Abstract

Motivation: Figures and captions convey essential information in biomedical documents. As such, there is a growing interest in mining published biomedical figures and in utilizing their respective captions as a source of knowledge. Notably, an essential step underlying such mining is the extraction of figures and captions from publications. While several PDF parsing tools that extract information from such documents are publicly available, they attempt to identify images by analyzing the PDF encoding and structure and the complex graphical objects embedded within. As such, they often incorrectly identify figures and captions in scientific publications, whose structure is often non-trivial. The extraction of figures, captions and figure-caption pairs from biomedical publications is thus neither well-studied nor yet well-addressed.

Results: We introduce a new and effective system for figure and caption extraction, PDFigCapX. Unlike existing methods, we first separate between text and graphical contents, and then utilize layout information to effectively detect and extract figures and captions. We generate files containing the figures and their associated captions and provide those as output to the end-user.

We test our system both over a public dataset of computer science documents previously used by others, and over two newly collected sets of publications focusing on the biomedical domain. Our experiments and results comparing PDFigCapX to other state-of-the-art systems show a significant improvement in performance, and demonstrate the effectiveness and robustness of our approach.

Availability and implementation: Our system is publicly available for use at: https://www.eecis.udel.edu/~compbio/PDFigCapX. The two new datasets are available at: https://www.eecis.udel.edu/~compbio/PDFigCapX/Downloads

Contact: pengyuan@udel.edu or shatkay@udel.edu

1 Introduction

Figures and captions convey essential information in biomedical documents. As such, there is a growing interest in mining figures and captions appearing within biomedical publications (Ahmed et al., 2016; Kuhn et al., 2014; Ma et al., 2015; Murphy et al., 2001; Shatkay et al., 2006). For example, the Mouse Genome Informatics at the Jackson Lab curates images extracted from the literature demonstrating mouse phenotypes or gene expression in the mouse (Blake et al., 2011; Smith et al., 2018). Several platforms, such as BioText, the Yale Image Finder, askHermes, Open-i and the GXD database aim to enable users to search for relevant biomedical figures and captions (Demner-Fushman et al., 2012; Finger et al., 2017; Hearst et al., 2007; Xu et al., 2008; Yu et al., 2010). However, the first step toward this goal, namely, extracting figure and caption pairs from biomedical documents is neither well-studied nor yet well-addressed. We thus introduce an effective new method to extract figures and captions from biomedical publications.

Such extraction is not a simple task due to the complex and diverse layout of scientific publications and the variations in figure structure, texture, and contents. As biomedical figures often comprise multiple image panels, identifying compound figures and their constituent panels has itself been a topic of much research (Chhatkuli et al., 2013; Li et al., 2018; Santosh et al., 2015). These
lines of research assume that the figures are already extracted from the publications, and do not focus on the extraction task.

We note that much work has been dedicated to analyze historical scanned documents, in which each document page is viewed as an image. Image-analysis methods, such as region classification or connected component based approaches were employed for identifying document structure (Bhowmik et al., 2018; Mehri et al., 2017; O’Gorman, 1993; Shafait et al., 2008). However, most of the scientific literature over the past two decades is stored in Portable Document Format (PDF) and not as scanned documents. As such, effective methods that extract images from PDF files are needed, and are the focus of the work reported here.

Most current biomedical publications are stored in PDF, in which figures are encoded as raster graphics (e.g. PNG, JPEG) or as vector graphics (e.g. SVG, EPS). Quite a few generic tools are available online, such as Apache PDFBox (https://pdfbox.apache.org), PDFMiner (https://github.com/euske/pdfminer) and Xpdf (http://www.xpdfreader.com), for converting a PDF document into a structured XML/HTML format and extracting figures. However, as these tools do not distinguish figure captions from the rest of the text in the article, they do not associate figures with their respective captions. Moreover, the above tools often extract individual components within the figure, rather than the complete figure as a whole. Figure 1 shows an example of a vector graphic image extracted by employing Xpdf. The original figure (Fig. 1a), which consists of numerous bars, line fragments and dots, is broken by Xpdf into multiple small images corresponding to the individual parts, shown in Figure 1b. Thus, these tools leave much to be desired for extracting figures or figure-caption pairs.

Specific approaches aiming to extract biomedical figures and captions from PDF documents utilizing readily available tools have been proposed (e.g. Choudhury et al., 2013; Lopez et al., 2011), but neither method handles vector graphics within documents. Identifying vector graphics that form actual figures is challenging because vector graphics may represent graphical objects that are not figures, such as the border line at a page’s margin or mathematical symbols within the document. Limited methods handling such figures have been proposed before (Choudhury et al., 2015; Shao and Futrelle, 2006) for extracting figures consisting of simple geometric components, such as curves and rectangles (e.g. line graphs, bar charts etc.), but have not been generalized to more complex figures, nor translated into working tools.

More recent methods, typically based on multiple domain-specific heuristic rules, have been developed for specific research areas, such as high-energy physics (PDFPlotExtractor, Praczyk et al., 2013) and computer science (pdffigures2, Clark and Divvala, 2016). While these tools utilize clustering and classification for separating certain types of graphics, vector graphics are often incorrectly extracted due to the complex figure and document structure. Moreover, as demonstrated by our experiments, these methods do not accurately extract figures and their respective captions from documents outside the domain in which they were developed, and specifically from biomedical documents where publications vary greatly in contents and layout.

In this paper, we present a new and effective scheme for extracting figure and associated captions from biomedical documents, which also proves to work well across different domains. Unlike earlier methods, our method does not directly analyze the raw graphical objects encoded in the PDF. Rather, it separates the text contents from the graphical contents of the PDF file, and applies Connected Component Analysis (CCA) (Gonzalez and Woods, 2002; Li et al., 2018; Shatkay et al., 2006) to the graphical contents in order to detect individual figures. Figure-caption pairs are then recovered by processing layout information of the PDF as well as the text-part extracted from the PDF file.

The rest of the paper presents the details of our method, and demonstrates its effectiveness through a series of experiments. Section 2 discusses the method itself; in Section 3, we present the experiments used to assess the performance of our method, along with the results obtained by our method and by other state-of-the-art systems used for comparison; Section 4 discusses and analyzes the results, while Section 5 concludes and outlines directions for future work.

2 Methods

Our goal is to extract figures and associated captions from biomedical documents. The complete framework for our approach is summarized in Figure 2. We first parse each PDF document (Step 1), employing the publicly available Xpdf parsing tool, which separates the PDF file into text-stripped pages that contain only graphical contents, while the text contents is stored in an HTML file.

Fig. 1. An example of image extraction of a vector graphics figure by the PDF parsing tool, Xpdf. (a) The original figure taken from Cui et al. (2012), Fig. 3. (b) The image, as extracted by Xpdf, broken into its small constituent image parts.
As noted above, publications discussing different topics or originating from different journals typically have different layouts. Basic layout information is obtained from the HTML file (Step 2), including the number of columns, contents region boundaries etc. To detect potential caption headers, we scan the HTML file for text lines beginning with terms such as Fig or FIG (Step 3), and record their position within the PDF page layout. As a figure typically appears directly above or below its caption, we designate the regions above and below the caption position as a potential figure region.

Within each text-stripped page, we employ CCA to identify graphical contents that are potential figure constituents (Step 4). Potential figure regions identified in Step 3 that are actually populated by graphical contents as detected in Step 4, are unambiguously designated as figure regions (Step 5). The associated figure caption is formed by expanding the continuous text block (Step 6) next to the potential caption header position detected in Step 3. The complete figure and its associated caption are then added as a pair to an output file. The rest of this section provides detail about each of the steps.

### 2.1 PDF parsing

We initially parse the document using the publicly available tool, Xpdf, which partitions the PDF file contents into textual contents stored in an HTML file as text objects, and graphical contents stored as text-stripped image-pages containing only the graphics from the original file. Each text object stored in the HTML typically corresponds to a text line in the PDF document, along with position information (starting point coordinates, font type and size, etc.) indicating the line’s exact starting position and size within the PDF document. The position and length of each text line in the original PDF can thus be estimated using the stored information. Figure 3 demonstrates the result of the parsing process applied to a single PDF page, where Figure 3a shows the original PDF page and Figure 3b shows the text-stripped page containing the graphical contents; the text objects comprising text lines and their respective positions are shown as rectangles in Figure 3c.

### 2.2 Document layout acquisition

Scientific documents usually adhere to typical layout and organization guidelines (e.g. margins are fixed and all contents appear within these margins). Obtaining layout parameters is thus important in guiding the figure and caption detection process. As text lines typically occupy most of the area in scientific publications, we use position information of text objects extracted in Step 1 to calculate the layout parameters. **Text-line height and width:** the text-line height, $h_{\text{height}}$, and width, $w_{\text{width}}$, indicate the typical height and width of text lines in body text—which are usually a function of font size and style. As text within the paper’s body typically has a characteristic font size, the text-line height and width parameters are set to the most frequently occurring value (i.e. the *mode*) of height and of width among all text objects obtained from the PDF file.

**Contents region:** this is the total region of the page in which any textual or figure contents can appear, typically located within well-defined margins. Notably, non-contents items such as the journal logo, date or side bars often appear on the margins, outside the contents region. Figure 3a shows the contents region and the respective margins for a single PDF page.

To identify the contents region we record the leftmost top point, $(x_l, y_l)$, and the rightmost bottom point, $(x_r, y_r)$, among the positions of all text lines of length $l_{\text{width}}$ appearing within all the pages of the PDF file. The contents region is represented as a bounding box, $[x_l, y_l; w_{\text{width}}, h_{\text{height}}]$, where $w_{\text{width}}$ and $h_{\text{height}}$ correspond to the width and height of the contents region, calculated as $w_{\text{width}} = |x_r - x_l|$ and $h_{\text{height}} = |y_r - y_l|$, respectively.

**Number of columns:** the number of columns within a page is an essential piece of information, as it has much impact on figure location and appearance. For instance, figures in single-column documents often span the whole width of the page and as such are typically wider than figures appearing in multi-column documents. The number of columns, denoted $c_{\text{num}}$, is calculated as: $c_{\text{num}} = \text{floor}(w_{\text{width}}/l_{\text{width}})$.

### 2.3 Identification of potential caption and figure regions

To identify likely candidate captions and figures, we begin by detecting a putative bounding box for each potential caption, utilizing the text information stored in the structured HTML file. As captions typically start with the prefix Fig or FIG, we consider each text line starting with such a prefix as a potential caption. For each potential caption, $Cap_i$, we record its position as it appears in the HTML file (see Section 2.1), and initially set the width of its bounding box to be the length of its first text line.

A figure is usually placed either directly above the topmost line of its caption, below its bottom line or to the side of its caption—where the figure’s top or bottom are aligned with the top or the bottom of the caption, respectively. As such, the figure region typically lies in close proximity to the caption region. We thus use the potential caption location to identify potential figure locations.

For a single-column page, with a single potential caption in it, $Cap_i$, the potential figure region, denoted $PF_{i,c}$, is initially estimated as the whole contents region of the page, $[x_l, y_l; w_{\text{width}}, h_{\text{height}}]$, as defined in Section 2.2. If the page contains multiple potential captions, $Cap_{i_1}, \ldots, Cap_{i_m}$, the height of the potential figure region $PF_{i,c}$, associated with each caption $Cap_{i_j}$, spans from the position of the previous caption $Cap_{i_{j-1}}$ to that of $Cap_{i_j}$; the width remains that of the contents region, $w_{\text{width}}$. For the first caption in a page ($Cap_1$), the associated potential figure region starts from the top of the contents region. The figure region associated with the bottom-most caption in the page includes the area all the way to the bottom of the contents region. Figure 4a illustrates potential caption and figure regions for a single-column document.

For multi column documents, the process is similar; the only difference is in the determination of the potential figure width. If the potential caption spans across the center of the page, the figure is assumed to span the whole page width, and as such the width of the potential figure region is set to the width of the whole contents region, $w_{\text{width}}$; otherwise, the width of the potential figure region is set to the width of a single column, $l_{\text{width}}$. Figure 4b shows an example of potential caption and figure regions within a multi column document.

### 2.4 Graphical contents detection

To detect the actual figures, we first detect their constituent parts within each text-stripped PDF page, by employing CCA (Gonzalez...
Typically, graphical contents within PDF pages consist of non-white areas. As such, we employ a binary mask to separate the non-white foreground from the background, by setting an intensity threshold value \( t \). Pixels whose intensity is greater than \( t \) are set to 1 (white), i.e., foreground, while all other pixels constitute the background and are thus set to 0 (black). In our experiments the threshold \( t \) is set to 0.95. We note that disconnected small parts of the figure, such as individual data points in graphs, parts of figure legends or arrows may be detected as individual distinct components by CCA. To ensure those small objects are incorporated into the detected figure, we dilate the foreground in the masked image, thus enlarging the small objects and connecting them to the rest of the figure. We then apply Connected Component Labeling (Gonzalez and Woods, 2002) to the dilated image, thus identifying connected components that constitute the actual figure contents. We then set a bounding box around each connected component, encompassing the smallest rectangle containing all the pixels within the component.

Figure 5 illustrates the detection process: The text-stripped PDF page is shown in Figure 5a. Figure 5b shows the application of the binary mask, and Figure 5c shows its dilation. Figure 5d shows connected components detected by our method, each surrounded by a bounding box shown as a blue rectangle.

2.5 Disambiguation of figure regions

Notably, as demonstrated by Figure 5, the previous step results in relatively small figure-constituents, which may combine to form one or more individual complete figures. To unambiguously identify the actual complete individual figures within each potential figure region \( \text{PFig}_i \) (see Section 2.3), we merge all bounding boxes of connected components within each potential figure into a single figure denoted \( \text{Fig}_i \). We note that text-elements that are essential parts of some figures (e.g. figure legends, or axes labels in graphs) are initially removed and stored in the HTML file as text objects during the initial parsing step (Section 2.1). To recover the textual contents and place it back into the figure, we merge each figure region \( \text{Fig}_i \) with all the text objects located close to it, i.e. within Manhattan distance smaller than 1/2 a line-height from the figure region, or with those objects located between the figure region and its respective potential caption region.

Figure 6 provides an example of the figure disambiguation process. Figure 6b shows the connected components detected as explained in Section 2.4, surrounded by solid blue bounding boxes, while the dashed green box indicates the potential caption; the outermost thicker orange box indicates the potential figure region. Figure 6c shows the figure region identified at the end of the disambiguation step.

2.6 Caption expansion

As discussed in Section 2.3, headers of potential captions that can help locate associated figures are first detected using prefixes such as Fig or FIG. To expand the header associated with each potential caption \( \text{Cap}_i \) into a complete caption, we detect the continuous text block following the potential header. The text block is defined as a sequence of text objects, where the last object ends with a period and its width is lower than that of the bounding box surrounding \( \text{Cap}_i \) (see Section 2.3), or where the last object is followed by a vertical gap (i.e. by a gap whose height exceeds the regular gap between body text lines).

To estimate the position of a caption on a page, we first merge the bounding boxes of all text objects within the text block defined above. The complete caption to be associated with a figure \( \text{Fig}_i \) is formed by combining the text content of all text objects from top to bottom within the detected text block. The complete figure and its associated caption are then stored as part of the output files, along with their bounding boxes that indicate their respective position on the corresponding PDF page.

3 Experiments and results

We compare the performance of our system to that of two publicly available state-of-the-art systems, pdffigures2 (Clark and Divvala, 2016) and PDFPlotExtractor (Praczyk et al., 2013), performing three tasks: figure extraction, caption extraction and the combined task of figure-caption pair extraction. Three different datasets for which we have the ground-truth are used. The first has been assembled and used before by the developers of pdffigures2 (Clark and Divvala, 2016). As pdffigures2 was developed by and for computer scientists, the dataset comprises non-biomedical publications.

To test the systems on biomedical publications we assembled and annotated two additional datasets representing different biomedical domains, as further described in Section 3.1. All the documents used in our experiments are PDF documents and are all successfully processed by all three methods. The evaluation process and measures, the datasets and the experimental results are all discussed below.

3.1 Datasets and evaluation

We test the performance of our system on three datasets. The first, denoted CS-J50, was introduced by Clark and Divvala (2015) for testing their system pdffigures2, focusing on computer science publications. The dataset comprises 150 PDF publications, selected from three top computer science conferences published during the period.
2009–2014, of which 10 were published in 2009, 20 in 2014 and 30 publications in each of the years 2010–2013. Figures and captions were manually annotated using bounding boxes and labeled by Clark et al. when the dataset was created. There are 458 figures, 458 captions and respectively 458 figure-caption pairs annotated in the dataset.

The CS-150 set is limited to computer science publications, which typically differ in figure and text organization and in contents from biomedical publications. To assess performance specifically within the biomedical domain, we have built and annotated two additional larger datasets—each comprising 200 documents, both focused on biomedical publications, using two distinct data sources as further described below. Noting that PDF image and text coding varies significantly between older and newer PDF files, which can affect the performance of figure extraction algorithms. The GS200 dataset used in the CS-150 dataset is focused on computer science and machine learning; the system pdffigures2—which was specifically built for this domain—indeed achieved the highest performance when extracting figures from this dataset (98.88% precision, 95.85% recall, 97.34% F-score). Our method, which is not specifically designed for this domain, still handles figure extraction from this dataset well, with 93.50% precision, 88.00% recall and 90.67% F-score, while outperforming the pdffigures2 and PDFPlotExtractor systems on the tasks of caption and caption-figure pair extraction over the CS-150 dataset.

The next three columns in the table show results obtained by our system, compared with those obtained by the other two methods. The results obtained by our system are statistically significant (p < 0.0001, paired t-test).

Figure 7 shows examples of four PDF pages annotated with bounding boxes around figures and captions reflecting the extraction results obtained by our system, compared with those obtained by the other two methods. The results obtained by our system are shown at the top (Fig. 7a–d), while results obtained by pdffigures2 (Fig. 7a1–d1) and PDFPlotExtractor (Fig. 7a2–d2) are shown in the middle and at the bottom.

In the following section we further analyze and discuss these results.

### 4 Discussion

As indicated above, the system pdffigures2 was developed for handling computer science papers, and as such shows excellent performance, for figure extraction from the CS-150 dataset, developed by its authors (98.88% precision, 95.85% recall, 97.34% F-score). Notably, the difference between the figure-extraction results

---

**Fig. 6.** An example of the figure disambiguation process. (a) The original PDF page (Christoffels et al., 2009, p.1). (b) Blue bounding boxes surrounding connected components, while the potential caption region is surrounded by a green dashed box; the entire potential figure region is shown encompassed by a thick orange box. (c) The figure region, shown within a red box, as identified at the end of the disambiguation process.
Table 1. Results obtained by our system, PDFigCapX, and by other state-of-the-art systems on the three tasks: figure extraction (A), caption extraction (B) and figure-caption pair extraction (C)

| System          | CS-150 Precision (%) | CS-150 Recall (%) | CS-150 F-score (%) | GXD-200 Precision (%) | GXD-200 Recall (%) | GXD-200 F-score (%) | PMC-200 Precision (%) | PMC-200 Recall (%) | PMC-200 F-score (%) |
|-----------------|-----------------------|-------------------|--------------------|------------------------|-------------------|---------------------|------------------------|-------------------|---------------------|
| pdffigures2     | 98.88                 | 95.85             | 97.34              | 91.85                  | 65.91             | 76.75               | 89.32                  | 41.47             | 56.64               |
| PDFPlotExtractor| 64.52                 | 85.37             | 73.49              | 56.85                  | 69.89             | 62.70               | 38.89                  | 64.49             | 48.52               |
| PDFigCapX       | 93.50                 | 88.00             | 90.67              | 89.86                  | 93.03             | 91.42               | 87.67                  | 90.79             | 89.20               |
| (B) Caption extraction |                  |                   |                    |                        |                   |                     |                        |                   |                     |
| pdffigures2     | 88.51                 | 85.81             | 87.14              | 90.01                  | 65.24             | 75.65               | 80.90                  | 38.18             | 51.88               |
| PDFPlotExtractor| 64.76                 | 79.47             | 71.36              | 88.69                  | 84.64             | 87.55               | 50.26                  | 56.98             | 53.41               |
| PDFigCapX       | 94.93                 | 87.55             | 91.09              | 88.74                  | 91.14             | 89.92               | 88.98                  | 81.40             | 85.02               |
| (C) Figure and caption pair extraction |                  |                   |                    |                        |                   |                     |                        |                   |                     |
| pdffigures2     | 87.39                 | 84.71             | 86.03              | 84.45                  | 60.60             | 70.56               | 75.98                  | 35.85             | 48.71               |
| PDFPlotExtractor| 60.01                 | 73.79             | 66.19              | 60.47                  | 57.30             | 58.84               | 34.70                  | 39.34             | 36.87               |
| PDFigCapX       | 90.07                 | 84.71             | 87.31              | 82.52                  | 84.75             | 83.62               | 83.26                  | 76.16             | 79.55               |

Note: The method is indicated in the leftmost column. The next three columns show the precision, recall and F-score obtained over the CS-150 dataset (458 figures, captions and pairs); the fifth to seventh columns show the same for the GXD-200 dataset (1335 figures, 1298 captions and 1298 pairs); the three rightmost columns show the results obtained on the PMC-200 dataset (1042 figures, 1032 captions and 1032 pairs). The highest values attained are shown in boldface.

obtained by pdffigures2 and that obtained by our system is not statistically significant ($P>0.05$, paired $t$-test). Our method, which is not designed for this domain, still handles this dataset well, with 93.50% precision, 88.00% recall and 90.67% $F$-score on the figure-extraction task—outperforming the PDFPlotExtractor system ($P<0.0001$). As shown in Table 1B and C, our system outperforms both of the other systems on caption extraction and caption-figure pair extraction over the CS-150 dataset.

When applied to the GXD-200 dataset (fifth to seventh columns), our method attains a significantly higher recall (93.03%) than the two other systems (65.91, 69.89%) for figure extraction, where the difference is highly statically significant ($P<0.0001$). We note that while the precision attained by our method over this dataset is slightly lower than that of pdffigures2, our recall and $F$-score are much higher (by 15%).

All three systems show the lowest performance on the more general PMC-200 dataset; this degradation is much more pronounced in the pdffigures2 and the PDFPlotExtractor systems. Notably, the PMC-200 dataset contains articles published during the period 1990–2008, where the PDF image and text coding varies significantly between older and newer PDF files. As both pdffigures2 and PDFPlotExtractor rely on direct analysis of encoded raw graphical objects, the performance of these systems is significantly impacted by the coding variations. In contrast, our system attains a significantly higher recall (90.79%) than that of the other two systems (41.47, 64.49%, $P<0.0001$) on the same dataset, with only a slight loss (<2%) in precision compared to pdffigures2.

These results demonstrate that our method provides an effective and robust means for figure extraction from PDF; it is particularly suitable in the realm of biomedical document curation, where relevant articles may span a wide range of publication years (thus varying in PDF encodings), and where recall is often viewed as more important than precision (Fang et al., 2012; Müller et al., 2004).

As seen in Table 1B, our method attains the highest precision, recall and $F$-score for caption extraction over both the CS-150 and the PMC-200 datasets. The difference in recall is particularly notable. Over the GXD-200 dataset (fifth to seventh columns), the precision of our method (88.74%) is slightly lower than that of pdffigures2 (90.01%), but our recall and $F$-score are again significantly higher than those obtained by the other two systems ($P<0.0001$).

In terms of caption extraction, the deterioration in performance of all three systems over the PMC-200 dataset is more pronounced than it is for figure extraction (Table 1A, rightmost columns). Among the PMC-200 articles, 34 were older documents published during the period 1990–2008, and 10 of those are scanned PDF documents, rather than originally produced in PDF format. Such scanning, which involves optical character recognition of the text, often leads to errors that make captions more difficult to detect. Thus, for caption extraction, the pdffigures2 tool shows a recall level of 38.18%, while PDFPlotExtractor’s recall is at 56.98%. In contrast, our method shows a significantly higher recall in the face of these hurdles, namely, 81.40% on the same dataset ($P<0.0001$). Our system thus proves to be much more resilient and reliable in identifying captions in a broad range of PDF files compared to currently available systems.

Last, Table 1C shows the results for the combined task of figure-caption pair extraction. Notably, all three methods show lower performance on this task as it requires both the figure and its respective caption to be correctly extracted. The lower performance is more pronounced in the results obtained on the GXD-200 and PMC-200 datasets as the figure-caption organization in biomedical publications is more complex than that in computer science publications. Over the CS-150 and PMC-200 datasets our method again attains both the highest precision and the highest recall. Notably, our recall and precision on the PMC-200 dataset are significantly higher than those of the other state-of-the-art methods (83.26% compared with 75.98% in precision; 76.16% compared with 39.34% in recall; 79.55% compared with 48.71% in $F$-score; $P<0.0001$). Over the GXD-200 dataset (fifth to seventh columns), the precision of our method (82.52%) is again slightly lower than that of pdffigures2 (84.45%), while retaining a significantly higher recall (84.75% compared with 60.6%) and a much higher $F$-score (83.65% compared with 70.36%), where the difference is highly statistically significant ($P<0.0001$). These results again demonstrate and validate our method as an effective and robust means for extracting figure-caption pairs.

Figure 7a–d shows several examples of figures and captions extracted by PDFigCapX. Our system correctly extracts the figure and its associated caption both in the simpler cases where the individual figure appears directly above the caption within a single- or a double-column page (Fig. 7a and b), and in more complex cases, where multiple figures and captions appear on the same page (Fig. 7c), or when the caption is placed adjacent to—but not directly above/below—the Figure 7d.
In contrast, Figure 7a1–d1 and a2–d2 demonstrates the (incorrect) extraction performed over the same pages by the other two systems, pdffigures2 and PDFPlotExtractor. As mentioned earlier, both systems directly handle the graphical objects encoded in the PDF in order to extract figures, and as such misidentify some of the graphics embedded within a complex document structure—even when the layout appears simple (e.g. Fig. 7a1, a2, b1 and b2). For instance, pdffigures2 may not find nor extract any figures or captions on a page (Fig. 7a1 and c1), while misidentifying caption regions (Fig. 7b1) or figure regions (Fig. 7d1). On the other hand, PDFPlotExtractor over-estimates certain figure boundaries (Fig. 7a2–d2), and misidentifies figure and caption regions (Fig. 7b2).

We note that while our method correctly extracts most figures and captions, there are still a few cases in which the extraction is inaccurate. For the figure-extraction task, particularly challenging are cases in which small graphical contents (e.g. mathematical symbols, formulae, reference brackets and border lines) appear in close proximity to a figure, making it hard to determine whether the figure's boundaries include/exclude the adjacent graphical contents. Figure 8a illustrates such a case, in which our method mis-assigns the whole page as the figure region. Here, PDFigCapX erroneously merges the correct figure (top left) with additional graphical contents (bottom right), although the latter graphics is not part of a figure but merely forms background for the table at the bottom of the page. While the overestimated figure is indeed inaccurate, the extracted region still contains the whole figure without omitting essential graphics parts, thus still useful for bio-curating.

Figure 8b provides an example of a challenging caption-extraction case, in which the caption region is underestimated by our method, appearing as though part of the caption has been missed. The caption bounding box as marked in the figure was estimated based on position information of caption text objects extracted during the PDF parsing step (see Section 2.1). As the formatting information was misidentified by Xpdf, the caption region was incorrectly calculated. However, in this case our method still actually correctly harvests the complete caption text, stored in the HTML file. As our evaluation measures are based on assessing the bounding boxes of detected objects compared to the ground truth boundaries, the evaluation measure penalizes the mis-calculated bounding box even though the caption text itself is correctly extracted. Our system thus actually performs better in terms of identified captions and correct figure-caption pairs than is reflected by the evaluation metrics. A more adequate metric should take into account the actual recovery of contents rather than bounding box coordinates; we plan to develop such a metric as part of our future work.

5 Conclusion

We presented a new and effective method and system, PDFigCapX, for extracting figures, captions and figure-caption pairs from biomedical documents. Earlier methods typically detect figures by directly searching the contents encoded in the PDF file and handling the raw graphical objects embedded in it. This strategy often leads to incorrect extraction when the figure and the document structures are complex, which is a common phenomenon within biomedical publications. In contrast, our method first completely separates the text contents from the graphical contents of the PDF file, and aims to recover figures and captions utilizing layout information. It applies CCA to the graphical contents in order to detect figures, and separately searches the text portion for capturing the contents encoded in the PDF file. As our evaluation measures are directly based on assessing the bounding boxes of detected objects compared to the ground truth boundaries, the evaluation measure penalizes the mis-calculated bounding box even though the caption text itself is correctly extracted. However, in this case our method still actually correctly harvests the complete caption text, stored in the HTML file. As our evaluation measures are based on assessing the bounding boxes of detected objects compared to the ground truth boundaries, the evaluation measure penalizes the mis-calculated bounding box even though the caption text itself is correctly extracted. Our system thus actually performs better in terms of identified captions and correct figure-caption pairs than is reflected by the evaluation metrics. A more adequate metric should take into account the actual recovery of contents rather than bounding box coordinates; we plan to develop such a metric as part of our future work.
science publications. The two new annotated sets, one of which focuses on relatively narrow set of topics and years, while the other covers a broader range, are likely to support further development of PDF parsing tools pertaining to biomedical text mining.

Our extensive experiments and results demonstrate that the new system is highly effective in terms of precision, recall and F-score; specifically, it significantly improves upon existing systems in terms of recall and F-score without much loss of precision (if any). Moreover, PDFigCapX retains its good performance over documents varying broadly in topic, style, publication year and overall organization. As such, it is ready to be applied in practice.

As part of future work, we are considering a new evaluation metric that will account for the actual recovered contents of both figures and captions, rather than merely the correct identification of bounding box positions. We shall also integrate PDFigCapX with our compound-figure detection and segmentation tool, FigSplit (Li et al., 2018), providing an end-to-end system for extracting image and text contents from biomedical PDF files.

Funding

This work was partially supported by the National Institutes of Health/ National Library of Medicine awards [R56LM011354, R01LM012527].

Conflict of Interest: none declared.

References

Ahmed, Z. et al. (2016) Mining biomedical images towards valuable information retrieval in biomedical and life sciences. Database, 2016, baw118.

Bando, T. et al. (2013) Dynamic expression pattern of leucine-rich repeat neuronal protein 4 in the mouse dorsal root ganglia during development. Neurosci. Lett., 548, 73–78.

Bhowmik, S. et al. (2018) Text and non-text separation in offline document images: a survey. IJDAR, 21, 1–20.

Blake, J. A. et al. (2011) The Mouse Genome Database (MGD): premier model organism resource for mammalian genomics and genetics. Nucleic Acids Res., 39, D842–D848.

Chihat, A. et al. (2013) Separating compound figures in journal articles to allow for subfigure classification. In: Proceedings of SPIE Medical Imaging 2013, Vol. 8674, Florida, USA.

Choudhury, S. R. et al. (2013) Figure metadata extraction from digital documents. In: Proceedings of IEEE ICEDAR, pp. 135–139.

Choudhury, S. R. et al. (2015) Automatic extraction of figures from scholarly documents. In: Proceedings of ACM DocEng, pp. 47–50.

Christoffels, V. M. et al. (2009) Tbx18 and the fate of epicardial progenitors. Nature, 458, E8.

Clark, C. and Divvala, S. (2015) Looking beyond text: extracting figures, tables and captions from computer science papers. In: Proceedings of the AAAI Workshop on Scholarly Big Data. pp. 1–8.

Clark, C. and Divvala, S. (2016) PDFigures 2.0: mining figures from research papers. In: Proceedings of IEEE/ACM JCDL, pp. 143–152.

Cui, Y. et al. (2012) Role of corin in trophoblast invasion and uterine spiral artery remodelling in pregnancy. Nature, 484, 246.

De Herrera, A. G. S. et al. (2013) Overview of the ImageCLEF 2013 medical tasks. In: CLEF Working Notes, Valencia, Spain.

De Herrera, A. G. S. et al. (2015) Overview of the ImageCLEF 2015 medical classification task. In: CLEF Working Notes, Toulouse, France.

De Herrera, A. G. S. et al. (2016) Overview of the medical tasks in ImageCLEF 2016. In: CLEF Working Notes, Evora, Portugal.

Demner-Fushman, D. et al. (2012) Design and development of a multimodal biomedical information retrieval system. JCS, 6, 168–177.

Fang, R. et al. (2012) Automatic categorization of diverse experimental information in the bioscience literature. BMC Bioinformatics, 13, 16.

Finger, J. H. et al. (2017) The mouse gene expression database (GXD): 2017 update. Nucleic Acids Res., 45, D730–D736.

Gonzalez, R. C. and Woods, R. E. (2002) Digital Image Processing. Prentice Hall, NJ, USA.

Hearst, M. A. et al. (2007) BioText Search Engine: beyond abstract search. Bioinformatics, 23, 2196–2197.

Jacobs, F. M. et al. (2009) Pim3 potentiates Nurrl1 in dopamine neuron terminal differentiation through release of SMRT-mediated repression. Development, 136, 531–540.

Kuhn, T. et al. (2014) Mining images in biomedical publications: detection and analysis of gel diagrams. J. Biomed. Semantics, 5, 10.

Li, P. et al. (2018) Compound image segmentation of published biomedical figures. Bioinformatics, 34, 1192–2299.

Lopez, L. D. et al. (2011) An automatic system for extracting figures and captions in biomedical PDF documents. In: Proceedings of IEEE BIBM, pp. 578–581.

MaK. et al. (2015) Utilizing image-based features in biomedical document classification. In: Proceedings of IEEE ICIP, pp. 4451–4455.

Mehri, M. et al. (2017) Texture feature benchmarking and evaluation for historical document image analysis. IJDAR, 20, 1–35.

Müller, H. M. et al. (2004) Texpresso: an ontology-based information retrieval and extraction system for biomedical literature. PLoS Biol., 2, e309.

Murphy, R. F. et al. (2001) Searching online journals for fluorescence microscope images depicting protein subcellular location patterns. In: Proceedings of IEEE BIBE, pp. 119–128.

Nakamura, T. et al. (2015) Mesoporous silica nanoparticles for 19F magnetic resonance imaging, fluorescence imaging, and drug delivery. Chem., Sci., 6, 1986–1990.

O’Gorman, L. (1993) The document spectrum for page layout analysis. IEEE Trans. Pattern Anal. Mach. Intell., 15, 1162–1173.

Pananghat, A. N. et al. (2016) IL-8 alterations in HIV-1 infected children with disease progression. Medicine, 95, e3734.

Praczyk, P. A. et al. (2013) Automatic extraction of figures from scientific publications in high-energy physics. Inform., Technol. Libr., 32, 25.

Santosh, K. C. et al. (2015) Stiched multpanel biomedical figure separation. In: Proceedings of IEEE CBMS, pp. 54–59.

Seiwert, D. et al. (2017) The non-bilayer lipid MGDG stabilizes the major light-harvesting complex (LHCII) against unfolding. Sci. Rep., 7, 5158.

Shafait, F. et al. (2008) Performance evaluation and benchmarking of six-page segmentation algorithms. IEEE Trans. Pattern Anal. Mach. Intell., 30, 941–954.

Shao, M. and Furetelle, R. P. (2006) Recognition and classification of figures in PDF documents. In: Proceedings of the International Workshop on Graphics Recognition, pp. 231–242.

Shatkay, H. et al. (2006) Integrating image data into biomedical text categorization. Bioinformatics, 22, e446–e453.

Smith, C. L. et al. (2018) Mouse Genome Database (MGD)-2018: knowledgebase for the laboratory mouse. Nucleic Acids Res., 46, D836–D842.

Sumida, H. et al. (2010) LPA4 regulates blood and lymphatic vessel formation during mouse embryogenesis. Blood, 116, 5060–5070.

Xu, S. et al. (2008) Yale Image Finder (YIF): a new search engine for retrieving biomedical images. Bioinformatics, 24, 1968–1970.

Yu, H. et al. (2010) Automatic figure ranking and user interfacing for intelligent figure search. PLoS One, 5, e12983.