Machine Learning Approaches to Biological Research: Bioimage Informatics and Beyond

Lecture 2: Concepts of automated image analysis

Robert F. Murphy
External Senior Fellow, Freiburg Institute for Advanced Studies
Ray and Stephanie Lane Professor of Computational Biology, Carnegie Mellon University

September 29‐October 1, 2009

Goal

This is a microtubule pattern

Assign proteins to major subcellular structures using fluorescent microscopy

The Challenge

• Problem is hard because different cells have different shapes, sizes, orientations
• Organelles/structures within cells are not found in fixed locations
• Therefore, describe each image numerically and use 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 statistical classification methods to "learn" how to distinguish each class using the features

Acquisition considerations

- For automated acquisition
  - Optimize autofocus parameters
  - Maintain constant camera gain, exposure time, number of slices
  - Select interphase cells or ensure sampling of cell cycle

- Collect sufficient images per condition
  - For classifier training or set comparison, more than number of features
  - For classification or clustering, base on confidence level desired
- Collect reference images if possible (DNA, membrane)
Annotation considerations

- Maintain adequate records of all experimental settings
- Organize images by cell type/probe/condition

Preprocessing

- Correction for/Removal of camera defects
- Background correction
- Autofluorescence correction
- Illumination correction
- Deconvolution

Preprocessing (continued)

- Registration
  - Not critical if only using DNA or membrane references
- Intensity scaling (constant scale or contrast stretched for each cell)
- Single cell segmentation
  - Manual, semi-automated, automated
- Region finding
  - Nucleus
  - Cytoplasmic annulus
  - Cell boundary
Segmentation of Images into Single Cell Regions

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

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

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 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 for Subcellular Pattern Analysis

Subcellular Location Features (SLF)

• Combinations of features of different types that describe different aspects of patterns in fluorescence microscope images have been created
• Motivated in part by descriptions used by biologists (e.g., punctate, perinuclear)
• To ensure that the specific features used for a given experiment can be identified, they are referred to as Subcellular Location Features (SLF) and defined in sets (e.g., SLF1)
Feature levels and granularity

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

Thresholding

- First type of feature is morphological
- 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-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 Illustration

Ridler-Calvard Method

original

original

thresholded

Otsu Method

• Find threshold to minimize the variances of the pixels below and above it

Adaptive Thresholding

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

Suitability of Automated Thresholding for Classification

• For the task of subcellular pattern analysis, automated thresholding methods perform quite well in most cases, especially for patterns with well-separated objects
• They do not work well for images with very low signal-noise ratio
• Can tolerate poor behavior on a fraction of images for a given pattern while still achieving good classification accuracies

Object finding

• After choice of threshold, define objects as sets of touching pixels that are above threshold
2D Features
Morphological Features

<table>
<thead>
<tr>
<th>SLF No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLF1.1</td>
<td>The number of fluorescent objects in the image</td>
</tr>
<tr>
<td>SLF1.2</td>
<td>The Euler number of the image</td>
</tr>
<tr>
<td>SLF1.3</td>
<td>The average number of above-threshold pixels per object</td>
</tr>
<tr>
<td>SLF1.4</td>
<td>The variance of the number of above-threshold pixels per object</td>
</tr>
<tr>
<td>SLF1.5</td>
<td>The ratio of the size of the largest object to the smallest</td>
</tr>
<tr>
<td>SLF1.6</td>
<td>The average object distance to the cellular center of fluorescence (COF)</td>
</tr>
<tr>
<td>SLF1.7</td>
<td>The variance of object distances from the COF</td>
</tr>
<tr>
<td>SLF1.8</td>
<td>The ratio of the largest to the smallest object to COF distance</td>
</tr>
</tbody>
</table>

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 Features

**DNA Features**

<table>
<thead>
<tr>
<th>SLF No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLF2.17</td>
<td>The average object distance from the COF of the DNA image</td>
</tr>
<tr>
<td>SLF2.18</td>
<td>The variance of object distances from the DNA COF</td>
</tr>
<tr>
<td>SLF2.19</td>
<td>The ratio of the largest to the smallest object to DNA COF distance</td>
</tr>
<tr>
<td>SLF2.20</td>
<td>The distance between the protein COF and the DNA COF</td>
</tr>
<tr>
<td>SLF2.21</td>
<td>The ratio of the area occupied by protein to that occupied by DNA</td>
</tr>
<tr>
<td>SLF2.22</td>
<td>The fraction of the protein fluorescence that co-localizes with DNA</td>
</tr>
</tbody>
</table>

### Skeleton Features

<table>
<thead>
<tr>
<th>SLF No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLF7.80</td>
<td>The average length of the morphological skeleton of objects</td>
</tr>
<tr>
<td>SLF7.81</td>
<td>The ratio of object skeleton length to the area of the convex hull of the skeleton, averaged over all objects</td>
</tr>
<tr>
<td>SLF7.82</td>
<td>The fraction of object pixels contained within the skeleton</td>
</tr>
<tr>
<td>SLF7.83</td>
<td>The fraction of object fluorescence contained within the skeleton</td>
</tr>
<tr>
<td>SLF7.84</td>
<td>The ratio of the number of branch points in the skeleton to the length of skeleton</td>
</tr>
</tbody>
</table>

### Illustration – Skeleton

![Skeleton Illustration]
2D Features

<table>
<thead>
<tr>
<th>SLF No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLF1.9</td>
<td>The fraction of the non-zero pixels that are along an edge</td>
</tr>
<tr>
<td>SLF1.10</td>
<td>Measure of edge gradient intensity homogeneity</td>
</tr>
<tr>
<td>SLF1.11</td>
<td>Measure of edge direction homogeneity 1</td>
</tr>
<tr>
<td>SLF1.12</td>
<td>Measure of edge direction homogeneity 2</td>
</tr>
<tr>
<td>SLF1.13</td>
<td>Measure of edge direction difference</td>
</tr>
</tbody>
</table>

2D Features

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 Features

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

<table>
<thead>
<tr>
<th>1 1 3 4</th>
<th>1 2 3 4</th>
<th>1 2 3 4</th>
<th>1 2 3 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 1 3</td>
<td>1 2 1 0 1</td>
<td>1 0 1 0 1</td>
<td>1 0 3 0 1</td>
</tr>
<tr>
<td>2 4 4 4</td>
<td>2 1 6 3 4</td>
<td>2 1 4 3 0</td>
<td>2 3 0 4 4</td>
</tr>
<tr>
<td>5 1 4 2 2</td>
<td>1 0 3 6 2</td>
<td>1 0 3 4 1</td>
<td>1 0 4 0 3</td>
</tr>
<tr>
<td>6 2 3 2 2</td>
<td>1 4 2 4</td>
<td>1 3 3 1 2</td>
<td>1 1 4 3 2</td>
</tr>
</tbody>
</table>

Solid plus some noise

Random
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.

Wavelet Transformation - 1D

A: approximation (low frequency)
D: detail (high frequency)
X=A1+D3+D2+D1
2D Wavelets - intuition

• Apply some filter to detect edges (horizontal; vertical; diagonal)

Slide courtesy of Christos Faloutsos

2D Wavelets - intuition

• Recurse

Slide courtesy of Christos Faloutsos

2D Wavelets - intuition

• Many wavelet basis functions (filters):
  – Haar
  – Daubechies (-4, -6, -20)
• http://www331.jpl.nasa.gov/public/wave.html

Slide courtesy of Christos Faloutsos
Daubechies D4 decomposition

Original image  Wavelet Transformation

2D Features
Wavelet Feature Calculation
• Preprocessing
  – Background subtraction and thresholding
  – Translation and rotation
• Wavelet transformation
  – The Daubechies 4 wavelet
  – 10 level decomposition
  – Use the average energy of the three high-frequency components at each level as features

3D Features
Morphological
• 28 features, 14 from protein objects and 14 from their relationship to corresponding DNA images
  – Based on number of objects, object size, object distance to COF
• Corresponding DNA image required
3D set

- 14 SLF-9 features that do not require DNA images
- 2 Edge features
  - Ratio of above threshold pixel along an edge
  - Ratio of fluorescence along an edge
- 26 3D Haralick texture features
  - Gray level co-occurrence matrix for 13 directions
  - Calculate 13 Haralick statistics for each direction
  - Average each statistic over 13 directions and use mean and range as separate features: result is 26 features

Object level features (SOF)

- Subset of SLFs calculated on single objects

<table>
<thead>
<tr>
<th>Index</th>
<th>Feature Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOF 1.1</td>
<td>Number of pixels in object</td>
</tr>
<tr>
<td>SOF 1.2</td>
<td>Distance between object center of fluorescence (COF) and DNA COF</td>
</tr>
<tr>
<td>SOF 1.3</td>
<td>Fraction of object pixels overlapping with DNA</td>
</tr>
<tr>
<td>SOF 1.4</td>
<td>A measure of eccentricity of the object</td>
</tr>
<tr>
<td>SOF 1.5</td>
<td>Euler number of the object</td>
</tr>
<tr>
<td>SOF 1.6</td>
<td>A measure of roundness of the object</td>
</tr>
<tr>
<td>SOF 1.7</td>
<td>The length of the object's skeleton</td>
</tr>
<tr>
<td>SOF 1.8</td>
<td>The ratio of skeleton length to the area of the convex hull of the skeleton</td>
</tr>
<tr>
<td>SOF 1.9</td>
<td>The fraction of object pixels contained within the skeleton</td>
</tr>
<tr>
<td>SOF 1.10</td>
<td>The fraction of object fluorescence contained within the skeleton</td>
</tr>
<tr>
<td>SOF 1.11</td>
<td>The ratio of the number of branch points in skeleton to length of skeleton</td>
</tr>
</tbody>
</table>

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**.

---

Temporal Textures
based on Co-occurrence Matrix

- Temporal co-occurrence matrix $P$:
  - $N_{x\text{pixel}} \times N_{y\text{pixel}}$ matrix, Element $P[i,j]$ is the probability that a pixel with value $i$ has value $j$ in the next image (time point).
- Thirteen statistics calculated on $P$ are used as features.

---

<table>
<thead>
<tr>
<th>Image at $t_0$</th>
<th>Image at $t_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 2 2 2 4</td>
<td>4 2 2 2 4</td>
</tr>
<tr>
<td>1 2 4 1 1</td>
<td>1 2 4 1 1</td>
</tr>
<tr>
<td>3 4 4 4 2</td>
<td>3 4 4 4 2</td>
</tr>
<tr>
<td>2 2 3 3 2</td>
<td>2 2 3 3 2</td>
</tr>
<tr>
<td>3 3 3 2 4</td>
<td>3 3 3 2 4</td>
</tr>
</tbody>
</table>

Temporal
co-occurrence
matrix (for
image that does
not change)

```
1 2 3 4
1 3 0 0 0
2 0 9 0 0
3 0 0 6 0
4 0 0 0 7
```
Temporal co-occurrence matrix (for image that changes)

<table>
<thead>
<tr>
<th>Time Interval</th>
<th>ER</th>
<th>Tubulin</th>
<th>DNA</th>
<th>TfR</th>
<th>Ac</th>
<th>Nucleolin</th>
<th>Mito</th>
<th>LAMP</th>
<th>gpp130</th>
<th>gian</th>
</tr>
</thead>
<tbody>
<tr>
<td>0s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>135s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>180s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>225s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>270s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>315s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>360s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>405s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Implementation of Temporal Texture Features

- Compare image pairs with different time interval, compute 13 temporal texture features for each pair.
- Use the average and variance of features in each kind of time interval, yields 13*5*2=130 features

Task: Learn to recognize all major subcellular patterns

2D Images of HeLa cells
2D Classification Results

<table>
<thead>
<tr>
<th>True Class</th>
<th>Output of the Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNA</td>
<td>99 1 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>ER</td>
<td>0 97 0 0 0 2 0 0 0 1 0</td>
</tr>
<tr>
<td>Gia</td>
<td>0 0 91 7 0 0 0 0 2 0 0</td>
</tr>
<tr>
<td>Gpp</td>
<td>0 0 74 32 0 2 0 0 1 0 0</td>
</tr>
<tr>
<td>Lam</td>
<td>0 0 1 0 86 1 0 0 0 10 0</td>
</tr>
<tr>
<td>Mit</td>
<td>0 3 0 0 0 92 0 0 3 3 0</td>
</tr>
<tr>
<td>Nuc</td>
<td>0 0 0 0 0 0 99 0 1 0 0</td>
</tr>
<tr>
<td>Act</td>
<td>0 0 0 0 0 0 0 100 0 0 0</td>
</tr>
<tr>
<td>TR</td>
<td>0 1 0 0 12 2 0 1 1 2 0</td>
</tr>
<tr>
<td>Tub</td>
<td>1 2 0 0 0 1 0 0 1 95 0</td>
</tr>
</tbody>
</table>

Overall accuracy = 92%

Human Classification Results

<table>
<thead>
<tr>
<th>True Class</th>
<th>Output of the Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNA</td>
<td>100 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>ER</td>
<td>0 90 0 0 3 6 0 0 0 0 0</td>
</tr>
<tr>
<td>Gia</td>
<td>0 0 56 36 3 3 0 0 0 0 0</td>
</tr>
<tr>
<td>Gpp</td>
<td>0 0 64 33 0 0 0 0 0 3 0</td>
</tr>
<tr>
<td>Lam</td>
<td>0 0 6 0 71 0 0 0 20 0 0</td>
</tr>
<tr>
<td>Mit</td>
<td>0 3 0 0 0 99 0 0 0 3 0</td>
</tr>
<tr>
<td>Nuc</td>
<td>0 0 0 0 0 0 0 100 0 0 0</td>
</tr>
<tr>
<td>Act</td>
<td>0 0 0 0 0 0 0 0 100 0 0</td>
</tr>
<tr>
<td>TR</td>
<td>0 13 0 0 3 0 0 0 83 0 0</td>
</tr>
<tr>
<td>Tub</td>
<td>0 3 0 0 0 0 0 3 0 93 0</td>
</tr>
</tbody>
</table>

Overall accuracy = 83%

Computer vs. Human

![Graph showing the comparison between computer accuracy and human accuracy.](image)
### 3D Classification Results

<table>
<thead>
<tr>
<th>True Class</th>
<th>Output of the Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNA</td>
<td>98 2 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>ER</td>
<td>0 100 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>Gia</td>
<td>0 0 100 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>Gpp</td>
<td>0 0 0 96 4 0 0 0 0 0 0</td>
</tr>
<tr>
<td>Lam</td>
<td>0 0 0 4 96 0 0 0 0 0 2</td>
</tr>
<tr>
<td>Mit</td>
<td>0 0 2 0 0 76 0 2 0 0 0</td>
</tr>
<tr>
<td>Nuc</td>
<td>0 0 0 0 0 0 100 0 0 0 0</td>
</tr>
<tr>
<td>Act</td>
<td>0 0 0 0 0 0 0 100 0 0 0</td>
</tr>
<tr>
<td>TR</td>
<td>0 0 0 0 2 0 0 0 0 0 98 2</td>
</tr>
<tr>
<td>Tab</td>
<td>0 2 0 0 0 0 0 0 0 0 98 0</td>
</tr>
</tbody>
</table>

Overall accuracy = 98%

---

### Supervised vs. Unsupervised Learning

- Work discussed so far demonstrates the feasibility of using classification methods to assign all proteins to known major classes
- Do we know all locations? Are assignments to major classes enough?
- Need approach to discover classes

---

### Location Proteomics

- **Tag many proteins**
  - We have used 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...
Principles of CD-Tagging (Jarvik & Berget)  
(CD = Central Dogma)

- Exon 1
- Intron 1
- Exon 2

Genomic DNA + CD-cassette

- Exon 3
- Tag
- Exon 1
- Tag
- Exon 2
- Tag

CD cassette

- Tag
- Tagged mRNA
- Tagged Protein

Location Proteomics

- Tag many proteins
- We have used 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 each cell
- 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

Jarvik et al. 2002
What Now?

Group ~90 tagged clones by pattern

Solu>on:
Group them automatically

 How?
 Features can be used to measure similarity of protein patterns
 Build Subcellular Location Tree
 Have multiple images per protein
 Sample repeatedly from available images, build cluster tree for each subsample, and form consensus tree

Chen et al 2003; Chen and Murphy 2005

Nucleolar Proteins
Punctate Nuclear Proteins

Nuclear and Cytoplasmic Proteins with Some Punctate Staining

Uniform