

Extracting information from text and images for location proteomics

Zhenzhen Kou
Center for Automated Learning &
Discovery
Carnegie Mellon University
Pittsburgh, PA
woomy@cs.cmu.edu

William W. Cohen
Center for Automated Learning &
Discovery
Carnegie Mellon University
Pittsburgh, PA
wcohen@cs.cmu.edu

Robert F. Murphy
Center for Automated Learning &
Discovery &
Dept. of Biological Sciences
Carnegie Mellon University
Pittsburgh, PA
murphy@cmu.edu

ABSTRACT

There is extensive interest in automating the collection, organization and summarization of biological data. Data in the form of figures and accompanying captions in literature present special challenges for such efforts. Based on our previously developed search engines to find fluorescence microscope images depicting protein subcellular patterns, we introduced text mining and Optical Character Recognition (OCR) techniques for caption understanding and figure-text matching, so as to build a robust, comprehensive toolset for extracting information about protein subcellular localization from the text and images found in online journals. Our current system can generate assertions such as “Figure N depicts a localization of type L for protein P in cell type C”.

Keywords

Information extraction, Bioinformatics, text mining, image mining, fluorescence microscopy, protein localization

1. INTRODUCTION

The vast size of the biological literature and the knowledge contained therein makes it essential to organize and summarize pertinent scientific results. This leads to the creation of curated databases, like the Entrez databases, SwissProt, and YPD. The information in these databases is largely incorporated by computer-generated links to relevant entries in other structured databases or entered manually by scientists in the relevant fields. However, curated databases are expensive to create and maintain. Moreover, they do not typically permit extensive links to specific supporting data, do not estimate confidence of assertions, do not allow for divergence of opinion, and do not readily permit updating or reinterpretation of previously entered information.

Information extraction (IE) methods can be used to at least partially overcome these limitations by creating self-populating knowledge bases that can automatically extract and store assertions from biomedical text [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. However, most existing IE systems are limited to extracting information only from text, not from image data. In this paper we describe techniques for extracting information about *protein subcellular locations* from both text and images.

These techniques build on previous work [12] in using image processing methods to analyze fluorescence microscope images

and extract a quantitative description of the localization patterns of the tagged proteins. This work was later extended to process images harvested from on-line publications [13]. Here we will describe a further extension to this system, which extracts detailed textual annotations of the images (and associated proteins) by analyzing the accompanying captions. The system is called SLIF (for Subcellular Location Image Finder), and our long-term goal is to develop a large library of annotated and analyzed fluorescence microscope images, in order to support data-mining.

More generally, there are many reasons for wishing to investigate extraction from the text and images in figures. Figures occupy large amounts of valuable page space, and are likely to be seen disproportionately by casual readers. Thus figure and caption pairs often concisely summarize a paper’s most important results as perceived by the author.

In the following sections, we will first describe how SLIF works. We will focus on the recent extensions to the system; specifically our approaches to extracting information from caption text, and associating this information with the image data.

2. SLIF

2.1 Overview

SLIF applies both image analysis and text interpretation to the figure and caption pairs harvested from on-line journals, so as to extract assertions such as “Figure N depicts a localization of type L for protein P in cell type C”. The protein localization pattern L is obtained by analyzing the figure, the protein name and cell type are obtained by analysis of the caption. Figure 1 illustrates some of the key technical issues. The figure encloses a prototypical figure harvested from a biomedical publication,¹ and the associated caption text. Note that the text “Fig. 1 Kinase...experiments” is the associated caption from the journal article, and that the figure contains several panels (independently meaningful subfigures).

¹ This figure is reproduced from the article “Ras Regulates the Polarity of the Yeast Actin Cytoskeleton through the Stress Response Pathway”, by Jackson Ho and Anthony Bretscher, *Molecular Biology of the Cell* Vol. 12, pp. 1541–1555, June 2001.

Fig. 1. Kinase inactive Plk inhibits Golgi fragmentation by mitotic cytosol. (A) NRK cells were grown on coverslips and treated with 2mMthymidine for 8 to 14 h. Cells were subsequently permeabilized with digitonin, washed with 1M KCl-containing buffer, and incubated with either 7 mg/ml interphase cytosol (IE), 7mg/ml mitotic extract (ME), or mitotic extract to which 20 mg/ml kinase inactive Plk (ME + Plk-KD) was added. After a 60-min incubation at 32C, cells were fixed and stained with anti-mannosidase II antibody to visualize the Golgi apparatus by fluorescence microscopy. (B) Percentage of cells with fragmented Golgi after incubation with mitotic extract (ME) in the absence or the presence of kinase inactive Plk (ME + Plk-KD). The histogram represents the average of four independent experiments.

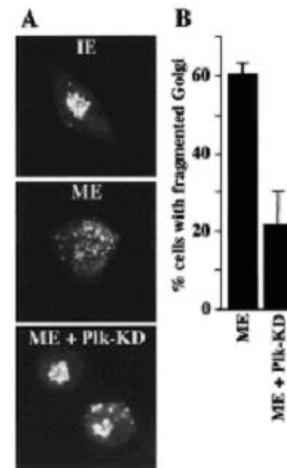


Figure 1: A figure caption pair reproduced from the biomedical literature.

Figure caption pair analysis involves several distinct tasks. The first is to extract all figure and caption pairs from articles in on-line journals and to identify those that depict fluorescence microscope images. The second is to identify numerical features that adequately capture information about subcellular location. The third is extraction of protein names and cell types from captions. The fourth is mapping the information extracted from the caption to the right sections of the figure. We will provide a brief review of SLIF in this section.

Figure 2 shows the overall structure of SLIF. Tasks described in light grey characters represent future or ongoing work. Not illustrated are supporting tools for browsing and querying the extracted information, which are also under development; however, flexible tools for accessing extracted information are also extremely important, since often only part of the information present in a figure is extracted.

The original SLIF system used a web robot to automatically retrieve PDF versions of online journal articles from PubMed Central that matched a particular query. Figures and accompanying captions were extracted and paired together using a modified version of PDF2HTML, a public domain tool. The figure-extraction step achieved a precision (number of correct figure-caption pairs returned divided by the number of figure-caption pairs returned) of 98% and a recall (number of correct pairs returned divided by the number of actual pairs) of 77%. The new version of SLIF includes web robots to extract papers from sources such as BioMedCentral, and we also have obtained an extensive collection of articles directly from the publisher. These sources are in XML format, so figure/caption pairs can be extracted without errors.

2.2 Figure processing

2.2.1 Decomposing figures into panels

For figures containing multiple panels, the individual panels must be recovered from the figure. In the current system, figures are decomposed into panels by recursively subdividing the figure by looking for horizontal and vertical white-space partitions. The system achieves a precision of 73% and a recall of more than 60% on these steps.

2.2.2 Identifying fluorescence microscope images

Once panels have been identified, it is necessary to determine what sort of image the panel contains, so that appropriate image processing steps can be performed. In the current system, panels are classified as to whether they are fluorescence microscope images (micrograph) using the grey-scale histogram as features. The *k*-nearest neighbor classifier used for this task achieves a precision of 97% and a recall of 92%.

2.2.3 Image preprocessing before feature computations

2.2.3.1 Annotation detection and removal

Many micrographs contain annotations such as labels, arrows and indicators of scale, within the image itself. They must be detected, analyzed, and then removed from the image. Annotation detection relies on finding areas that are bright and have sharp edges. Annotation removal consists of filling the annotation area with background pixel values. On a test set of 100 fluorescence microscope panels, this step achieves precision of 83% and recall of 82%.

2.2.3.2 Multi-cell image segmentation

Many (if not most) published fluorescence microscope images contain more than one cell and our methods for classifying subcellular location patterns require images of a single cell. Each micrograph is therefore segmented into individual cells by a “seeded watershed” algorithm [13]. The seeded watershed segmentation works well for some location classes (e.g., tubulin, with 52% precision and 41% recall) but is not expected to work well for others (e.g., Golgi, with 62% precision but only 32% recall). Improving this step is a subject of current research.

2.2.3.3 Determining the scale of each micrograph

Automated analysis of fluorescence microscope images requires knowing the scale of an image since some of our previously developed subcellular location features (SLF) directly depend on the scale of the images. Imaging processing techniques are used to locate the *scale bar* associated with a panel. The size of the scale bar is extracted from the accompanying caption. Scale bar extraction is currently done with a precision of 76% and a recall of 50%. Improving this step is a subject of current research.

2.2.3.4 Subcellular location pattern computation

Finally, SLFs are produced that summarize the localization pattern of each cell. We have extensively characterized a

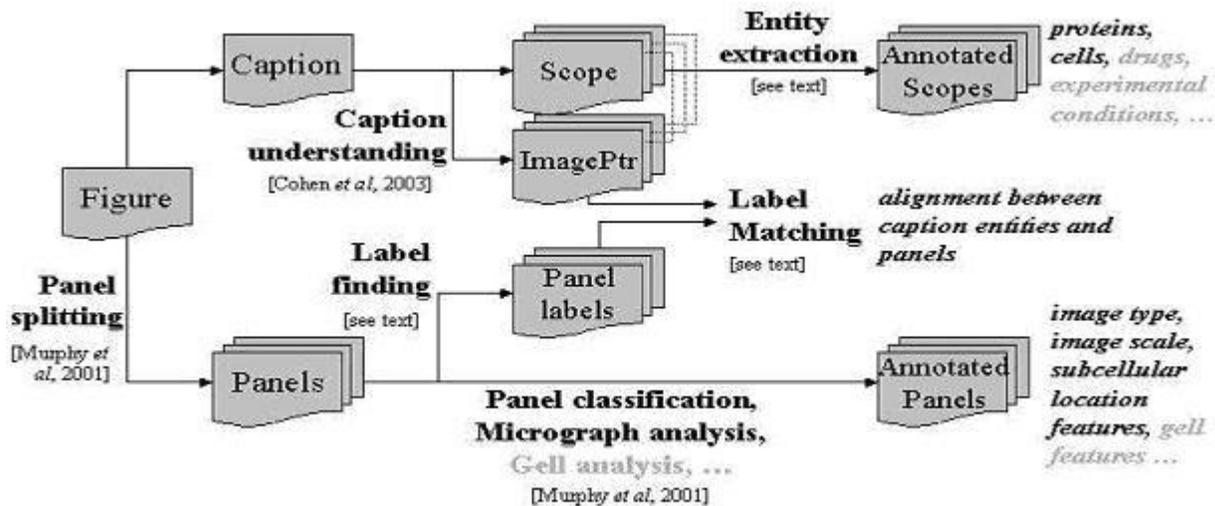


Figure 2: Diagram of SLIF.

number of feature types for this purpose and have achieved over 92% accuracy using these features on single cells [12, 13]. Our work in [13] demonstrated the feasibility of determining the preparation, cell type and microscopy method, and image alterations introduced during publication.

2.3 Caption processing

2.3.1 Entity name extraction

Caption interpretation aims to identify the name and cell type of the visualized protein in each microscope image. These extraction tasks have been heavily studied in the literature; however, there are still few publicly-available extraction systems. Rather than expend substantial resources to developing our own extractors, for the current version of SLIF we hand-coded some relatively simple extraction methods for this task. Protein names tend to be either single words with upper case letters, numerical figures, and non-alphabetical letters such as “Nef”, “p53”, or compound words with upper case letters, numerical letters, and non-alphabetical letters such as “Interleukin 1 (IL-1)-responsive kinase”, or single lowercase words ending in -in or -ase such as “actin”, “tubulin”, “insulin”. Similar rules are used to identify cell type. The protein-name extractor obtains a precision of 63% and a recall of 95%, and the cell-type extractor obtains a precision of 85% and recall of 92%.

2.3.2 Entity to panel alignment

To integrate the features obtained via figure processing and entity names extracted from captions, *entity to panel alignment* must be done. The goal here is to determine, for each entity extracted from the caption, to which panel that entity is associated. The linkage between the images which are the figure panels and the text of captions is usually based on textual labels which appear as annotations to the images, and which are also interspersed with the caption text. So, entity to panel alignment is based on extracting the labels from panels, and extracting the corresponding *image pointers* from captions. *Image pointers* are strings in the caption that refer to places in the accompanying images, for example, “A”, and “B”, in Figure 1.

In analyzing caption text, we decided to break down the task of entity to panel alignment into several subtasks. The first step is *image-pointer extraction*. After image pointers are extracted, they are *classified* according to their linguistic function. *Bullet-style* image pointers function as compressed versions of bulleted lists, for example, the strings “(A)” and “(B)” in figure 1. *NP-style* image pointers are used as proper names in grammatical text, for example, the string “(A)” in the text: “Following a procedure similar to that used in (A), ...”. *Citation-style* image pointers are interspersed with grammatical caption text, in the same manner that bibliography citations are interspersed with ordinary text. The remaining image pointers in Figure 1 are citation-style.

We combined the steps of extraction and classification, as follows. Most image pointers are parenthesized, and relatively short. We thus hand-coded an extractor that finds all parenthesized expressions that are (a) less than 40 characters long and (b) do not contain a nested parenthesized expression, and also extracts all whitespace-surrounded expressions of the form “x”, “X”, “x-y” or “X-Y” that are preceded by one of the words “in”, “from”, or “panel”. This extractor has high recall (98%) but only moderate precision (74.5%) on the task of finding image pointers.

Using a classifier trained with machine learning approaches, we then classify extracted image pointers as bullet-style, citation-style, NP-style, or “other”. Image pointers classified as “other” are discarded, which compensates for the relatively low precision of the extractor. This classifier has an overall accuracy of 87.8%. Performance is extremely good (recall of 98% and precision of 94.6%) on bullet-style labels, which are the ones most likely to severely impact performance. Most errors are made by incorrectly rejecting citation-style image pointers [14].

After image-pointer classification, the *scope* of each image pointer is determined. The scope of an image pointer specifies, indirectly, what text should be associated with that image pointer. The scope of an NP-style image pointer is the set of words that (grammatically) modify the proper noun it serves as.

The scope of a bullet-style image pointer is all the text between it and the next bullet-style image pointer. The scope of a citation-style image pointer is some sequence of tokens around the image pointer, usually corresponding to a nearby noun phrase--currently approximated with heuristic hand-coded methods.

In the remainder of the paper, we will discuss in detail the methods used to find the panel-label annotations which appear in images, and the methods used to match these annotations to image pointers. In the remainder of this section, we will briefly review how image pointers are found and associated with extracted entities [14].

3. PANEL LABEL EXTRACTION AND PANEL-TEXT MATCHING

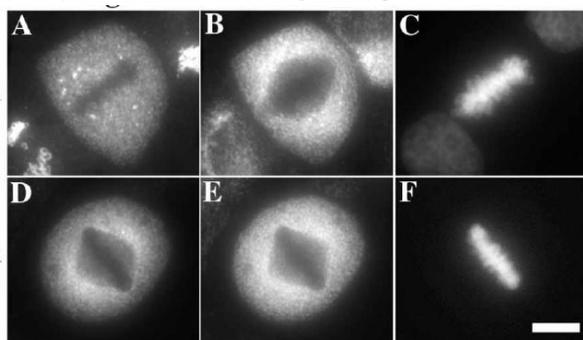


Figure 3: Panels with internal labels.

Extracting panel labels and mapping information derived from captions to panels are crucial steps in SLIF since it serves as the bridge between image analysis and caption interpretation. Since panels containing an internal label (such as “A”~”F” in Figure 3) hold the vast majority, we first focus our work on extracting these internal labels.

Automatic detection and recognition of panel labels is a challenging problem because the label is usually a single character embedded in the panel, and the background might be complex. However, current OCR (Optical Character Recognition) technology is largely restricted to finding text printed against clean backgrounds, and cannot handle text printed against shaded or textured background, or embedded in images directly [15, 16]. Our current system applies a four-stage strategy to the label contained within the panel itself, *text detection*, where a segmentation scheme is used to focus attention on regions where a panel label may occur, *image enhancement*, where the text region is enhanced by increasing the resolution and converting the gray image to a binary image, *OCR (Optical Character Recognition)*, where the enhanced text image is passed through an OCR engine for recognition, *string match*, where the OCR results of all the panels in one figure are passed through the list of panel labels obtained by interpreting the caption associated with this figure, so that the possible missing (or incorrect) labels from OCR could be corrected. The final step also serves as the mapping between labels recognized from the image and image pointers obtained by caption

interpretation so as to combine the information extracted from figure and caption.

Below we will describe results of our four-stage strategy for panel label extraction and panel-text matching. These experiments are based on a dataset of 427 hand-labeled panels from 95 randomly-chosen Pubmed Central papers.

3.1 OCR directly on panels

Directly running the OCR software we are using, GOCR[17], on panels only yielded 15 labels; this emphasizes the point that current OCR software is not well suited to recognizing text embedded in images.

3.2 Text detection

Because characters usually form regions of high contrast against the background, a typical text region can be characterized as a rectangular region with a high density of sharp edges. Therefore our text detection method relies on finding areas that have sharp edges. We used the Roberts method [18] for edge detection. Applying edge detection to the original panel (image A) resulted in a binary image B. Image B contains the edges of the labels as well as some noise. We noticed that the noise usually consisted of short line segments while label edges were represented by longer continuous regions or short nearly-connected segments; for example the edges for a letter “a”, might be disjoint. We therefore used a two-stage process to reduce noise. We first closed the binary image using a 3x3 pixel structural element to connect the disjoint sections making up the edges of the labels. Then we removed any objects of size 25 pixels or less to delete any remaining noise. This results in a binary image C in which connected regions have a high density of sharp edges. As an example, Figure 4 shows the process of text detection from A to C. The text region appeared as a connected component in C.

We then bounded the connected components in C with their maximum and minimum coordinates in the x and y directions to get candidates for text regions. Several constraints were then applied to filter out candidates that are not text regions.

- Since a panel label is usually a single letter, the ratio between the height and the width of a text region should be in a certain range. We only kept regions with this ratio greater than or equal to 0.3, i.e. we discarded horizontally strip-shaped regions.
- Panel labels are usually located in one corner of the panel, so the distance between the boundary of a text region and the panel boundary should be small. We only kept regions with this horizontal/vertical distance less than 1/10 of the width/height of the panel.
- Panel labels are usually small in size compared with the panel, so the area of the text region should be in a certain range. Experimentally, we noticed that the height/width of the text region ranged between $1/20^{\text{th}}$ and $1/4^{\text{th}}$ of the height/width of the panel, so we only kept regions with areas between $1/20^2$ and $1/4^2$ of the area of the panel.

The experimental results are shown in Table 1. 380 of the 467 candidate text regions were correctly detected. While 81.3% precision appears low, most of the regions incorrectly considered to be text regions do not contain characters at all. They therefore do not yield any characters during OCR and do not affect the final set of panel labels.

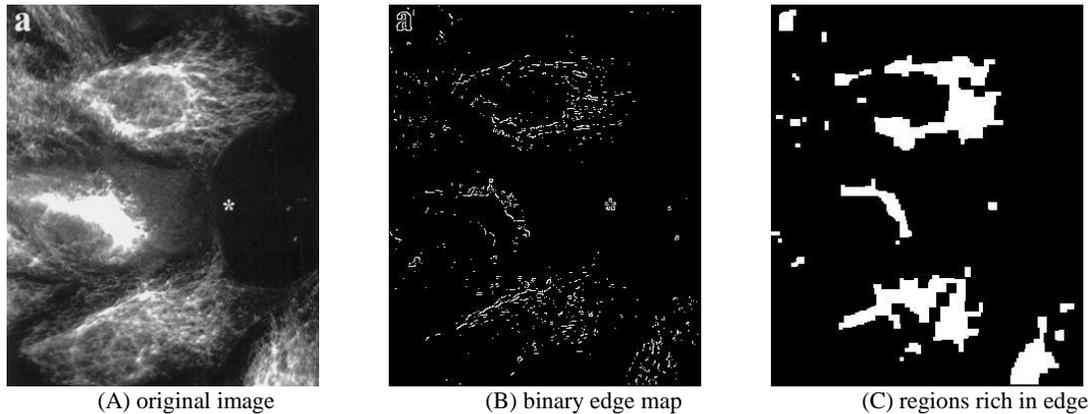


Figure 4: Process of text detection.

No. of panels	Total predicted 'text' region	Actual text region	Precision	Recall
427	467	380	81.3%	89.0%

Table 1: Text detection result.

3.3 Intensity normalization

Because GOCR assumes the character to be black font on a white background, we must identify whether the text appears as a black font in a white background or a white character in a black background before applying OCR. In panel images, the intensity value of text may be lower or higher than that of the background. For the later, the intensity of text images should be inverted before running GOCR. We call this procedure *intensity normalization*. Our normalization method is as follows: First, we choose the top 20% pixels and the bottom 15% pixels of a text region, where pixels are mainly background pixels, and calculate the mean value m_1 of these pixels. Then we choose the middle area from $0.3h$ to $0.65h$ (h is the height of text region) and calculate the mean value m_2 . If $m_1 < m_2$, we consider the intensity of text to be lower than that of the background; otherwise the intensity of text is considered to be higher than that of the background.

3.4 Image enhancement

Applying GOCR to normalized text region obtained a precision of 71.3% and a recall of 63.5%. We hypothesized that most errors are because GOCR is designed for recognizing high resolution text printed against clean backgrounds. In order to increase the recognition rate, we introduced an interpolation method and a binarizing algorithm to increase the image quality.

3.4.1 Sub-pixel Interpolation

One crucial condition for GOCR success is sufficient resolution of the input image. GOCR prefers fonts of 20~60 pixels. However, label containing areas in figures can be less than 20 x 20 pixels. GOCR usually fails on such low resolution images. To obtain higher resolution images, we expanded regions smaller than 20 x 20 pixels by applying bicubic interpolation [19]. Bicubic interpolation estimates the grey value at a pixel in the destination image by an average of 16 pixels surrounding the closest corresponding pixel in the source image.



Figure 5: (a) original text region (b) binary image

3.4.2 Binarization

Complex backgrounds pose another difficulty for OCR. GOCR can accept a grey image as input and it does the binarization to separate text from the background by global thresholding [17, 20]. Unfortunately, global thresholding is usually not possible for complicated images. Consequently, GOCR works poorly in these cases. Figure 5 shows one example where GOCR failed when given the original text region (a), while GOCR successfully recognized the binary image (b) obtained by dynamic thresholding (described in the following paragraph).

A number of binarization algorithms have been proposed. We chose Niblack's method [21], which performed well in a recent survey [22]. Niblack's algorithm calculates the threshold dynamically by gliding a rectangular window across the image.

Another effective method for binarization is based on Gaussian mixture models [23, 24]. Histograms of gray value frequencies are modeled as a mixture of Gaussians as shown in Figure 6(e). The distribution of the character intensity will correspond to one Gaussian model and the background will also correspond to some Gaussian model(s). The number of Gaussian mixtures is crucial in appropriately modeling the background. Inappropriate selection of Gaussian mixtures will result in errors in character recognition. We determine the number of Gaussian mixtures by considering if the character can be segmented from the background. In Figure 6, (a) is the original text image, (b) and (c) are segmentation results by assuming two and three Gaussian mixture models respectively. The character is confused with the background if using only two Gaussian mixtures, but can be extracted using three Gaussian mixtures. Hence we choose three as the number of Gaussian mixtures. The underlying reason for three being a good choice for the number of Gaussian mixtures could be that the character is more uniform in intensity than the background. As demonstrated in Figure 6(e), the character intensity corresponds to one Gaussian and the background intensity corresponds to two Gaussian models. The parameters

of the Gaussian mixture models can be estimated via the EM algorithm [24].

Each of the two binarization algorithms introduced in this section has advantages. The modified Niblack’s algorithm is faster, but sometimes it is too sensitive to local noise. As shown in Figure 6, the dynamic thresholding (d) didn’t work as well as the Gaussian mixture model (c) for this case. The Gaussian mixture model is good at grey value distribution modelling but the estimation of the parameters is more expensive. Therefore we used the following strategy: first we apply the modified Niblack’s algorithm and run GOCR; if no characters are recognized, we apply the Gaussian mixture model algorithm and run GOCR again.

With image enhancement, we obtained a precision 79.1% of and a recall of 70.7% (Table 2).

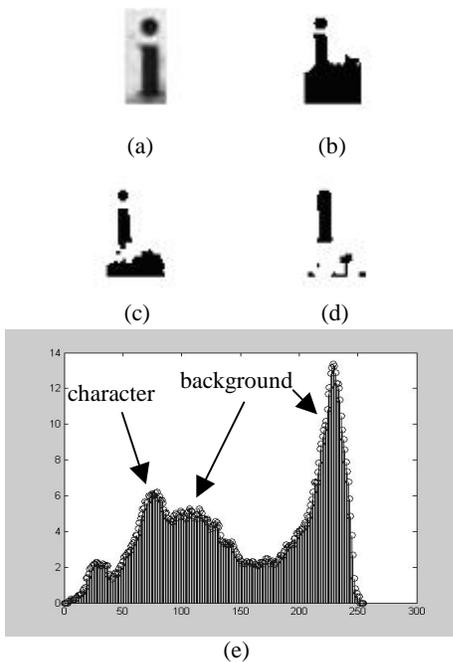


Figure 6: Binarization based on Gaussian mixture models. (a) is the original text region, (b) and (c) are thresholding results by assuming 2 and 3 Gaussian mixture models respectively, (d) is the segmentation result by dynamic thresholding, (e) is the smoothed histogram of text regions.

3.5 Modification based on grids and string match

Even with image enhancement, we might still fail to extract labels from some panels. Part of the reason is that our current binarizing algorithm is not robust enough. For example, the current binarization process couldn’t recognize the character in Figure 7 correctly.

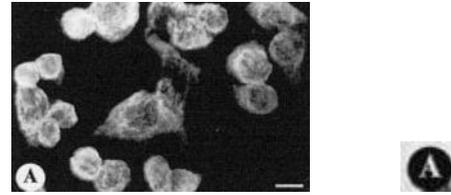


Figure 7: A case where current algorithm failed. (a) original panel, (b) detected text region.

To succeed on such cases, we can turn to the “context” information, i.e., labels extracted from other panels which are in the same figure as the failed one. If labels of these “sibling” panels are extracted correctly, we can use the context to guess what label the panel holds.

To find out the missing/incorrect label, we must figure out all the possible labels and the pattern of how these labels are assigned to the panels. We can get an idea of the range of labels from caption analysis. Since the accuracy of caption parsing is now higher than OCR, the list of labels generated from caption parsing is considered to be reliable here.

In general the arrangement of labels might be complex: labels may appear outside panels, or several panels may share one label. However, in the majority of cases, panels are grouped into grids, each panel has its own label, and labels are assigned to panels either in column-major or row-major order. The six panels shown in Figure 3 are typical of this case. For this case, we analyze the locations of the panels in the figure and reconstruct this grid, i.e., the number of total columns and rows, and also determine the row and column position of each panel.

Given the list of all panel labels, extracted from caption analysis, the grid, and the distribution of panels, we compute the label sequence assigned to panels in column-major order and row-major order, resulting in two strings SC and SR. Taking Figure 3 as an example, suppose that panel C’s label is mis-recognized as “G”, and that no label is found for panel E. In this case the string SC will be “ADB GF” and SR will be “ABGD F”.

Then we computed the similarities between the string of labels S resulting from caption analysis, and the strings SC and SR resulting from OCR and grid analysis. For instance, if caption analysis produces the string “ABCDEF”, we would compare this string to “ADB GF” and “ABGDF”. Here we used Needleman-Wunsch edit distance (using substitution costs reflecting likely OCR errors, and implemented with a package described elsewhere [25]) to compute the similarity between two strings. The edit-distance alignment for the string with the smaller distance to the OCR result is then used to correct the OCR result. For Figure 3, using our strategy, we can infer that the labels should be ABCDEF.

Table 2 shows the contribution of the modification process. Note that this procedure also produces a mapping between labels extracted from images of panels and those generated by caption interpretation.

No. of panels	No. correctly detected text region	OCR directly on panels			OCR on intensity-normalized text regions			OCR on enhanced text regions			OCR on enhanced text regions, after string-match corrections		
		No.	Prec.	Recall	No.	Prec.	Recall	No.	Prec.	Recall	No.	Prec.	Recall
427	380	15	3.9%	3.5%	271	71.3%	63.5%	302	79.1%	70.7%	316	83.2%	74.0%

Table 2: Number of panels for which the correct labels were extracted, using various algorithms.

4. CONCLUSION

Mining biological literature is crucial to organize and summarize scientific results. Most existing IE systems for the biological domain are limited to extracting information from text while figure and caption pairs in scientific publications are extremely dense in information. We have set as our long-term goal building an accurate automated toolset, SLIF, to extract information about protein subcellular localization from the text and images found in online journals.

In this paper, we gave a review of SLIF and emphasized a newly developed module. Building on our previously developed tools to find fluorescence microscope images depicting protein subcellular location patterns, we introduced caption processing to extract image pointers and entity names from the text. Image processing and OCR techniques were used to extract panel labels. Finally image pointer to panel label alignment were done to integrate the results. Our current system can generate assertions such as "Figure N depicts a localization of type L for protein P in cell type C". We believe that SLIF demonstrates the feasibility of performing IE from both text and images in biological sources.

5. REFERENCES

- [1] Blaschke, C., Andrade, M. A., Ouzounis, C., and Valencia, A.: Automatic extraction of biological information from scientific text: Protein-protein interactions. In *Proceedings of the 1999 International Conference on Intelligent Systems for Molecular Biology (ISMB-99)*. 1999, pp. 60–67.
- [2] Sekimizu, T., Park, H., and Tsujii, J. Identifying the interaction between genes and gene products based on frequently seen verbs in Medline abstracts. In *Genome Informatics*, pp. 62–71. Universal Academy Press, Inc, 1998.
- [3] Pustejovsky, J., Castaño, J., Zhang, J., Kotecki, M., and Cochran, B.: Robust relational parsing over biomedical literature: Extracting inhibit relations. In *Proceedings of 2002 the Pacific Symposium on Biocomputing (PSB-2002)*. 2002, pp. 362–373.
- [4] Thomas, J., Milward, D., Ouzounis, C., Pulman, S., and Carroll, M.: Automatic extraction of protein interactions from scientific abstracts. In *Proceedings of 2000 the Pacific Symposium on Biocomputing (PSB-2000)*. 2000, pp. 538–549.
- [5] Stephens, M., Palakal, M., Mukhopadhyay, S., Raje, R., and Mostafa, J.: Detecting gene relations from medline abstracts. In *Pacific Symposium on Biocomputing*. 2001, pp. 483–496.
- [6] Humphreys, K., Demetriou, G., and Gaizauskas, R.: Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures. In *Proceedings of 2000 the Pacific Symposium on Biocomputing (PSB-2000)*. 2000, pp. 502–513.
- [7] Fukuda, K., Tsunoda, T., Tamura, A., and Takagi, T.: Toward information extraction: Identifying protein names from biological papers. In *Proceedings of 1998 the Pacific Symposium on Biocomputing (PSB-1998)* 1998, pp. 707–718.
- [8] Rindfleisch, T., Tanabe, L., Weinstein, J. N., and Hunter, L.: Edgar: Extraction of drugs, genes and relations from the biomedical literature. In *Proceedings of 2000 the Pacific Symposium on Biocomputing (PSB-2000)*. 2000, pp. 514–525.
- [9] Bunesco, R., Ge, R., Mooney, R. J., Marcotte, E., and Ramani, A. K. Extracting gene and protein names from biomedical abstracts. Unpublished Technical Note, 2002, <http://www.cs.utexas.edu/users/ml/publication/ie.html>.
- [10] Craven, M. and Kumlien, J.: Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB-99)*. AAAI Press, 1999, pp. 77–86.
- [11] Stapley, B., Kelley, L., and Sternberg, M.: Predicting the sub-cellular location of proteins from text using support vector machines. In *Proceedings of the 2002 Pacific Symposium on Biocomputing*. 2002, pp. 374–385.
- [12] Murphy, R.F., Velliste, M., and Porreca, G.: Robust Classification of Subcellular Location Patterns in Fluorescence Microscope Images. In *Proceedings of 2002 IEEE Intl Workshop Neural Networks Signal Processing (NNSP 12)*, pp. 67-76.
- [13] Murphy, R. F., Velliste, M., Yao, J., and Porreca, G.: Searching Online Journals for Fluorescence Microscope Images Depicting Protein Subcellular Location Patterns. In *Proceedings of IEEE Int Symp Bio-Informat Biomed Eng (BIBE 2001) 2*, pp. 119-128.
- [14] Cohen, W., Wang, R., Murphy, R.F.: Understanding Captions in Biomedical Publications, in KDD-2003 (to appear).
- [15] Wu, V., Manmatha, R., Riseman, E. M.: TextFinder: An Automatic System to Detect and Recognize Text In Images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, November 1999, 21(11), pp. 1224-1229.

- [16]Mori, S., Suen, C. Y., and Yamamoto, K.. Historical review of OCR research and Development. *Proceedings of The IEEE*, 80(7), July 1992, pp. 1029-1058.
- [17]GOCR0.37, <http://jocr.sourceforge.net/>.
- [18]Shapiro, L., and Stockman, G.. *Computer Vision*. Prentice Hall, 2001, p156-176.
- [19]Bicubic Interpolation for Image Scaling, <http://astronomy.swin.edu.au/~pbourke/colour/bicubic/>
- [20]Otsu, N.. A thresholding selection method from gray-level histogram. *IEEE Transactions on Systems, Man, and Cybernetics*, 1979,9(1): pp. 62-66.
- [21]Niblack, W.. *An Introduction to Digital Image Processing*, pp.115–116. Englewood Cliffs, N.J.: Prentice Hall, 1986.
- [22]Wolf, C., Jolion, J., and Chassaing, F.. Text Localization, Enhancement and Binarization in Multimedia Documents. In *Proceedings of the International Conference on Pattern Recognition (ICPR) 2002*, volume 4, pp. 1037-1040.
- [23]Yang, J., Gao, J., Zang, Y., Chen, X., and Waibel, A.. An Automatic Sign Recognition and Translation System. In *Proceedings of the Workshop on Perceptive User Interfaces (PUI), 2001*.
- [24]Mitchell, T.. *Machine learning*. McGraw Hill, 1997. pp. 191-196.
- [25]Cohen, W., Ravikumar, P., and Fienberg, S.E.. A Comparison of String Distance Metrics for Name-Matching Tasks. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web(IIWeb-03)*, To appear.