Journal of Statistical Software

June 2009, Volume 30, Issue 10. http://www.jstatsoft.org/

An Interactive Java Statistical Image Segmentation System: GemIdent

Susan Holmes
Stanford University

Adam Kapelner
Stanford Medical School

Peter P. Lee
Stanford Medical School

Abstract

Supervised learning can be used to segment/identify regions of interest in images using both color and morphological information. A novel object identification algorithm was developed in Java to locate immune and cancer cells in images of immunohistochemically-stained lymph node tissue from a recent study published by Kohrt *et al.* (2005). The algorithms are also showing promise in other domains. The success of the method depends heavily on the use of color, the relative homogeneity of object appearance and on interactivity. As is often the case in segmentation, an algorithm specifically tailored to the application works better than using broader methods that work passably well on any problem. Our main innovation is the interactive feature extraction from color images. We also enable the user to improve the classification with an interactive visualization system. This is then coupled with the statistical learning algorithms and intensive feedback from the user over many classification-correction iterations, resulting in a highly accurate and user-friendly solution. The system ultimately provides the locations of every cell recognized in the entire tissue in a text file tailored to be easily imported into R (Ihaka and Gentleman 1996; R Development Core Team 2009) for further statistical analyses. This data is invaluable in the study of spatial and multidimensional relationships between cell populations and tumor structure. This system is available at `http://www.GemIdent.com/` together with three demonstration videos and a manual.

Keywords: interactive boosting, cell recognition, image segmentation, Java.

1. Introduction

We start with an overview of current practices in image recognition and a short presentation of the clinical context that motivated this research, we then describe the software and the complete workflow involved, finally the last two sections present technical details and potential improvements. The interactive algorithm, although developed to solve a specific problem in

Figure 1: The original image (left), a mask superimposed on the original image showing the results of pixel classification (center), the original image marked with the centers of the oranges (right).

histology, works on a wide variety of images. For instance, locating of oranges in a photograph of an orange grove (see Figure 1).

Any image that has few relevant colors, such as green and orange in the above example, where the objects of interest vary little in shape, size, and color, can be analyzed using our algorithm. First, we will describe the application to cell recognition in microscopic images.

1.1. Previous research

As emphasized in recent reviews (Ortiz de Solirzano *et al.* 1999; Mahalanobis *et al.* 1996; Wu *et al.* 1995, 1998; Yang and Parvin 2003; Gil *et al.* 2002; Fang *et al.* 2003; Kovalev *et al.* 2006), automated computer vision techniques applied to microscopy images are transforming the field of pathology. Not only can computerized vision techniques automate cell type recognition but they enable a more objective approach to cell classification providing at the same time a hierarchy of quantitative features measured on the images. Recent work on character recognition (Chakraborty 2003) shows how efficient interactivity can be in image recognition problems, with the user pointing out mistakes in real time, thus providing online improvement. In modern jargon, we call this interactive boosting (Freund and Schapire 1997). Current cell image analysis systems such as **EBImage** (Sklyar and Huber 2006) and **Midas** (Levenson 2006) or **ImageJ** (Collins 2007) do not provide these interactive visualization and correction features.

1.2. The specific context of breast cancer prognosis

Kohrt *et al.* (2005) showed that breast cancer prognosis could be greatly improved by using immune population information from immunohistochemically-stained lymph nodes. To take this analysis a step further, we would like to detect and pinpoint the location of each and every cancer and immune cell in the high-resolution full-mount images of lymph nodes acquired via automated microscopy. This task is harder than classification of an entire slide as normal or abnormal as done in Maggioni *et al.* (2004) for instance.

A typical tissue contains a variety of regions characteristic of cancer, immune cells, or both. It would not be possible for a histopathologist to identify and count all the cells of each type on such a slide. Even if a whole team of cell counters were available, the problems of subjectivity and bias on such a scale would discredit the results. It is very useful to have an automated

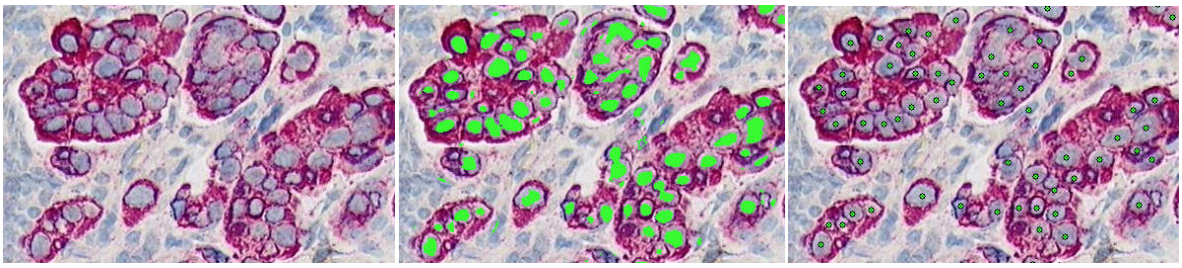

Figure 2: The original image (left), a mask superimposed on the original image showing the results of pixel classification (center), the original image marked with the centers of the nuclei (right).

system to identify and count cells objectively.

Kohrt *et al.* (2005) also showed that T-cell and dendritic cell populations within axillary lymph nodes of patients with breast cancer are significantly decreased in patients who relapsed. No study thus far has examined the spatial variability in lymphocyte populations and phenotypes as related to lymph node-infiltrating tumor cells and clinical outcome. The location of tumor-dependent immune modulation has significant sequelæ, given the critical role of lymph nodes in activation of the immune response. This suggests that tissue architecture could yield clues to the workings of the immune system in breast cancer. This can be investigated through spatial analysis of different cell populations. The limiting step to date has been locating every cell.

1.3. GemIdent

This algorithm has been engineered into the software package named **GemIdent** (Kapelner *et al.* 2007a,b). The distribution, implemented in Java, includes an easy-to-use GUI with four panels – color (or stain) training, phenotype training/retraining (see Figures 8, 10, and 11), classification, and data analysis (see Figure 13) with a final data output into a text file which is easy to input and analyse in R. The Java implementation ensures that full platform-independence is supported. In addition to supporting standard RGB images (in tiff or jpg format), the distribution also includes support for image sets derived from the Bacus Laboratories Incorporated Slide Scanner (BLISS, Bacus Laboratories, Inc.) and the CRI Nuance Multispectral Imager (CRI Inc., Woburn, MA).

In this paper, we focus on the software itself, its use in conjunction with R and the developments which were engineered to analyze multispectral images where the chromatic markers (called chromagens) are separated a priori .

Figure 2 shows **GemIdent** employed in the localization of cancer nuclei in a typical Kohrt (Kohrt *et al.* 2005) image. The algorithm internals are detailed in Section 2 but we give a brief summary here for potential users who do not need to know the technical details.

Our procedure requires interactive training: the user defines the target objects sought from the images. In the breast cancer study, this would be the cancer nuclei themselves. The user must also provide examples of the "opposite" of the cancer nuclei, i.e. the "non-object" (NON). Figure 3 shows an excerpt from the training step.

New images can be classified into cancer nuclei and non-cancer nuclei regions using a statistical

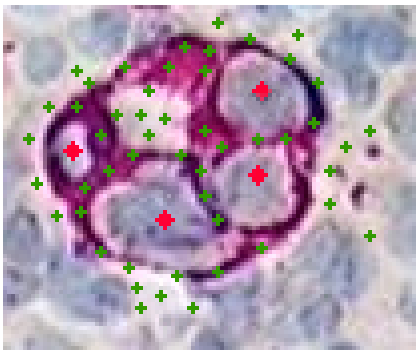

Figure 3: An example of training in a microscopic image of immunohistochemically stained lymph node. The cancer membranes are stained for Fast Red (displays maroon). There is a background nuclei counterstain that appears blue. The phenotype of interest are cancer cell nuclei. Training points for the nuclei appear as red diamonds and training points for the NON appear as green crosses.

learning classifier (Hastie *et al.* 2001). Using simple geometric techniques, the centers of the cancer nuclei can then be located and an accurate count tabulated.

The user can then return and examine the mistakes, correct them, and retrain the classifier. In this way a more accurate classifier is iteratively constructed. The process is repeated until the user is satisfied with the results.

In Section 3 we detail a classification and analysis of an entire multispectrally-imaged Kohrt lymphnode, complete with screenshots of the software at each step.

2. Algorithm

Image acquisition

Any multiple wavelength set of images can be used, as long as they are completely aligned. In this example, the images were acquired via a modified CRI Nuance Multispectral Imager (CRI Inc., Woburn, MA). The acquisition was completely automated via an electronically controlled microscopy with an image decomposition into subimages. Multiple slides were sequentially scanned via an electronically controlled cartridge that feeds and retracts slides to and from the stage. The setup enables scanning of multiple slides with large histological samples (some being $10mm+$ in diameter and spanning as many as $5,000$ subimages or stages) at $200X$ magnification.

Each raw exposure is a collection of eight monochromatic images corresponding to the eight unique wavelength filters scanned (for information on dimension reduction in spectral imaging see Lee *et al.* 1990). The choice of eight is to ensure a complete span of the range of the visual spectrum but small enough to ensure a reasonable speed of image acquisition and avoid unwieldly amounts of data. These multispectral images are composed of 15-bit pixels whose values correspond to the optical density at that location. The files for each wavelength for each subimage or stage are typically TIF files of about 3MB each.

Prior to whole-tissue scanning, a small representative image is taken that contains lucid ex-

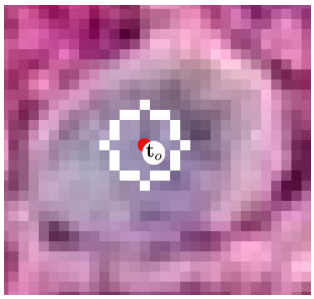

Figure 4: Example of a ring mask: $\mathbf{c}_4$ superimposed onto a typical cancer nuclei from the Kohrt images, white pixels designate the points which participate in the ring's score.

amples of each of the S chromagens. The user selects multiple examples of each of the S chromagens. This information is used to compute a "spectral library." The same spectral library can then be used for every histological sample that is stained with the same chromagens.

After whole-tissue scanning, the CRI Nuance Multispectral Imager (CRI Inc., Woburn, MA) combines the intermediate images using spectral unmixing algorithms to yield S orthogonal chromagen channel images. These "spectrally unmixed" images are also monochromatic and composed of 15-bit pixels whose values correspond to a pseudo-optical density.

We use the spectrally unmixed images *directly* as score matrices, which we call F_s where s is the chromagen of interest. The program has been used with various scoring generation mechanisms and the quality of the statistical learning output has proved quite robust to these changes. Multispectral microscopy has the advantage of providing a clean separation of the chromagen signals.

Training phase

1. Interactive acquisition of the training sets for objects of interest.

   For each of the P phenotypes or categories of objects of interest, a training set is built: the user interactively chooses example points, $\mathbf{t} = (i, j)$, in the images where the phenotype *is* located forming the lists: $T_1^+, T_2^+, \ldots, T_P^+$. In addition, the user interactively chooses example points where *none* of the P phenotypes are located, i.e. the "NON" forming the list T^-.

Learning phase

2. Feature definition:

   (a) Define a "ring", $\mathbf{c}_r$, to be the 0/1 mask of the locus of pixels, $\mathbf{t}$, $r + \epsilon$ distance away from the center point $\mathbf{t}_o$ where ϵ refers to the error inherent in discretizing a circle (see Figure 4). Generate a set of rings, for $r \in \{0, 1, 2, \ldots, R\}$ where R is the maximum radius of the largest phenotype to be identified and it can be modified by a multiplicative factor which controls the amount of additonal information beyond R to be included: $C \equiv \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_R\}$

(b) For a point $\mathbf{t}_o = (i_o, j_o)$ in the training set and for a ring $\mathbf{c}_r$, create a "Ring Score," $\ell_{q,r}$, by summing the scores for all points in the mask:

$$\ell_{s,r}(\mathbf{t}_o) = \sum_{\mathbf{t} \in \mathbf{c}_r} F_s(\mathbf{t}_o + \mathbf{t})$$

(c) Repeat for all rings in C and for all S score matrices to generate the observation record, $\mathbf{L}_i$ of length $S \times R$ by vectorizing $\ell_{s,r}$ and appending the categorical variable denoting the phenotype, p:

$$\mathbf{L}(\mathbf{t}_o) = \big(\ell_{1,1}, \ell_{1,2}, \ldots, \ell_{1,R}, \ell_{2,1}, \ell_{2,2}, \ldots, \ell_{2,R}, \ldots\ldots, \ell_{S,1}, \ell_{S,2}, \ldots, \ell_{S,R}, p\big)$$

(d) Now compute an observation record for each point $\mathbf{t}$ in the training sets for all phenotypes and for the 'NON' category.

3. Creation of a classifier:

   All observation vectors are concatenated row-wise into a training data matrix, and a supervised learning classifier that will be used to classify all phenotypes is created.

   In this implementation we use the statistical learning technique known as "Random Forests" developed by Breiman (2001) to automatically rank features among a large collection of candidates. This technique has been compared to a suite of other learning techniques in a cell recognition problem in Kovalev *et al.* (2006) who found it to be the best technique providing both the most accurate and the least variable of all the techniques compared. As all supervised learning techniques (Hastie *et al.* 2001), the method depends on the availability of a good training set of images where the pixels have already been classified into several groups.

   At this point, all previous data generated can be discarded. Most machine learning classifiers, including our version of Random Forests, provide information on which scores $\ell_{q,r}$ are important in the classification (see an example in Figure 9).

Classification phase

4. Pixel classification:

   For each image I to be classified, an observation record is created for each pixel (steps 2a and 2b), $\mathbf{L}(\mathbf{t}_o)$, and then evaluated by the classifier. The result is then stored in a binary matrix, with 1 coding for the phenotype and 0 for the opposite. There are P binary matrices, $B_1, B_2, \ldots, B_P$, (one for each phenotype):

$$I \xrightarrow{\text{supervised learning classifier}} B_1, B_2, \ldots, B_P$$

   To enhance speed, k pixels can be skipped at a time and the B_p's can be closed k times (dilated then eroded using a 4N mask, see Gonzalez *et al.* 2004) to fill in holes.

5. Post-processing

   Each pixel is now classified. Contiguously classified regions, called "blobs" hereon are post-processed in order to locate centers of the relevant objects, such as the middle of an orange or the cell nucleus.

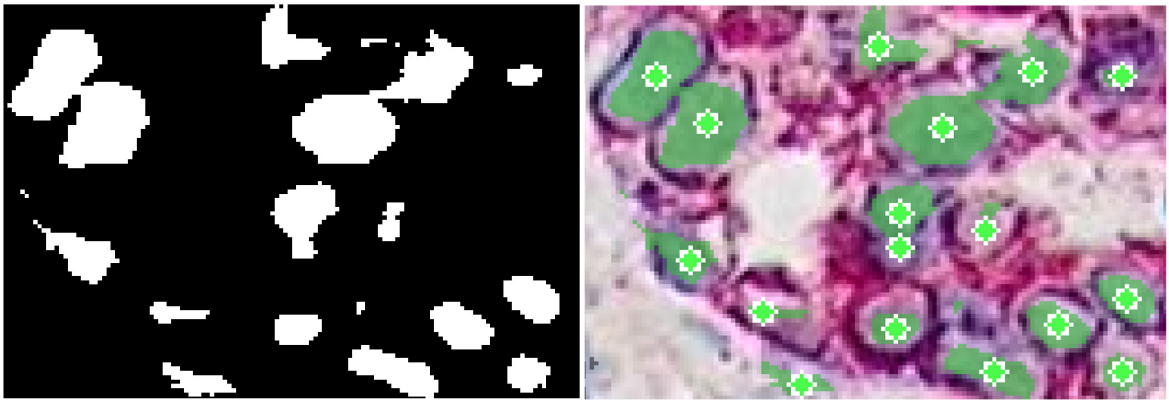

Figure 5: Left: Excerpt of B_{cancer} matrix. Right: The results of the centroid-finding algorithm superimposed on the marked image.

Define the matrices C to hold centroid information:

$$B_1, B_2, \ldots, B_P \xrightarrow{\text{blob analysis}} C_1, C_2, \ldots, C_P$$

There are many such algorithms for blob analysis. We used a simple one which we summarize below. For a more detailed description, see Appendix A. For an example of the results it produces, see Figure 5

A sample distribution of the training blob sizes is created by reconciling the user's training points then counting the number of pixels in each blob using the floodfill algorithm. We calculated the 2nd percentile, 95th percentile, and the mean. For each blob obtained from the classification, we used those sample statistics to formulate a heuristic rule that discards those that are too small and quarantines those that are too large. Those that are too large are split up based on the mean blob size. To locate each blob's centroid, the blob's coordinates were averaged.

Additional training phase(s)

6. Retraining-reclassification iterations:

   After classification (and post-processing if desired), the results from the B's and the C's can be superimposed onto the original training image. The user can add false negatives (hence adding points to the T_p^+'s) and add false positives (hence adding points to T^-). The observation records can then be regenerated, the supervised learning classifier can be relearned, and classification can be rerun with greater accuracy. This retraining-reclassification process can be repeated until the user is satisfied with the results.

3. A complete example

Immunohistochemical staining (IHC) refers to the process of identifying proteins in cells of a slice of tissue through the specificity with which certain antibodies bind to special antigens

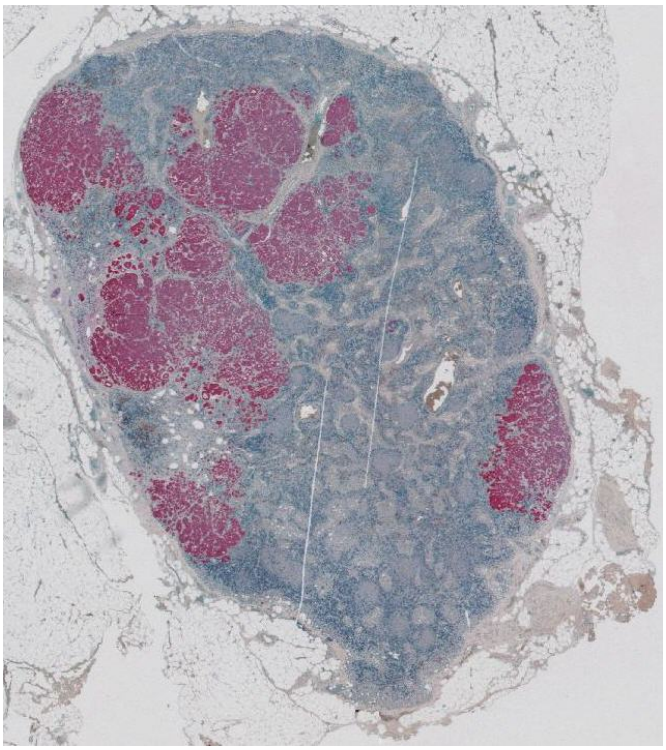

Figure 6: The tumor invaded lymph node as it appears through the microscope, at this resolution we can see the red zones which are Tumor invaded.

present on (or inside of) the cell. Combining a stain called a chromagen with the antibodies allows visualization and reveals their localization within the field. IHC is thus widely used to discover in situ distributions of different types of cells. Until recently most of the data collection was done by manual cell counting. In this example we will show how different types of immune cells as well as cancer cells were detected and localized in large numbers using statistical learning for image segmentation.

A lymphnode tissue section (see Figure 6) was stained for the folllowing markers by the following chromagens: CD1a targeting dendritic cells, $3,3'-$Diaminobenzidine (DAB); CD4 targeting T-cells, Ferangi Blue; and AE1/AE3 targeting cytokeratin within breast cancer cells, Vulcan Red. In addition, slides were stained with blue Hematoxylin to reveal all nuclei. The tissue was then imaged multispectrally (see "Image acquisition" in Section 2) and the set was loaded as a new project into **GemIdent**.

3.1. Selecting the training set

When the image set is first opened, a training helper appears (see Figure 7). Each number represents the snapshots (called stages) captured individually by the system. The user then chooses a small, but representative number of images to compose the initial "training set."

3.2. Initial training

The user begins in the "phenotype training" tab and first enumerates the phenotypes being

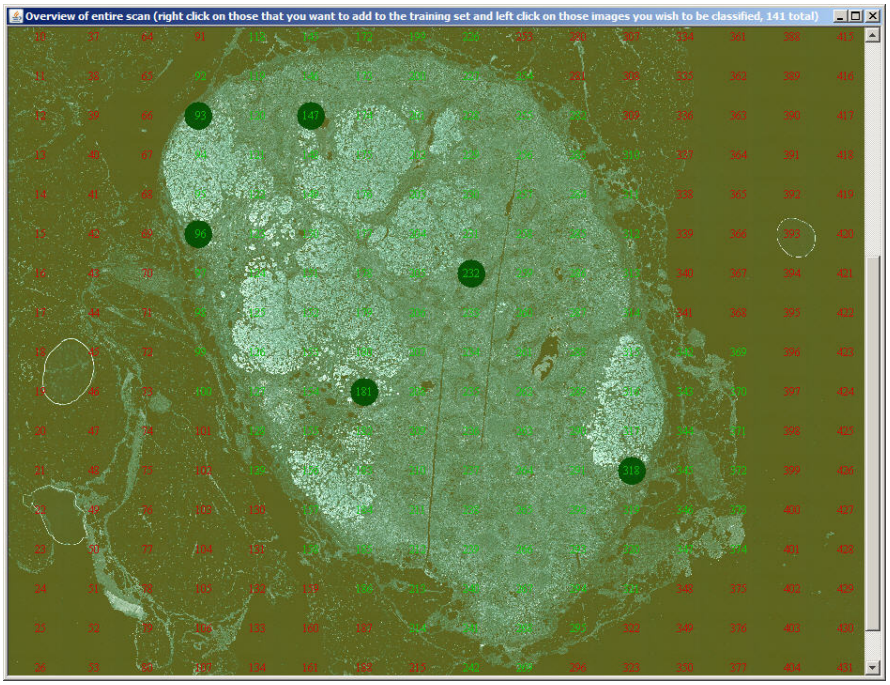

Figure 7: An assembly of all the individual stages in the complete scan. The parts of the images that we want to train and classify are indicated in green, the ones that will be discarded have red numbers, the green disks show the training set.

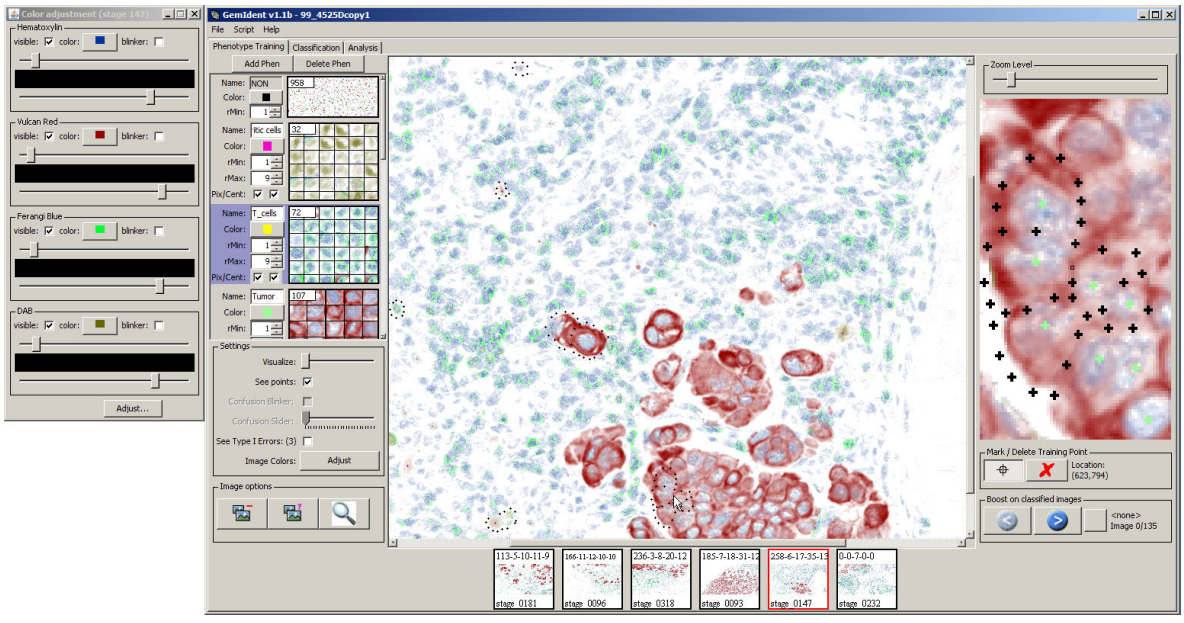

Figure 8: The pixels chosen as training points are shown as crosses. The color of the crosses indicates the training phenotype chosen.

sought and chooses an identifying color for each phenotype. `Tumor` cell nuclei were chosen to be green; `T-cell` nuclei, yellow; `Dendritic cell` nuclei, pink; and unspecified nuclei (called `other_cell`), orange (see left pane in Figure 8).

The user then begins interactively selecting pixels that exemplify each phenotype as well as examples of pixels that do not represent any of the phenotypes (ie the "NON" pixels) by clicking on the image (see Section 2, Step 1). A magnification window with customizable zoom helps the user select precise pixels, allowing for a highly accurate training set. The current image being trained can be alternated by selecting from thumbnails representing the training set (see main, right, and bottom panels in Figure 8).

In multispectral image sets, each individual chromagen signal can be customized (see the "Color Adjustment" dialog window in Figure 8). The display color can be chosen arbitrarily and the range of signal to be displayed can also be varied. In addition, chromagens can be flashed on/off. This is useful when training sets with phenotypes represented by chromagen colocalizations (none of the phenotypes in this example exhibit this property).

There is also a rare event finder that searches the entire set for instances of rare markers or colocalizations (not shown).

3.3. Classification

After specifying a training set for all phenotypes and the NON's, the user transitions to the classification tab (not shown). Here, the user can customize the number of trees in the random forest classifier, number of CPUs to be utilized, which images to classify, and a few other options before classification.

GemIdent then proceeds to creates a random forest (see Section 2, Steps 2 and 3). Upon completion, a graphic is displayed illustrating the importances of each feature in the classifier (see Figure 9).

Then, every pixel is assigned a phenotype or put in the "NON" group by evaluating it in the random forest (see Section 2, Step 4). The output is zones or "blobs" of different phenotypes. Upon completion, the user can choose to resolve the blobs into centroids (see Section 2, Step 5), thus enabling cell counting and cell localization.

After the centroids have been located, the program can evaluate its errors. The type I errors are shown in a small window and written to a summary file in the output directory:

```
filename,Num_Dendritic_cells,Num_T_cells,Num_other_cells,Num_Tumor
stage_0096,23,482,479,44
stage_0093,29,230,259,615
stage_0126,3,46,166,816
stage_0147,18,1203,907,81
.................
Error Rates
Dendritic_cells         2.78
T_cells                 4.35
other_cells             0.0
Tumor                   7.34
Totals,922,85108,75457,27790
```

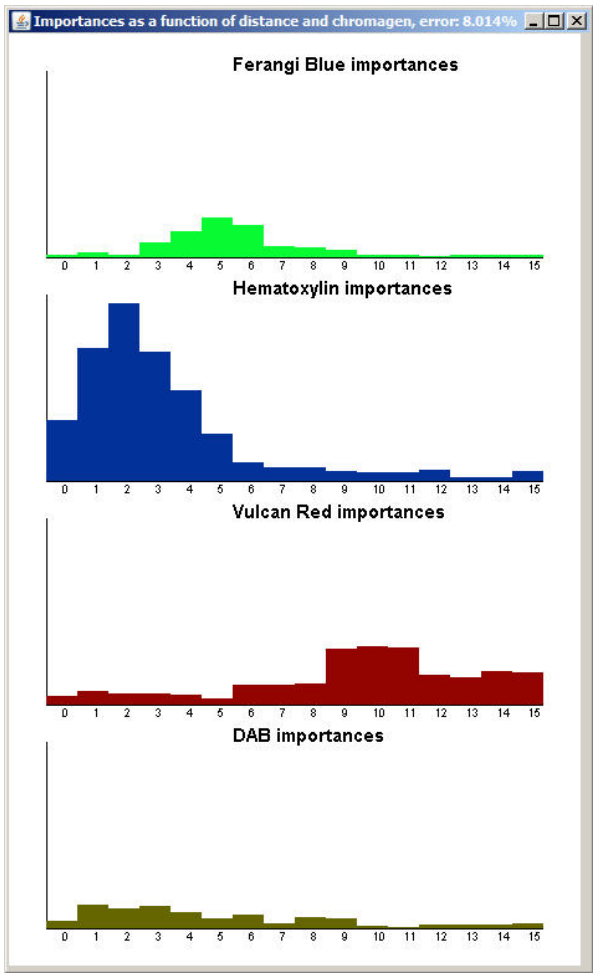

Figure 9: Bar charts for interpreting the distance at which the chromagens influence the classifier. Each plot represents a chromagen with the x-axis indicating the ring score's radius and the y-axis indicating relative importance (with the largest bar being the most important). The Hematoxylin was found to be very important at $r \in [0, \ldots, 5]$. The forest learned that cancer cell nuclei, T-cell nuclei, and unspecific cell nuclei all have hematoxylin-rich centers. Ferangi Blue was found to be most important at $r \in [4, \ldots, 6]$ indicating that the classifier learned that the T-cells are positive at their membranes. Vulcan Red was found to be most important at $r \in [6, \ldots, 15]$ indicating that cancer cells are positive on their membranes and their diameter varies dramatically. DAB was found to be important at $r \in [1, \ldots, 6]$ indicating that dendritic cells appear devoid of a nucleus and vary in size.

The user then transitions back to the "phenotype training" tab to view the results (see Figures 10 and 11).

3.4. Retraining

GemIdent undoubtedly made mistakes during its initial classification and the user will wish to improve the results by retraining over the mistakes (see Section 2, Step 6).

The software has many tools that aid the user in retraining:

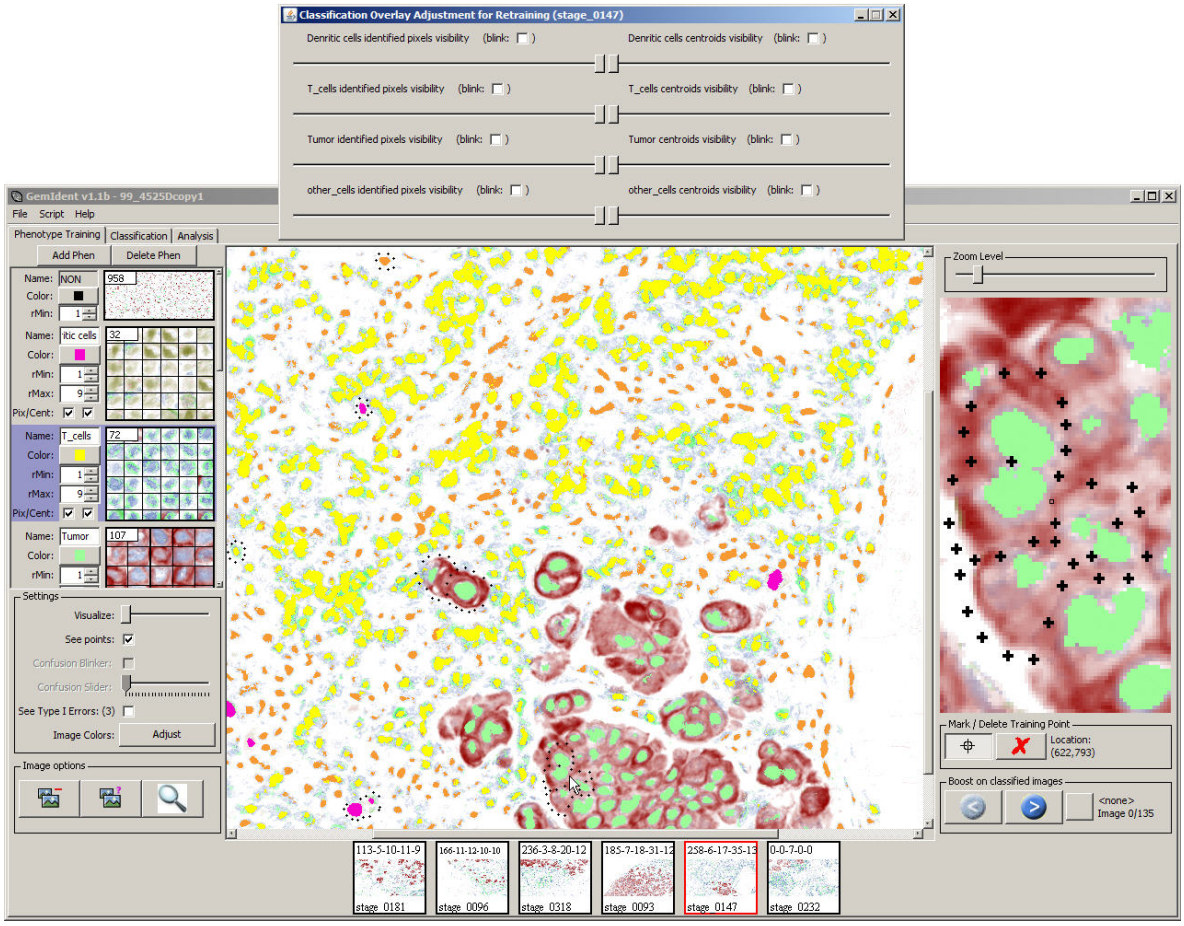

Figure 10: After the training sets have been classified, the pixels in the different phenotypes appear as blobs. The result masks are overlaid atop the original image.

- Alpha sliders for pixel and centroid masks:

  Figure 10 shows the pixel masks overlaid and Figure 11 shows the centroid masks overlaid atop the original image. Both screenshots include the "overlay adjustment" dialog box (top). The sliders in this dialog can be adjusted to vary the opacity of the mask. When the masks are translucent, the user can view the underlying tissue and the classification results *simultaneously*, and **GemIdent**'s errors can be located with ease. The "blink" checkboxes in the dialog box are used when the user feels translucent viewing does not adequately display both the tissue and the mask and wishes to flash the mask on/off.

- Localizing type I errors:

  Points where the user trained positive for a phenotype, but after classification did not yield a nearby centroid, are called "type I errors." **GemIdent** has an option to visualize these errors so the user can train over them (see Figure 12).

- Adding other images to the training set Images not specifically in the training set can be observed and if errors are found, they can be added to the training set. This ensures

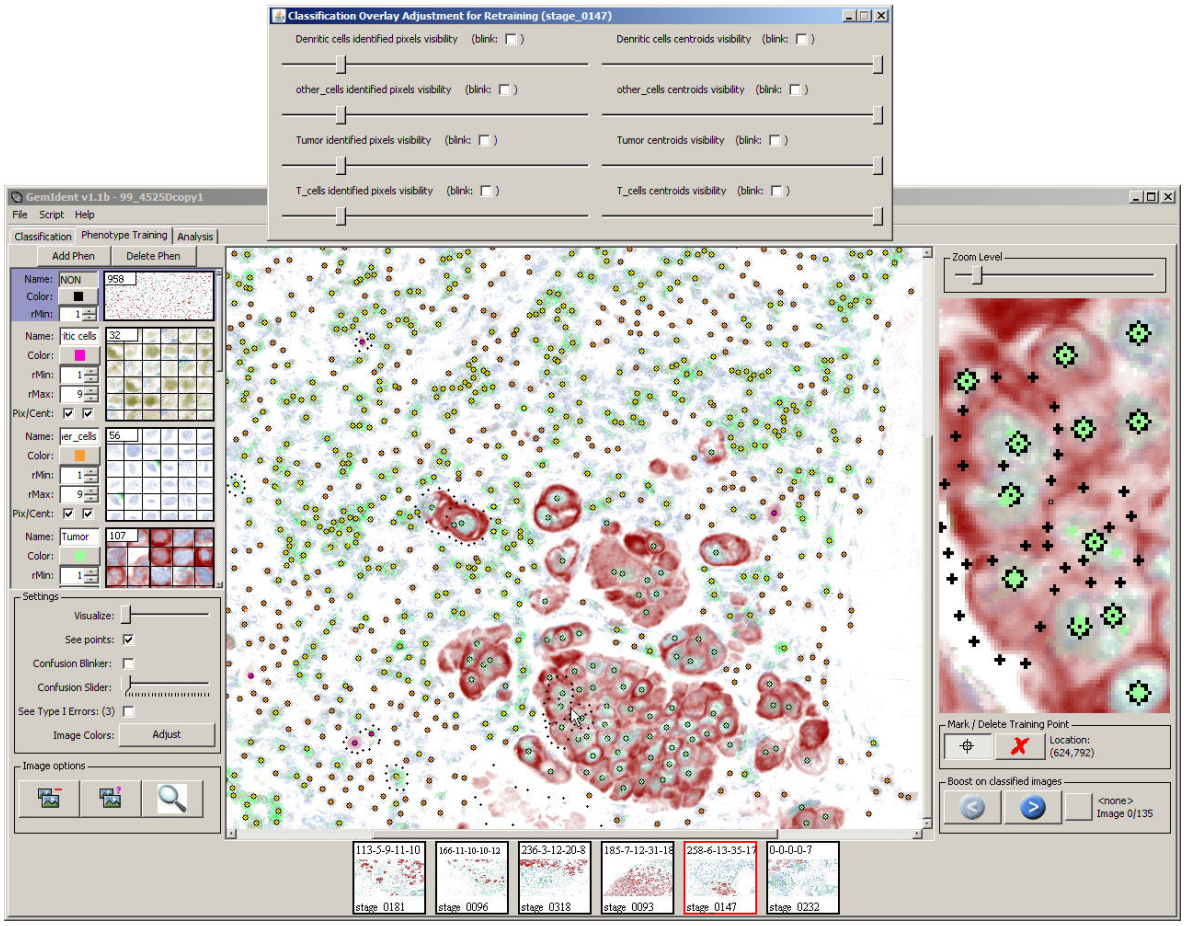

Figure 11: The cell centroids are marked by black stars filled with the relevant phenotype's color. The result masks are overlaid atop the original image.

that the training set becomes more representative of the entire image set, which is a type of boosting. **GemIdent** provides a convenient option where these images can be cycled through and perused quickly.

After retraining, the image set can be reclassified and reviewed. The user can iterate until satisfied with the error rates.

3.5. Simple analyses and reporting

The data analysis tab (see Figure 13) supports basic queries, displays histograms, and generates summary reports.

3.6. Data analysis with R

During classification, the centroid data is written to a text file named after the particular phenotype they belong to with 5 columns: the stage number (which is the subimage to which the cell belonged), and the local and global X and Y coordinates. For instance, the file `99_4525D-Tumor.txt` contains:

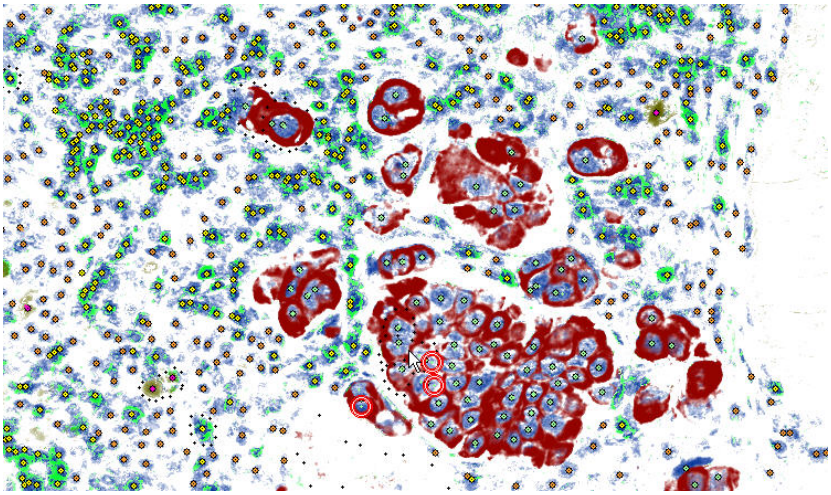

Figure 12: Type I errors are displayed surrounded by red circles.

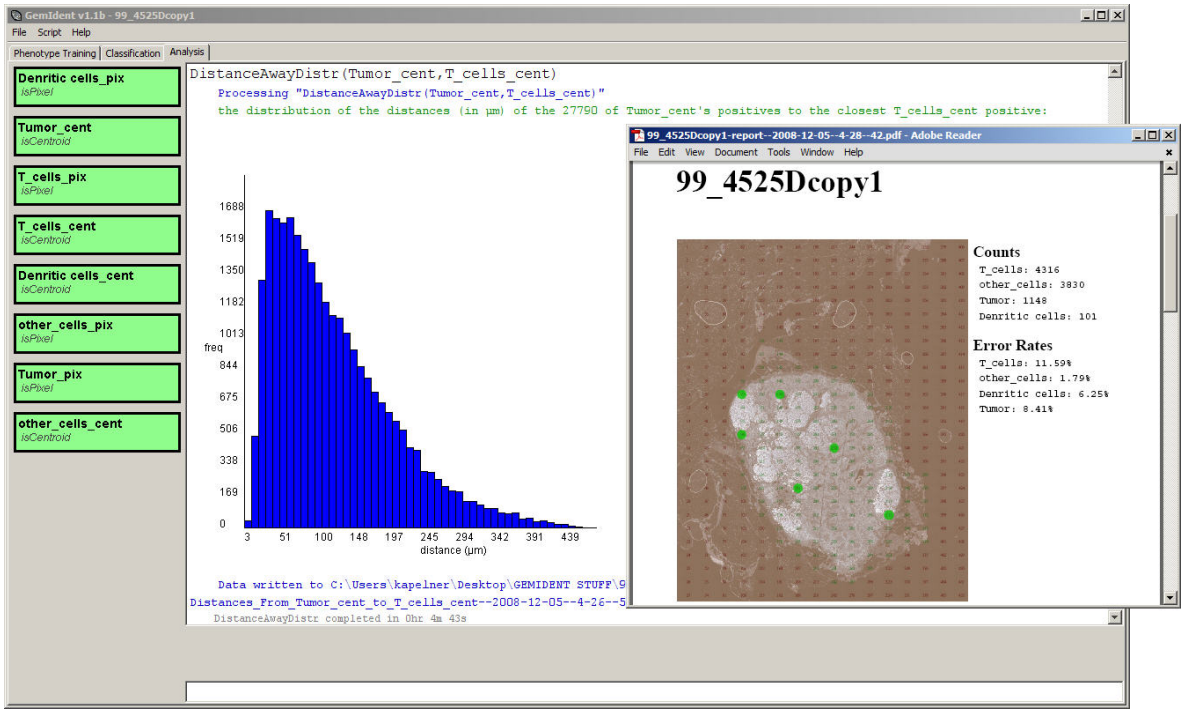

Figure 13: The data analysis tab. The left panel displays the images currently in memory. By default, pixel and centroid results are loaded. The main window shows the result of a query asking for the distribution of distances from Tumor cells to their nearest T-cell neighbors. The open PDF document is the autogenerated report which includes a thumbnail view of the entire image set, counts and type I error rates for all phenotypes, as well as a transcript of the analyses performed.

```
filename,locX,locY,globalX,globalY
stage_0096,201,51,4040,13037
stage_0096,214,91,4053,13077
stage_0096,220,76,4059,13062
stage_0096,230,107,4069,13093
.....................
```

This data is easily read into R packages such as **spatstat** (Baddeley and Turner 2005) and transformed into an object of the ppp class. As in this short example:

```
R> DCs <- read.delim2(DCname, sep = ",", header = TRUE)
R> other <- read.delim2(othername, sep = ",", header = TRUE)
R> Tums <- read.delim2(Tumorname, sep = ",", header = TRUE)
R> Tcells <- read.delim2(tcellname, sep = ",", header = TRUE)
R> list.tcells <- Tcells[,4:5]
R> list.tumor <- Tums[,4:5]
R> list.DCs <- DCs[,4:5]
R> list.all <- rbind(other[,4:5], Tums[,4:5], Tcells[,4:5], DCs[,4:5])
R> list.maxs <- apply(list.all, 2, max)
R> list.counts <- c(nrow(other), nrow(Tums), nrow(Tcells), nrow(DCs))
R> list.slides <- ppp(list.all[,1], list.maxs[2] - list.all[,2],
+    window = owin(c(0, list.maxs[1]), c(0, list.maxs[2])),
+    marks = as.factor(c(
+      rep("Other", list.counts[1]), rep("Tumor", list.counts[2]),
+      rep("Tcells", list.counts[3]), rep("DCs", list.counts[4]))))
R> list.Wins <- levelset(density(list.slides, 200), 10^(-4), ">")
R> list.slidesW <- ppp(list.all[,1], list.maxs[2] - list.all[,2],
+    window = list.Winsl, marks = as.factor(c(
+      rep("Other", list.counts[1]), rep("Tumor", list.counts[2]),
+      rep("Tcells", list.counts[3]), rep("DCs", list.counts[4]))))
R> plot(density(list.slidesW, 200))
R> denst <- density(
+    list.slidesW[which(list.slidesW[[i]]$marks == "Tumor")], 200)
R> plot(denst, main = "Tumor and DCs as dots", cols = tim.colors(32))
R> contour(denst, levels = quantile(denst$v, 0.95, na.rm = TRUE),
+    lwd = 1, add = TRUE)
R> text(list.slidesW[which(list.slidesW$marks == "DCs")], ".")
```

The analysis performed on these data show the spatial landscape of the lymph node immune cells quite clearly (see Figure 14).

4. Conclusions and future uses

The success of the method resides on the collection of all the data features in a neighborhood of a pixel, and the selection by random forests of the pertinent features for each particular

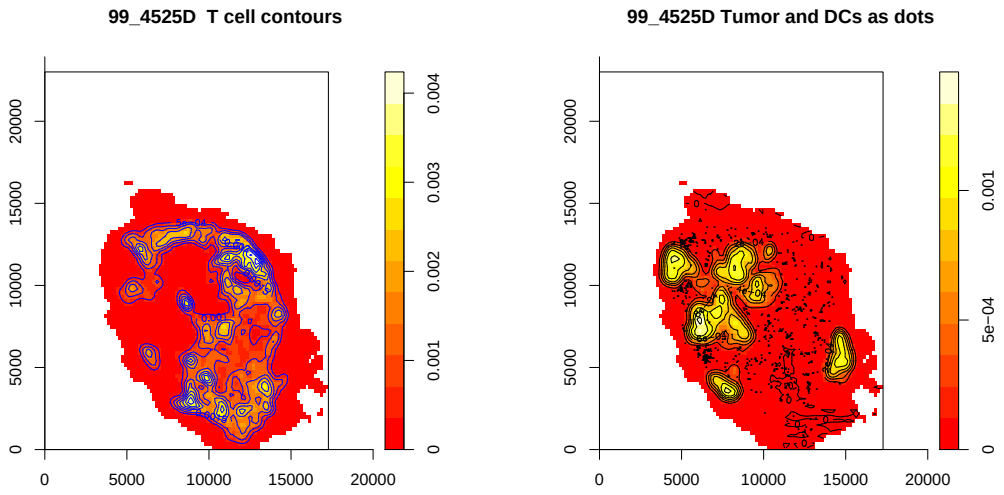

Figure 14: Output from the estimation of spatial densities using kernel density estimates from **spatstat** (Baddeley and Turner 2005).

phenotype. The iterative boosting enables the user to increase accuracy to a satisfactory level.

Although we have concentrated here on static multispectral images, fluorescent images can be classified in a similar way. Instead of using distributions in colorspace to obtain scores, density estimation can be used to compute scores in the unidimsensional space of the fluorescent layer intensity images.

Furthermore, the algorithm is not only restricted to static images: Film is a composition of images called "frames" displayed over time. Identification can be done in moving images using the changing frames as the "z-axis" and instead of scores computed via sums of rings, it can be sums of sphere-surfaces. The algorithm can also be generalized to phenotype identification in n-dimensions. The algorithm also may be applied to identification of objects in satellite imagery, face recognition, automatic fruit harvesting and countless other fields.

There is no doubt that images will continue to provide data to solve many biological mysteries. The **GemIdent** project is a step towards combining human expertise and statistical learning to supply a feasible and objective data collection process in the presence of high case to case variability.

Acknowledgments

We thank Holbrook Kohrt for many useful insights and recommendations, Adam Guetz for a careful reading of the manuscript and some good ideas on how to improve future versions. We thank Kyle Woodward for help designing and implementing a GUI for **GemIdent** and Francesca Setiadi for the data collection and helpful suggestions. We thank CVSDude.com for providing a home for **GemIdent**'s source code. The referees and editors for JSS provided comments which helped improve the paper. This work was funded by a DOD Era Hope Scholar grant to PPL and by awards NSF-DMS 02-41246 and NSF/NIGMS R01GM086884 to SH.

References

Baddeley A, Turner R (2005). "**spatstat**: An R Package for Analyzing Spatial Point Patterns." *Journal of Statistical Software*, **12**(6), 1–42. URL http://www.jstatsoft.org/v12/i06/.

Breiman L (2001). "Random Forests." *Machine Learning*, **45**(1), 5–32.

Chakraborty A (2003). *An Attempt to Perform Bengali Optical Character Recognition*. Ph.D. thesis, Stanford University.

Collins TJ (2007). "**ImageJ** for Microscopy." *BioTechniques*, **43**(S1), S25–S30. doi:10.2144/000112517.

Fang B, Hsu W, Lee ML (2003). "On the Accurate Counting of Tumor Cells." *IEEE Transactions on Nanobioscience*, **2**(2), 94–103.

Freund Y, Schapire R (1997). "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting." *Journal of Computer and System Sciences*, **55**(1), 119–139.

Gil J, Wu H, Wang BY (2002). "Image Analysis and Morphometry in the Diagnosis of Breast Cancer." *Microscopy Research and Technique*, **59**, 109–118.

Gonzalez RC, Woods RE, Eddins SL (2004). *Digital Image Processing Using MATLAB*, pp. 245–255, 352–364. Pearson Education.

Hastie T, Tibshirani R, Friedman J (2001). *The Elements of Statistical Learning*. Springer-Verlag, New York.

Ihaka R, Gentleman R (1996). "R: A Language for Data Analysis and Graphics." *Journal of Computational and Graphical Statistics*, **5**(3), 299–314.

Kapelner A, Holmes S, Lee PP (2007a). "**GemIdent**, Version 1.1." URL http://www.GemIdent.com/.

Kapelner A, Lee PP, Holmes S (2007b). "An Interactive Statistical Image Segmentation and Visualization System." *Medivis*, pp. 81–87. doi:10.1109/MEDIVIS.2007.5. IEEE Conference on Medical Visualization – BioMedical Visualisation (Medivis 2007).

Kohrt HE, Nouri N, Nowels K, Johnson D, Holmes S, Lee PP (2005). "Profile of Immune Cells in Axillary Lymph Nodes Predicts Disease-Free Survival in Breast Cancer." *PLoS Medicine*, **2**(9), e284.

Kovalev V, Harder N, Neumann B, Held M, Liebel U, Erfle H, Ellenberg J, Neumann B, Eils R, Rohr K (2006). "Feature Selection for Evaluating Fluorescence Microscopy Images in Genome-Wide Cell Screens." *IEEE Transactions on Biomedical Engineering*, **1**, 276–283. doi:10.1109/CVPR.2006.121.

Lee JB, Woodyatt AS, Berman M (1990). "Enhancement of High Spectral Resolution Remote Sensing Data by a Noise-Adjusted Principal Components Transform." *Geoscience and Remote Sensing*, **28**, 295–304.

Levenson RM (2006). "Spectral Imaging Perspective on Cytomics." *Cytometry*, **69A**(7), 592–600.

Maggioni M, Warner FJ, Davis GL, Coifman RR, Geshwind FB, Coppi AC, DeVerse RA (2004). "Algorithms from Signal and Data Processing Applied to Hyperspectral Analysis: Application to Discriminating Normal and Malignant Microarray Colon Tissue Sections." *Technical Report 1311*, Yale University. URL http://www.math.duke.edu/~mauro/Papers/ColonCarcinoma.pdf.

Mahalanobis A, Kumar BVKV, Sims SRF (1996). "Distance-Classifier Correlation Filters for Multiclass Target Recognition." *Applied Optics*, **35**(17), 3127–3133. doi:10.1364/AO.35.003127.

Ortiz de Solirzano C, Garcea Rodriguez E, Jones A, Pinkel D, Gray JW, Sudar D, Lockett SJ (1999). "Segmentation of Confocal Microscope Images of Cell Nuclei in Thick Tissue Sections." *Journal of Microscopy*, **193**(3), 212–226.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Sklyar O, Huber W (2006). "Image Analysis for Microscopy Screens." *R News*, **6**(5), 12–16. URL http://CRAN.R-project.org/doc/Rnews/.

Wikipedia (2009). "Flood Fill — Wikipedia, The Free Encyclopedia." URL http://en.wikipedia.org/wiki/Flood_fill, accessed 2009-04-12.

Wu HS, Barba J, Gil J (1998). "A Parametric Fitting Algorithm for Segmentation of Cell Images." *IEEE Transactions on Biomedical Engineering*, **45**(3), 400–407.

Wu K, Gauthier D, Levine MD (1995). "Live Cell Image Segmentation." *IEEE Transactions on Biomedical Engineering*, **42**(1), 1–12.

Yang Q, Parvin B (2003). "Harmonic Cut and Regularized Centroid Transform for Localization of Subcellular Structures." *IEEE Transactions on Biomedical Engineering*, **50**(4), 469–475.

A. Simple blob analysis algorithm

The first step of the blob analysis is the creation of heuristic rules built from the training data:

Step 1 For all points $\mathbf{t} \in T$ (the training set for a phenotype), we verify if they are inside a blob corresponding to this phenotype. If so, use the floodfill algorithm(Wikipedia 2009) to extract the containing blob's coordinates. The collection of these containing blobs is represented by: $\Omega = \{\omega_1, \omega_2, \ldots, \omega_N\}$ These are our reference blobs.

Step 2 For each of the reference blobs, count the number of pixels contained within and record the sizes in the vector $\mathbf{v}$. Now, record reference statistics about: the low threshold value, v_L, (**GemIdent** uses the 2^{nd} percentile), the median, v_M, and the upper threshold value, v_H (**GemIdent** uses the 95^{th} percentile).

We are now going to use the insight into blob sizes in the reference statistics to find centroids for the blobs obtained from the classification:

Step 1 A floodfill algorithm (Wikipedia 2009) is used to extract all blobs, to create the collection Ω.

Step 2 For each blob in the collection, ω, measure its size, v. If $v < v_L$, ignore it — the blob is too small and is most likely noise. If $v_L \leq v \leq v_H$, find the (x, y) center of ω and set $C(x, y)$ true (indicating the presence of a centroid). In a normal application, where the phenotypes in the image are fairly spaced apart, these two tests will cover greater than 90% of the blobs. If $v > v_H$, ie the blob is large, it must be split and multiple centroids must be marked. Proceed to Step A.

Step A Define $n = \text{floor}(v/v_M)$. Cut the large blob into n sub-regions using a square mask of semiperimeter $s = \sqrt{\frac{v}{\frac{n}{\pi}}}$, the radius of the average circle if the large blob's pixels were split into discs. Mark the centers of each of these cut squares in C.

In fact we use several levels of such statistics to improve the centroid calculations.

Affiliation:

Susan Holmes
Department of Statistics
Sequoia Hall
Stanford
CA 94305, United States of America
E-mail: susan@stat.stanford.edu
URL: http://www-stat.stanford.edu/~susan/

Adam Kapelner, Peter P. Lee
Hematology
Stanford Medical School
Stanford
CA 94305, United States of America

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 30, Issue 10 *Submitted:* 2008-08-09
June 2009 *Accepted:* 2009-01-15