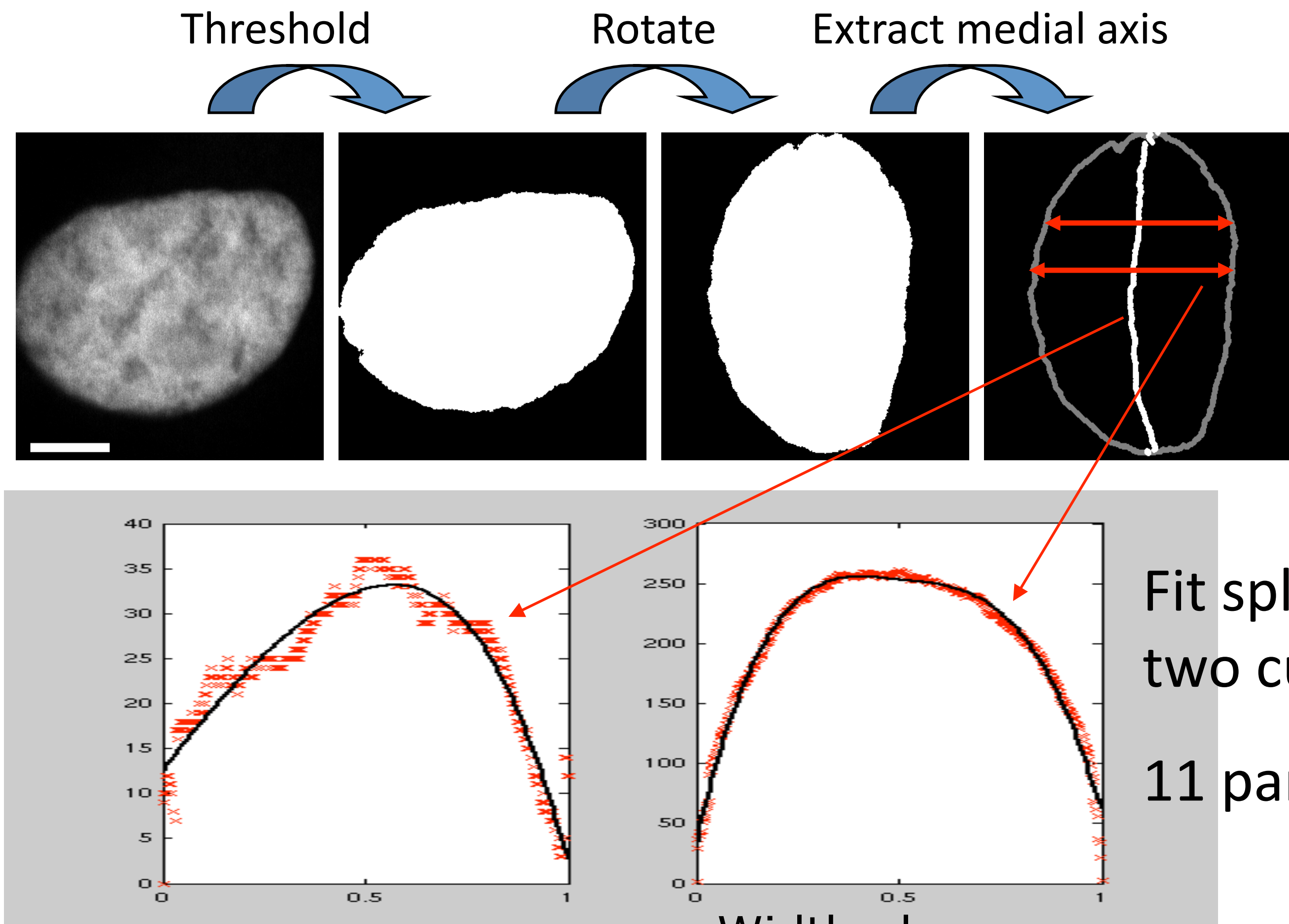
Generative Model Components



Nuclear shape models

- Modified medial axis model
- Diffeomorphic model
  - S. Yang, D. Köhler, K. Teller, T, Cremer, P. Le Baccon, E. Heard, R. Eils, and K. Rohr, MICCAI 2006, LNCS 4190, pp. 907–914, 2006

Nuclear Shape - Medial Axis Model



Medial axis

Width along the medial axis

Fit splines to two curves :

11 parameters

Shape generation

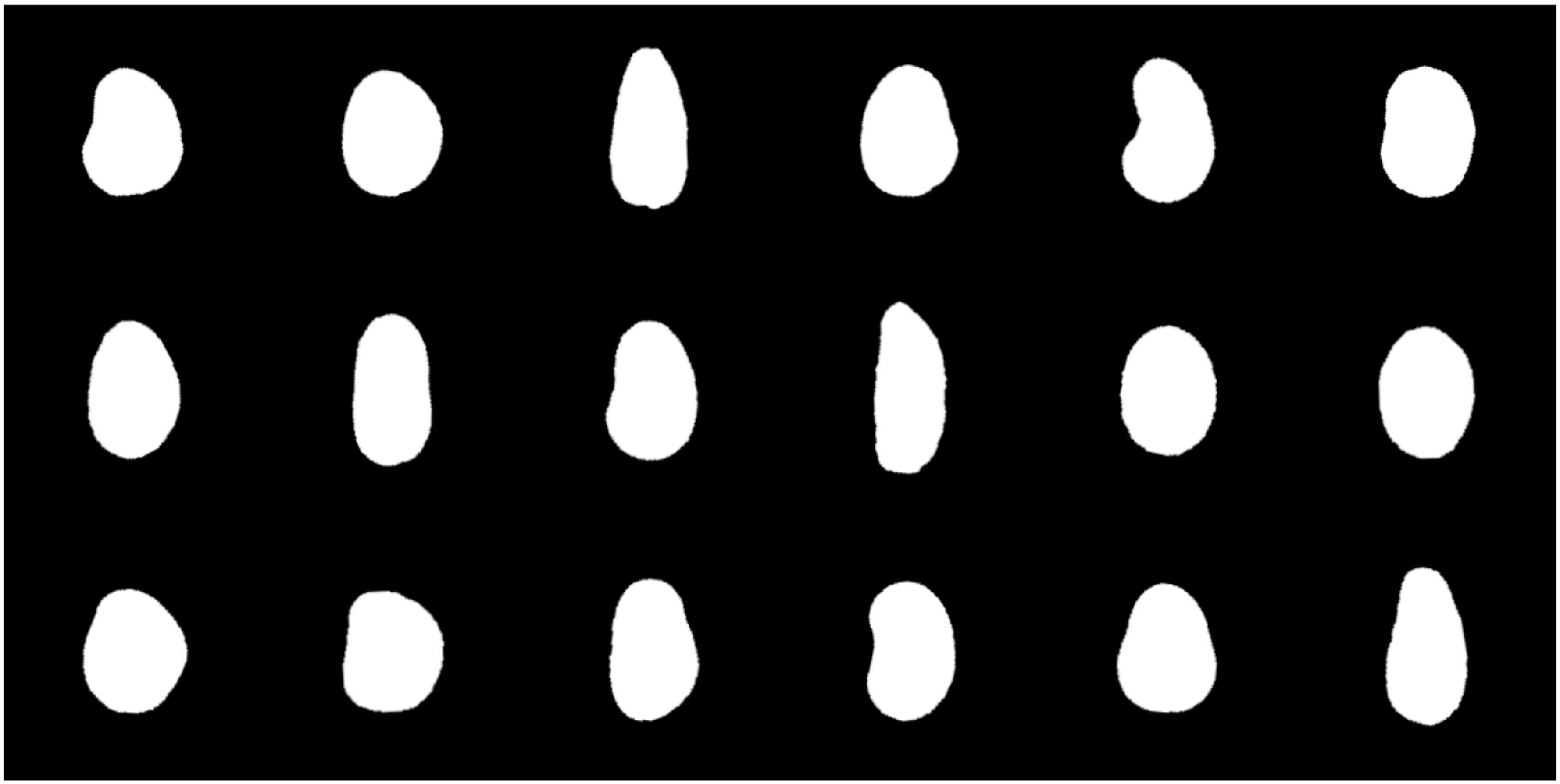
- 11 parameters for each object
  - 5 parameters for each curve
  - the length of the medial axis
- Learn the distribution of parameters over many nuclei
  - Assume multivariate normal
- Randomly sample parameters from distribution
- Construct nuclear shape using the sampled parameters

Synthesized nuclear shapes

Diffeomorphic analysis of nuclear shape

- Can use distance between shapes to characterize shape space instead of parameters of model – Gustavo Rohde

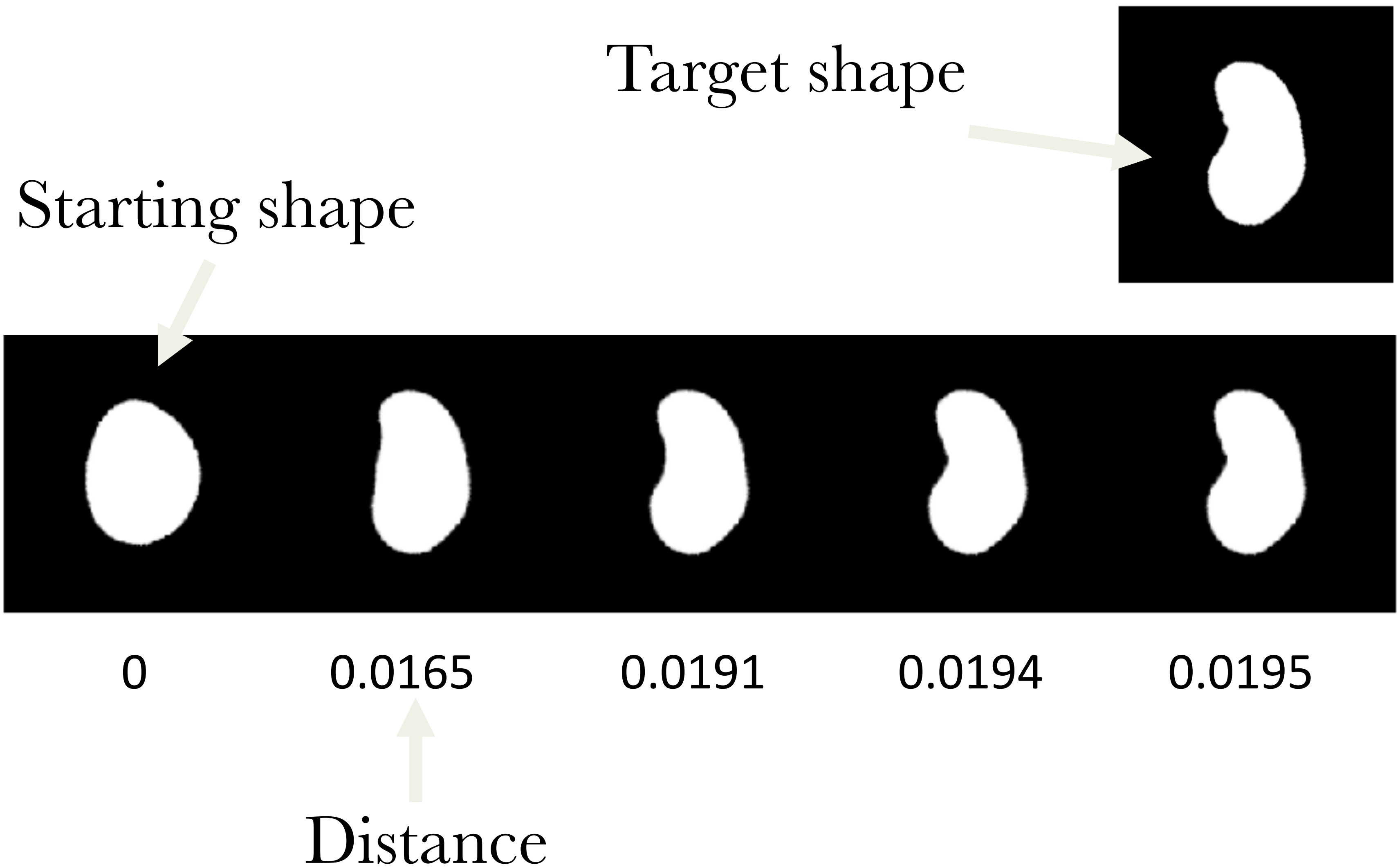
Concept: measure distances between all examples as means of characterizing shape space



Finding deformation field

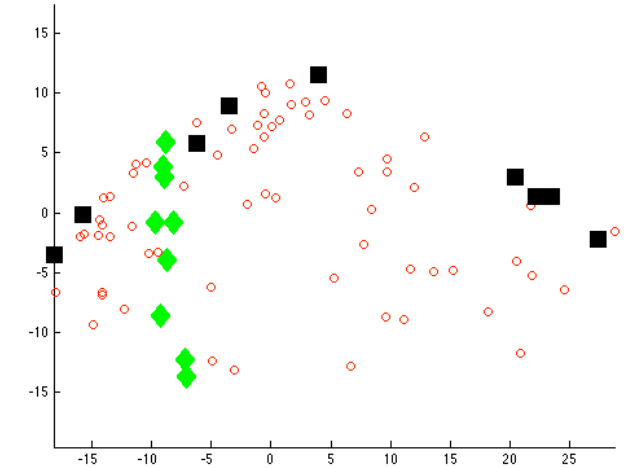
- Goal: Find a function g(x,t) which smoothly transforms an image I_n into an image I_m as t goes from 0 to T
- Choose g(x,t) to minimize sum of
  - Total deformation in g from 0 to T
  - Distance between I_m and $I_n(g(x,T))$

Mapping two shapes to each other

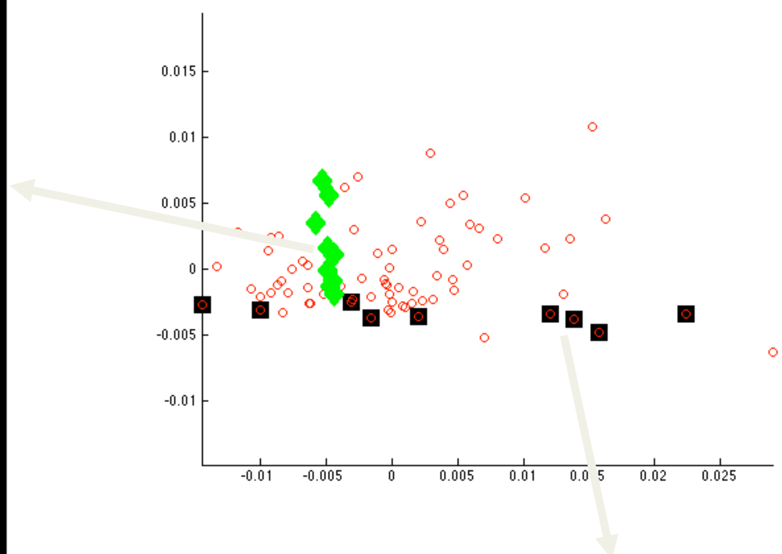

Target shape
Starting shape
0
0.0165
0.0191
0.0194
0.0195
Distance

Characterizing shape space

- Find deformation fields from each image to every other image
- Calculate distance between each pair of images as total deformation required between them
- Use multidimensional scaling (MDS) to find variables (principal components) that compactly represent variation



First 2 components from MDS directly on perimeter coordinates

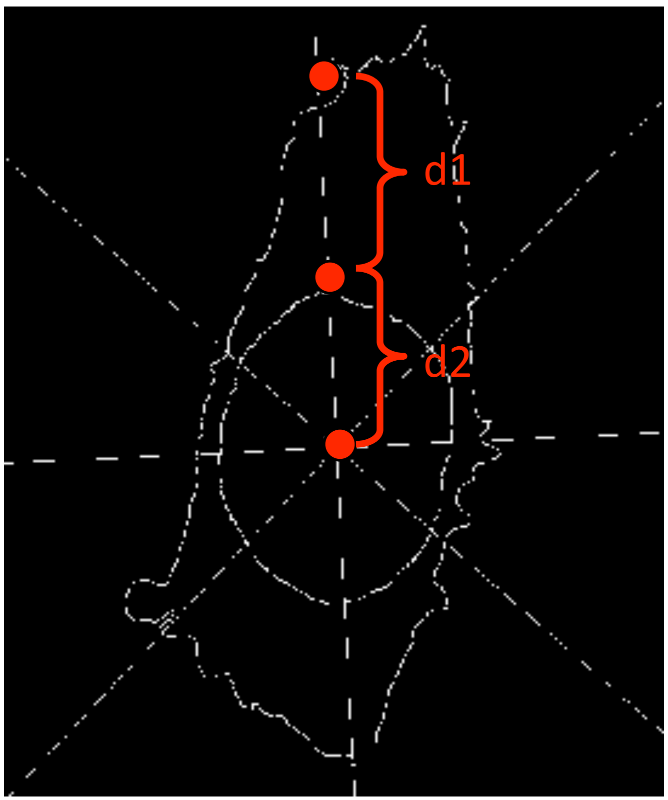


First 2 components from MDS on distance matrix from LDDMM

Cell shape models

- Conditional radial distance ratio model
- Diffeomorphic model (in progress)

Cell Shape
Description: Distance Ratio



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

Represent single shape as vector of ratios for *n* angles and represent variation using PCA

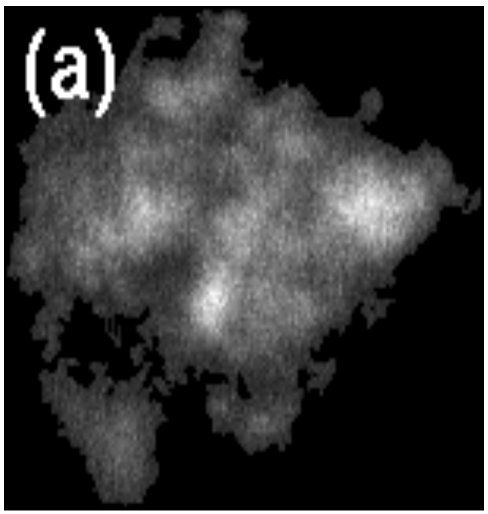
Diffeomorphic analysis of cell shape

Models for protein-containing objects

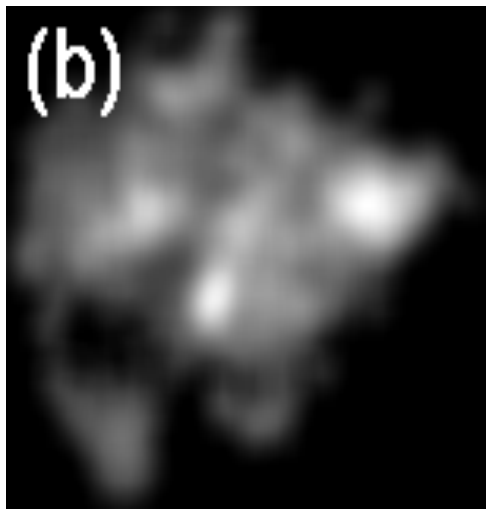
- Object library
- Gaussian objects
  - Mixture of Gaussians with number of objects determined from number of local minima
  - Learn distributions for number of objects and object size
  - Learn probability density function for object position relative to nucleus and cell shape

Modeling Vesicular Organelles

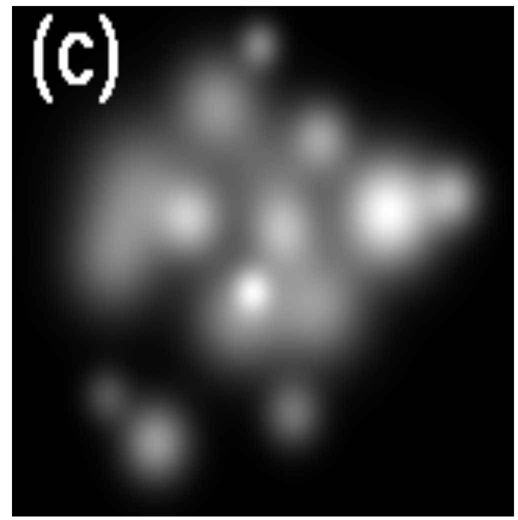
Original



Filtered

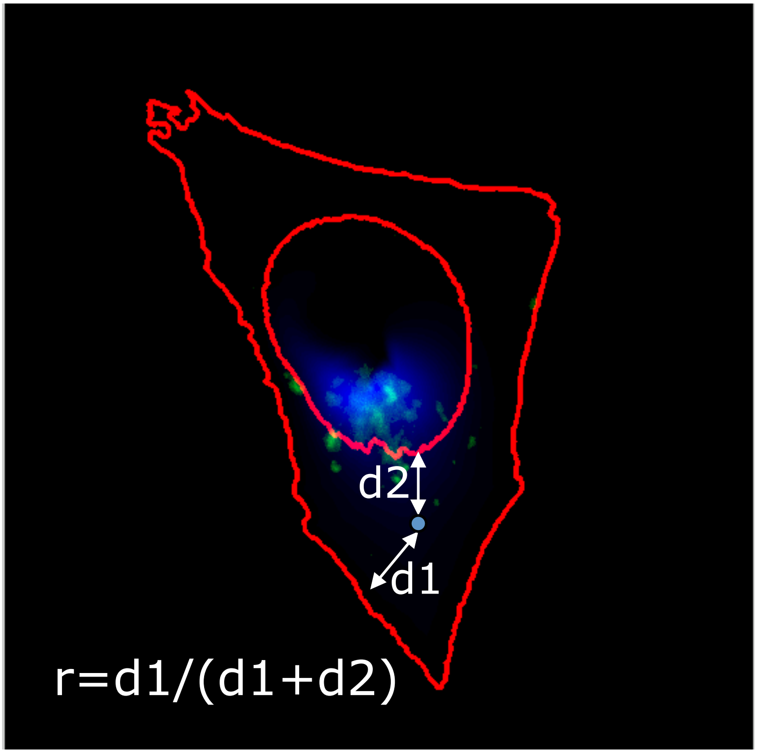


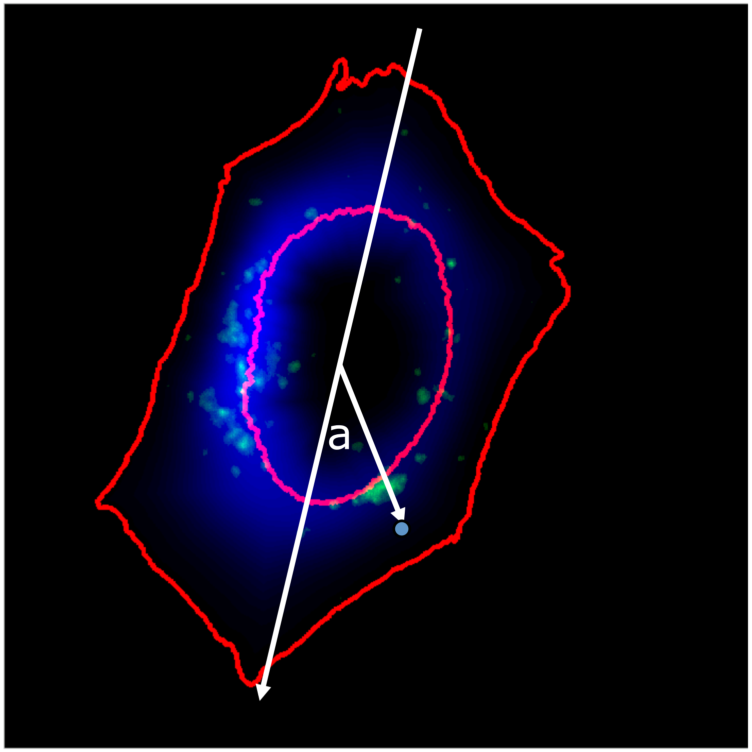
Fitted Gaussians



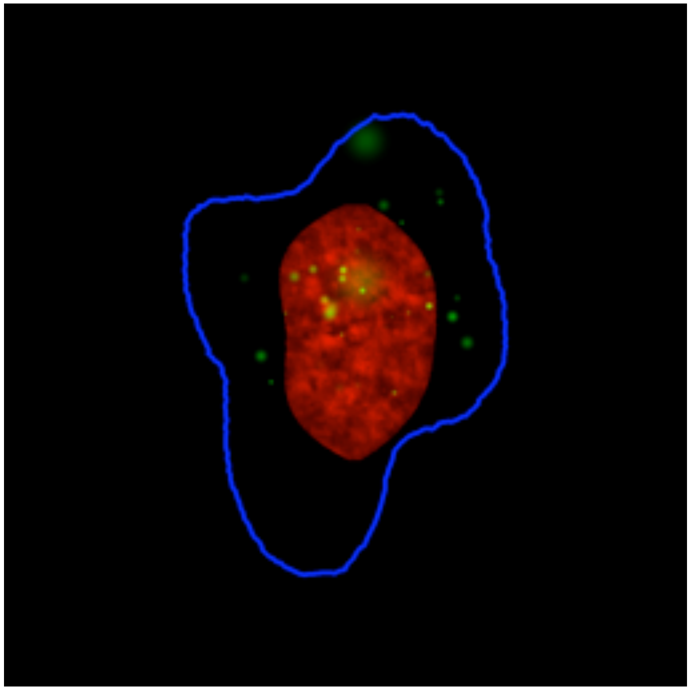
Position Model

r: normalized distance, a: angle to major axis

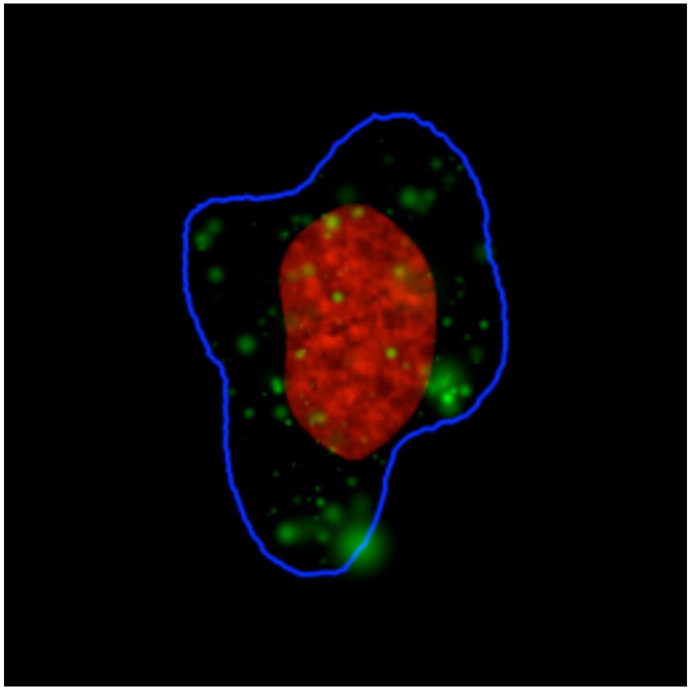




Synthesized Images



Lysosomes



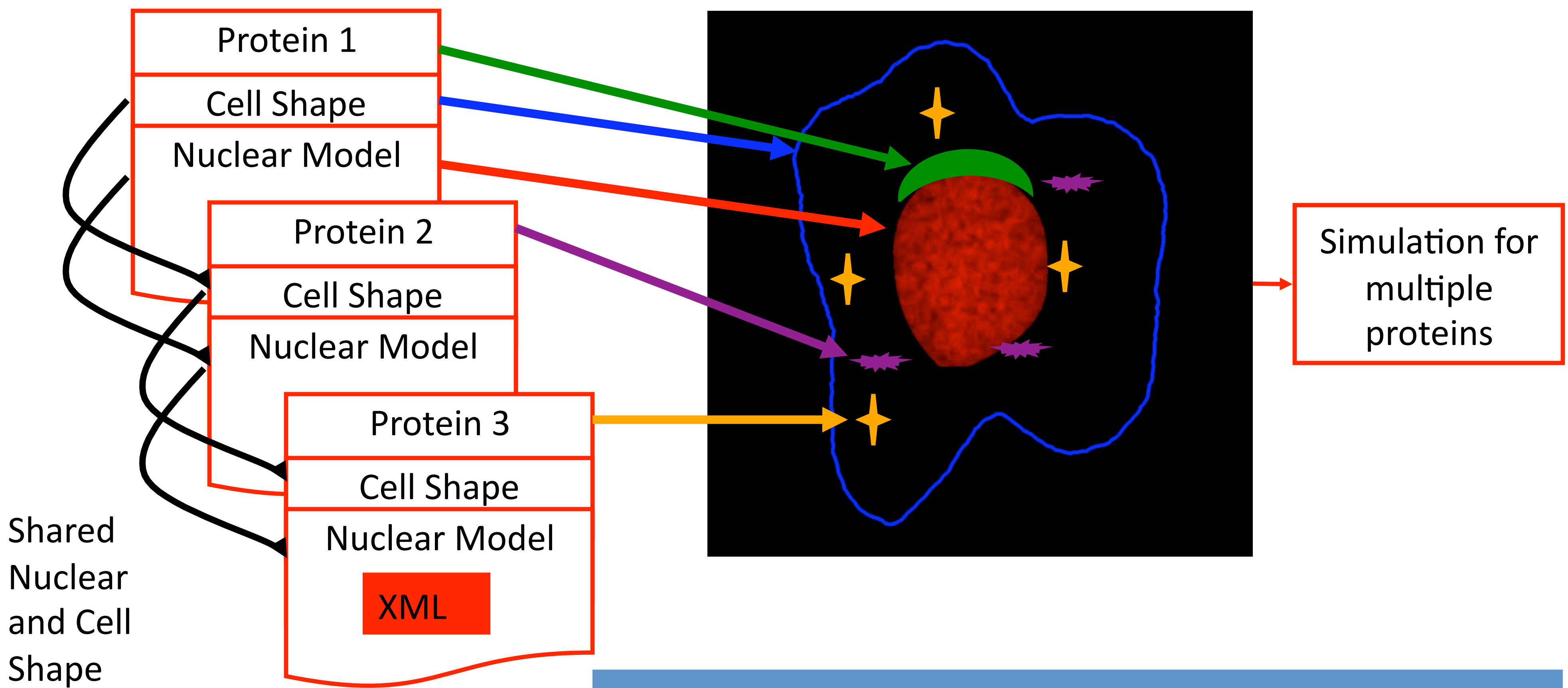
Endosomes

- SLML toolbox - Ivan Cao-Berg, Tao Peng, Ting Zhao
- Have portable tool for generating images from model

Framework for conditional subcellular location models

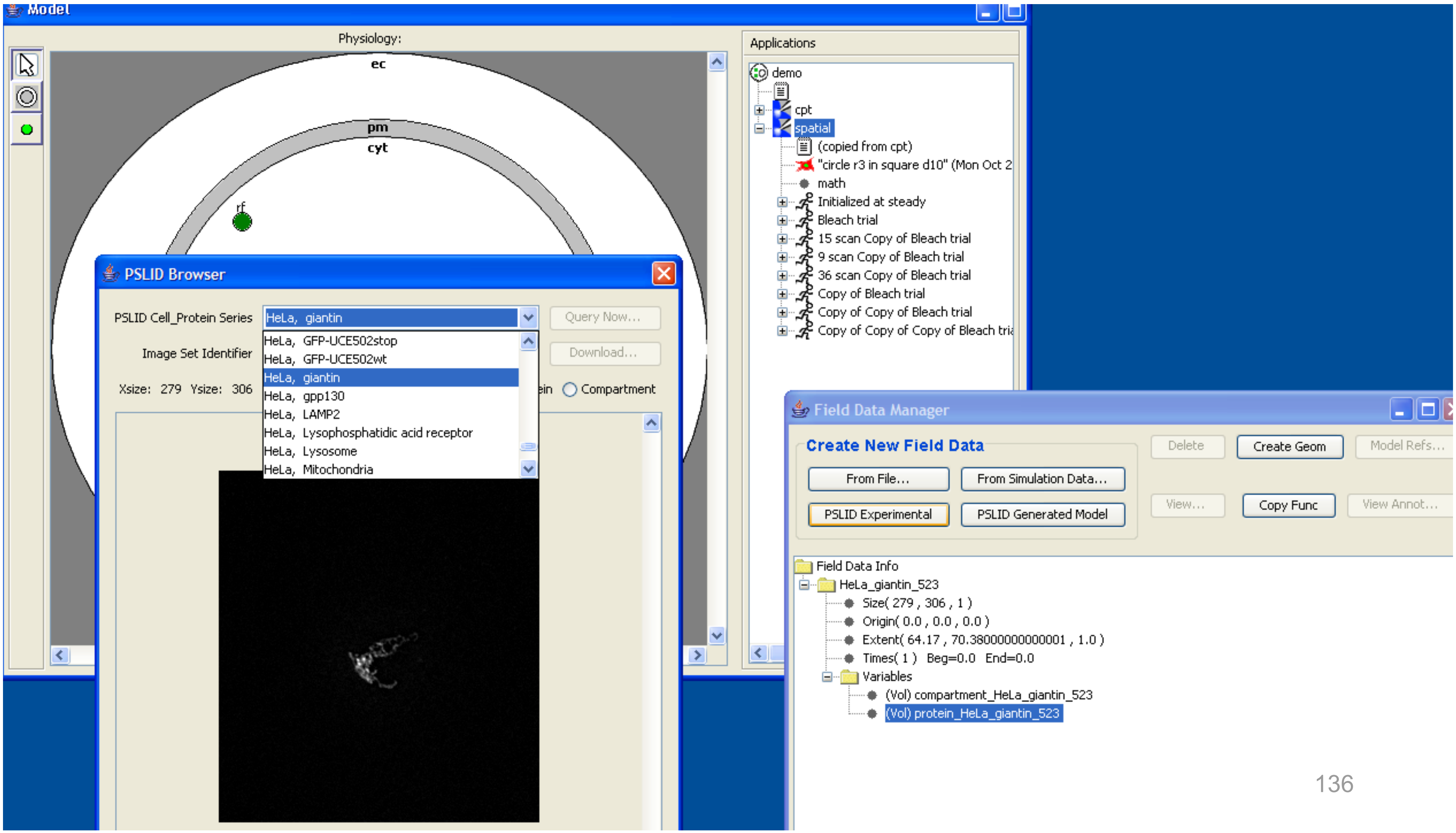
- SLML: slots for different parts of cell model
  - Nucleus
  - Plasma membrane
  - Specific protein
- Each slot can hold one of multiple types of models, each of which is probabilistic
- Each slot's model can be conditional (dependent) on another

Combining Models for Cell Simulations



Integrating with Virtual Cell (University of Connectiicut)) and M-Cell (Pittsburgh Supercomputing Center)

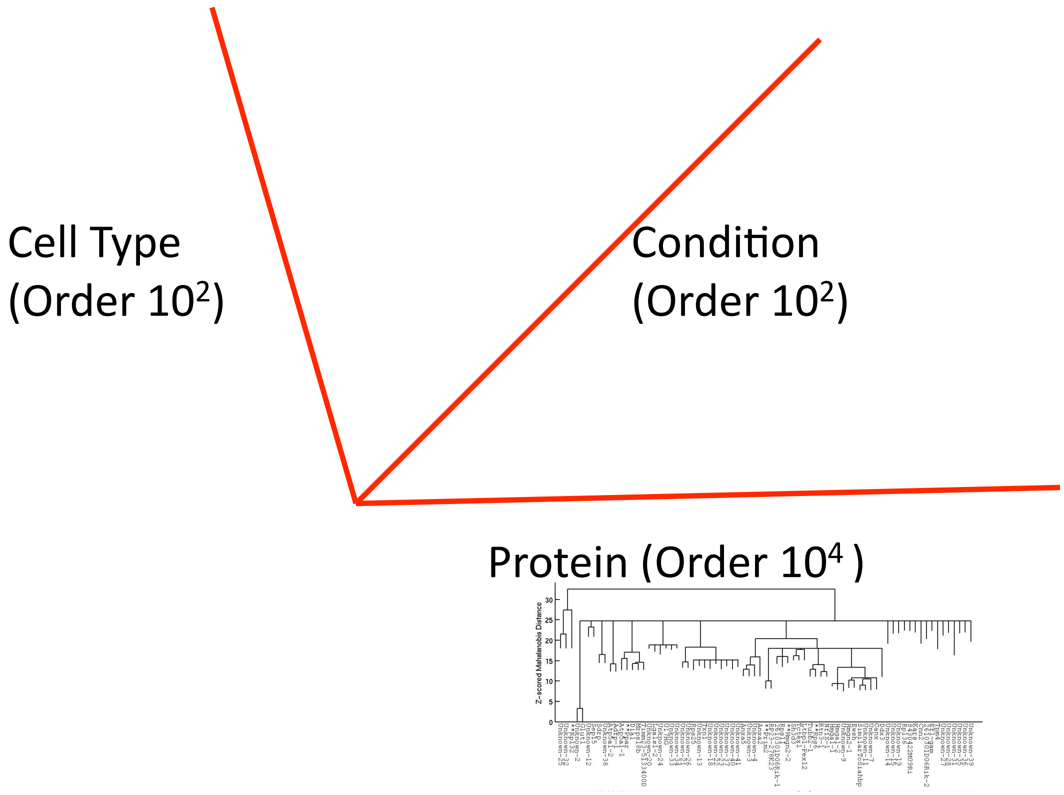
Virtual Cell-PSLID interface



Summary

- Automated analysis of subcellular patterns in cells and tissues demonstrated - computers better than people
- Complex patterns can be unmixed – useful for monitoring transitions between patterns (e.g., translocations)
- Generative models can be built directly from data to summarize results and make predictions – useful for cell simulations

The problem of subcellular location analysis



Plus: Time scale from subsecond to years

Towards an ultimate understanding of subcellular location

- Learn probabilistic models of subcellular location for different cell types
  - What do these models look like for different types of proteins and different organisms?
- Identify variables that affect the parameters of the models across cell types
- Construct "general model"

Acknowledgments

- Past and Present Students and Postdocs
  - Michael Boland (Hopkins), Mia Markey (UT Austin), Gregory Porreca (Harvard), Meel Velliste (U Pitt), Kai Huang, Xiang Chen (Yale), Yanhua Hu, Juchang Hua, **Ting Zhao (HHMI Janelia Farm)**, Shann-Ching Chen (Scripps), Elvira Garcia Osuna (CMU), **Justin Newberg**, **Estelle Glory**, **Tao Peng**, Luis Coelho
- Funding
  - NSF, NIH, Commonwealth of Pennsylvania
- Collaborators/Consultants
  - David Casasent, Simon Watkins, **Jon Jarvik, Peter Berget**, Jack Rohrer, Tom Mitchell, Christos Faloutsos, **Jelena Kovacevic, William Cohen, Geoff Gordon**, B. S. Manjunath, Ambuj Singh, Les Loew, Ion Moraru, Jim Schaff, Paul Campagnola, **Gustavo Rohde**
- Centers
  - Molecular Biosensors and Imaging Center - TCNP (Waggoner)
  - National Center for Integrative Biomedical Informatics - NCBC (Athey)

Alexander von Humboldt Foundation

Alexander von Humboldt
Stiftung/Foundation

Freiburg Institute for Advanced Studies