

Exploring Image Analysis in R: Applications and Advancements

by Tim Brauckhoff, Julius Rublack, and Stefan Rödiger (corresponding author)

Abstract This review offers an overview of image processing packages in R, covering applications such as multiplex imaging, cell tracking, and general-purpose tools. We found 38 R packages for image analysis, with *adimpro* and *EBImage* being the oldest, published in 2006, and *biopixR* among the newest, released in 2024. Of these packages, over 90 % are still active, with two-third receiving updates within the last 1.5 years. The pivotal role of bioimage informatics in life sciences is emphasized in this review, along with the ongoing advancements of R's functionality through novel code releases. It focuses on complete analysis pipelines for extracting valuable information from biological images and includes real-world examples. Demonstrating how researchers can use R to tackle new scientific challenges in image analysis, the review provides a comprehensive understanding of R's utility in this field.

1 Introduction

Advancements in microscopy and computational tools have become pivotal to biological research, facilitating detailed investigation of cellular and molecular processes previously inaccessible. Consequently, imaging methodologies, staining protocols, and fluorescent labeling — particularly those employing genetically encoded fluorescent proteins and immunofluorescence — have resulted in a substantial increase of the capacity to examine cellular structures, dynamics, and functions (Swedlow, Goldberg, and Eliceiri 2009; Peng et al. 2012; Chessel 2017; Moen et al. 2019; J. Schneider et al. 2019).

As with any significant advance in today's world, software is required to facilitate the acquisition, analysis, management, and visualization of image data resulting from these techniques. The current techniques have allowed the capture of biological phenomena with an unparalleled level of complexity and resolution (Eliceiri et al. 2012). As a result, an ever-growing amount of image data is being generated (Peng et al. 2012). Alongside the three spatial dimensions, images now encompass additional dimensions like time and color channels. Biomedical images exhibit this high level of complexity, as evidenced by the analysis of dense cell turfs where cells may partially overlap (Peng 2008; Swedlow, Goldberg, and Eliceiri 2009). The increase in complexity demands computational approaches. Nevertheless, the challenge posed is not solely due to complexity. As imaging technology advances, the volume of image data generated from experiments also sees a steep rise (Peng 2008; Caicedo et al. 2017).

The need for quantitative information from images to understand and develop new biological concepts has led to the emergence of bioimage informatics as a specialized field of study (Eliceiri et al. 2012; Murphy 2014). Bioimage informatics is primarily concerned with the extraction of quantitative information from images to interpret biological concepts or develop new ones (Chessel 2017; Moen et al. 2019; J. Schneider et al. 2019). Bioimage informatics focuses on the automation of objective and reproducible image data analysis, while concurrently developing tools for the visualization, storage, processing, and analysis of such data (Swedlow and Eliceiri 2009; Peng et al. 2012). Crucial advancements range from cell phenotype screening, drug discovery and cancer diagnosis to gene function, metabolic pathways, and protein expression patterns. The basic operations in bioimage informatics are feature extraction and selection, segmentation, registration, clustering, classification, annotation and visualization (Peng 2008).

Due to recent advancements, the utilization of microscopy in biology has evolved into a quantitative approach, as opposed to solely a visual one. Thus, various essential open-source platforms, applications, and languages have emerged, which have now become well-established within the life science community (Paul-Gilloteaux 2023). Python, R, and MATLAB are among the most favored programming languages in bioinformatics (Giorgi, Ceraolo, and Mercatelli 2022), with Python and R being extensively used in biomedicine (Roesch et al. 2023). R plays a pivotal role in the fields of statistics, bioinformatics, and data science. It is a versatile statistical software that is used in various assays for example in gene expression analyses (Rödiger, Böhm, and Schimke 2013; Rödiger, Burdukiewicz, Blagodatskikh, and Schierack 2015; Michał Burdukiewicz et al. 2022; Chilimoniuk et al. 2024). Furthermore, it is one of the top ten most prevalent programming languages across the globe, with a thriving community that has developed numerous extensions and packages for various applications (Giorgi, Ceraolo, and Mercatelli 2022). Originally developed for statistical analysis, R and its packages now offer robust capabilities for image analysis and automation (Chessel 2017; Haase et al. 2022). The growing demand for automation and data-driven analysis underscores the necessity

for flexible and integrated computational tools. R's expanding ecosystem of packages, ranging from general-purpose image processing to specialized, domain-specific workflows, facilitates the creation of customized solutions tailored to diverse research needs. The extensible framework and robust statistical capabilities support seamless integration of image analysis with downstream data interpretation, promoting reproducibility and efficiency across the entire analytical pipeline (Rödiger, Burdukiewicz, Blagodatskikh, Jahn, et al. 2015; Chessel 2017; Giorgi, Ceraolo, and Mercatelli 2022; Haase et al. 2022).

R can integrate with other programming languages through the use of packages such as *reticulate* (Ushey, Allaire, and Tang 2024) for Python, which enables users to leverage the strengths of multiple languages within their research workflows, enhancing flexibility across diverse domains. Another example for this is Bio7. Bio7 is an open-source platform designed for ecological modeling, scientific image analysis, and statistical analysis. It provides an R development environment and integration with the ImageJ application (Austenfeld and Beyschlag 2012). ImageJ is a widely-used, public-domain Java-based software suite specifically developed for biological image processing and analysis, that supports various file formats, advanced image manipulation techniques, and a vast array of plugins and scripts (C. A. Schneider, Rasband, and Eliceiri 2012).

A common difficulty in bioinformatics is the large number of file formats, some of which are proprietary. A lack of standardization means that general tools must deal with this vast array of file formats. The open source approach provides access to the code of applications, packages, and extensions, thereby facilitating modification and further development by the community. This enhances reproducibility and validation, offering flexibility and adaptability for scientific discovery. Making open source methods ideally suited to the diverse and interdisciplinary field of biological imaging research (Swedlow and Eliceiri 2009; Rödiger, Burdukiewicz, Blagodatskikh, Jahn, et al. 2015). The Open Microscopy Environment (OME) offers a standardized, open-source framework for the management, analysis, and exchange of biological imaging data, with a particular focus on the integration and preservation of rich metadata — such as experimental conditions, cell types, acquisition parameters, microscope specifications, and quantification methods (Goldberg et al. 2005). A central objective of OME is to ensure lossless storage and interoperability across diverse proprietary and non-proprietary platforms. This objective addresses the common issue of metadata loss during format conversions within image analysis pipelines. By establishing standardized formats and protocols, OME fosters compatibility between proprietary systems and enhances reproducibility. The widely adopted OME-TIFF format extends the traditional TIFF structure by embedding metadata in XML, enabling efficient storage and retrieval of large, multidimensional datasets commonly encountered in fluorescence imaging (Linkert et al. 2010; Leigh et al. 2016; Besson et al. 2019). In addition, the OME-ZARR format, developed under the Next-Generation File Format (NGFF) initiative, has been optimized for scalable, cloud-based storage of large N-dimensional arrays, with metadata stored in human-readable JSON. The system's capacity for partial data access is a notable feature, contributing to enhanced performance in distributed workflows by combining formats such as OME-TIFF, Hierarchical Data Format 5 (HDF5), and Zarr (Moore et al. 2021, 2023)¹. Increasing adoption of these formats by commercial imaging software vendors further strengthens their relevance and sustainability (Linkert et al. 2010). In the context of R-based workflows, the *RBioFormats* package provides a native interface to the OME Bio-Formats Java library. This enables the reading of proprietary file formats and associated metadata, output to OME-TIFF, and seamless integration of image acquisition with downstream analysis (Andrzej Oleś, John Lee 2023). This facilitates the establishment of flexible, standardized, and reproducible image analysis pipelines within the R ecosystem.

The heterogeneous and dynamic nature of images present a constant challenge for image analysis. Capturing precise and high-quality images that accurately represent the changing characteristics of an experiment can be difficult, even for experienced researchers (Swedlow, Goldberg, and Eliceiri 2009). Additionally, visualizing and analyzing multi-gigabyte data sets requires substantial computational power. The process of detailed analysis of image sequences, which involves identifying and tracking objects, followed by the presentation of the resulting data and the exploration of the underlying biological mechanisms, adds further complexity (Swedlow and Eliceiri 2009). To at least simplify the process of selecting the appropriate software, this review provides an overview of R packages suitable for image analysis and outlines their applications in biological laboratory settings.

2 Methods

In this study, a review of the literature was conducted over the period September 2023 to March 2024. The objective was to identify and analyze R packages that are suitable for bioimage informatics applications. The primary resources included the Comprehensive R Archive Network (CRAN)²,

¹<https://ngff.openmicroscopy.org/about/index.html>, accessed 07/13/2025

²<https://cran.r-project.org/>, accessed 04/17/2025

GitHub repositories³, rOpenSci's r-universe⁴, the Bioconductor repository⁵, OpenAlex database, PubMed, and Google Scholar. The chosen sources allowed for an extensive coverage of R package repositories while also providing access to relevant scientific literature. By combining these resources, the study aimed to provide a comprehensive overview of available tools and techniques within the domain of bioimage informatics using R.

The search strategy centered around pertinent keywords, including "bioimage," "biomedical image analysis," "imaging," "microscopy," "histology," and "pathology" and the following search strings:

- https://openalex.org/works?page=1&filter=title_and_abstract.search%3AR%20packages%20for%20image%20analysis
- https://openalex.org/works?page=1&filter=title_and_abstract.search%3Aimage%20processing%20in%20R
- https://openalex.org/works?page=1&filter=title_and_abstract.search%3Amicroscopy%20imaging%20in%20R%20packages
- <https://pubmed.ncbi.nlm.nih.gov/?term=biomedical+image+analysis&filter=dates.1963%2F1%2F1-2025%2F3%2F26&filter=pubt.review&filter=other.excludepreprints>
- https://scholar.google.de/scholar?hl=de&as_sdt=0%2C5&q=image+analysis+in+R&btnG=
- https://scholar.google.de/scholar?hl=de&as_sdt=0%2C5&q=histology+in+R&btnG=
- https://scholar.google.de/scholar?hl=de&as_sdt=0%2C5&q=pathology+in+R&btnG=

The identified packages were then subjected to an analysis to understand their usage, dependencies on other libraries, repository hosting platforms, and licensing terms.

The examples provided, along with this review, were created using RMarkdown. All computations were performed using the R programming language, version 4.3.3, on a 64-bit x86_64-pc-linux-gnu platform with the Ubuntu 22.04.3 LTS operating system. We utilized the RStudio Integrated Development Environment (IDE, 2023.09.0+463 "Desert Sunflower", Ubuntu Jammy).

This review will examine a variety of R packages designed for image analysis, including both general-purpose tools and those crafted for specific applications. This overview aims to demonstrate the diverse capabilities and adaptability of these tools within and beyond biological research contexts. Given the significant interest in the localization of microplastics in cells and the environment, our examples will primarily focus on the analysis of microbead particles made of polymethylmethacrylate (PMMA), which measure approximately 12 µm and fall within the microplastic size range (Geithe et al. 2024). As microbeads are round, spherical objects in images, they visually resemble other commonly imaged objects such as seeds and cells.

3 Dividing to conquer - advanced segmentation strategies

Image segmentation is a crucial preliminary step in image analysis and interpretation. It involves dividing an image into distinct regions by assigning a label to each pixel. The primary objective is to delineate regions pertinent to the specific task (Peng 2008; Ghosh et al. 2019; Jürgen Niedballa et al. 2022b). This process frequently employs features such as pixel intensity, gradient magnitude, or texture measures. Based on these features, segmentation techniques can be classified into three categories: region-based, edge-based, or classification-based. Classification-based methods assign class labels to pixels based on their feature values, whereas region-based and edge-based techniques focus on within-region homogeneity and between-region contrast. One straightforward method of segmentation is thresholding, which involves comparing pixel values against one or more intensity thresholds. This process typically separates the image into foreground and background regions (Sonka and Fitzpatrick 2000; Jähne 2002).

Another image segmentation method was proposed by Ren and Malik (2003). This approach integrates a preprocessing step that segments the image into superpixels, feature extraction based on Gestalt cues, evaluation of the extracted features, and the training of a linear classifier. Superpixels are clusters of pixels that are similar with respect to properties such as color and texture, resulting in larger subregions of the image. The primary objective of this preprocessing step is to simplify the image and reduce the number of regions considered for segmentation. Previously, this involved evaluating every single pixel. The division of the image into regions larger than pixels but smaller than objects allows

³<https://github.com/>, accessed 04/17/2025

⁴<https://ropensci.r-universe.dev/builds>, accessed 04/17/2025

⁵<https://www.bioconductor.org/>, accessed 04/17/2025

for the superpixels to encompass a greater quantity of information, adhere to the boundaries of natural image objects, reduce the presence of noise and outliers, and enhance the speed of the subsequent segmentation process. In summary, this method can be described as segmentation based on low-level pixel grouping (Ren and Malik 2003; Hossain and Chen 2019; Mouselimis et al. 2023).

However, segmentation is not limited to the differentiation of the foreground and background. Pixel classification plays a critical role in a number of applications, including visual question answering, object counting, and tracking. In these applications, classification occurs not just spatially but also temporally. These applications are diverse, encompassing fields such as traffic analysis and surveillance, medical imaging, and cell biology (Ghosh et al. 2019). While a relatively straightforward technique, thresholding has inherent limitations in distinguishing between background, noise, and foreground. Therefore, the next section will offer a more sophisticated approach, by presenting a package that utilizes deep learning for image segmentation (Smith et al. 2021).

3.1 **imageseg**: a deep learning package for forest structure analysis

By venturing beyond the traditional laboratory setting, the **imageseg** package offers a unique approach to analyzing forest structures through deep learning-based image segmentation, utilizing TensorFlow (<https://www.tensorflow.org/>). This R package employs the power of convolutional neural networks with the U-Net architecture to streamline image segmentation tasks (Jürgen Niedballa et al. 2022b). According to the authors, this R package has been designed to be user-friendly, with pre-trained models that require only input images, making it accessible even to those without specialist knowledge. A comprehensive vignette accompanies the package, which provides detailed instructions on how to set up the software and explains how to utilize its functions effectively (Juergen Niedballa et al. 2022). Developed primarily for forestry and ecology applications, **imageseg** includes pre-trained data sets representing various aspects of forest structure, such as canopy and understory vegetation density. Its flexibility allows for customization with different training data, enabling users to develop customized image segmentation workflows for other fields such as microscopy and cell biology. The package supports both binary and multiclass segmentation. For image processing within the R programming environment, the **imageseg** package integrates with the **magick** package (Jürgen Niedballa et al. 2022b).

3.2 **EBIImage**: specialized segmentation strategy for touching objects

The segmentation of closely adjacent objects, which is particularly prevalent in cell microscopy, represents a common challenge that is addressed by the **EBIImage** package, which is equipped with a variety of segmentation algorithms. A typical approach involves the application of either global or adaptive thresholding, followed by connected set labeling, with the objective of distinguishing individual objects. To achieve more precise segmentation of touching objects, techniques such as watershed transformation or Voronoi segmentation are employed (Pau et al. 2010).

The watershed algorithm is employed to delineate touching microbeads (Figure 1A-C). Initially, the image is transformed into a binary image by applying a threshold (Figure 1B). After utilizing the watershed function the result is visualized by assigning distinct colors to the microbeads, effectively illustrating the algorithm's capacity to differentiate between touching objects (Figure 1C).

```
# Load necessary library
library(EBIImage)

# Load the image from the specified path
image <- readImage("figures/beads.png")

# Display the original image
EBIImage::display(image)

# Apply a threshold to the original image to create a binary image
img_thresh <- thresh(image, offset = 0.05)

# Read the binary image and display it
EBIImage::display(img_thresh)

# Perform watershed segmentation on the distance map of the thresholded image
segmented <- EBIImage::watershed(distmap(img_thresh))

# Color the labels of the segmented image
```

```

segmented_col <- colorLabels(segmented)

# Display the resulting image after watershed segmentation
EBImage::display(segmented_col)

```

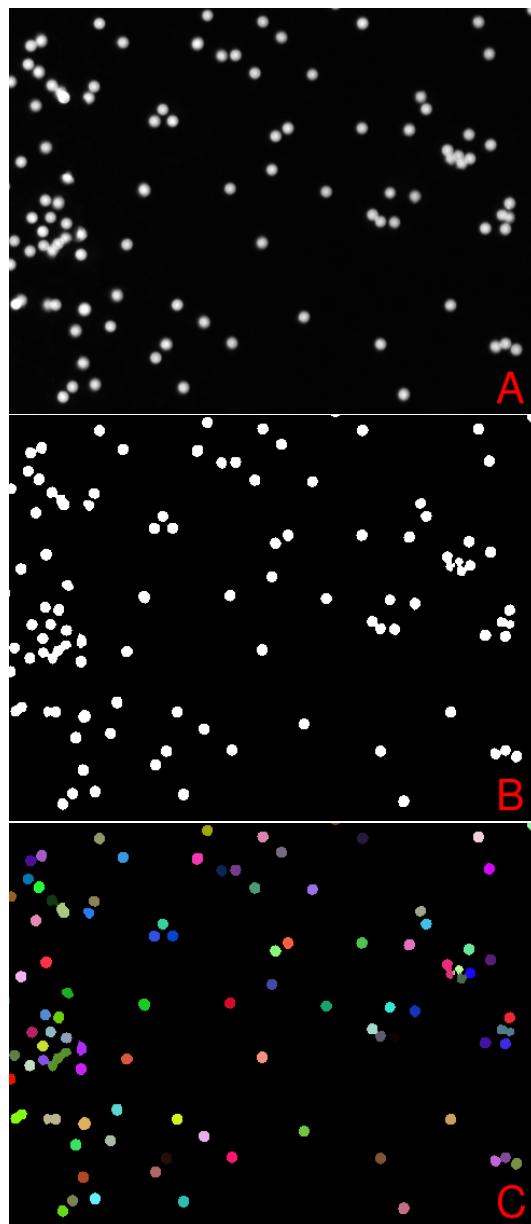


Figure 1: Watershed Segmentation in EBImage: A) Original image used for watershed segmentation in EBImage. B) The `thresh()` function was employed to generate a binary image with the objective of effectively separating the foreground from the background. The binary representation of the image facilitates further segmentation processes by simplifying the image. C) Presents the result of the watershed segmentation, which is visually represented by the assignment of a distinct color to each object. This technique is particularly effective in differentiating touching objects, as evidenced by the clear separation of microbeads in the image.

4 Unveiling the hidden - feature extraction

The primary objective of feature extraction is to condense the original data into significant objects that encapsulate crucial information pertinent to each specific image (Jude Hemanth and Anitha 2012). Feature extraction may be applied to a predefined region of interest (ROI) or may involve the identification of the ROI, a process often referred to as segmentation, which was reviewed in

the previous sections. Within any given ROI, a multitude of attributes typically exist, representing different states of the object under analysis. These attributes, or features, are of vital importance for the interpretation of the detected objects and can enable applications such as disease diagnosis or the identification of promising candidates. Features related to individual pixels may include aspects such as neighborhood relationships, connectivity, and gradients, which are one-dimensional descriptions. Nevertheless, more intelligible and interpretable information is frequently derived from descriptions of regions or objects (Sonka and Fitzpatrick 2000; Shirazi et al. 2018). Object-level features encompass a range of characteristics, including size, shape, texture, intensity, and spatial distribution. Shape features can be further categorized into specific characteristics, including perimeter, radius, circularity, and area. It is crucial to acknowledge that the successful extraction of object features is dependent on the quality and accuracy of the image segmentation process (Shirazi et al. 2018).

This section is devoted to an examination of R packages that enable the automated extraction of quantitative features. The `biopixR` package offers automated and interactive object detection strategies. The `pliman` package, initially developed for the analysis of plant images, has the potential to be adaptable to a range of different domains. The `FIELDimageR` package is capable of supporting the analysis of drone-captured images from agricultural field trials as well as images from pollen, which exhibit similar characteristics to cellular images. These tools provide novel perspectives for interdisciplinary research, facilitating the adaptation of methodologies across diverse fields.

4.1 `biopixR`: versatile biological image processing

The `biopixR` package is a comprehensive toolbox developed primarily for microbead analysis. It encompasses a range of functions, including image importation, preprocessing, segmentation, feature extraction, and clustering. The primary objective is to enable the detection of objects and the extraction of quantitative data, including intensity values, shape, and texture characteristics. These functionalities are integrated into user-friendly pipelines that support batch processing, thereby enhancing accessibility. The preprocessing capabilities include edge restoration and a variety of filter functions (Brauckhoff, Kieffer, and Rödiger 2024).

To illustrate the feature extraction process, the analysis focuses on a microbead image (Figure 2A). The image is initially converted to grayscale. Afterward the `objectDetection()` function is applied to detect image objects. The extracted objects are then represented visually by plotting the highlighted contours of the objects and enumerating the microbeads according to their cluster IDs, thus distinguishing them as individual entities (Figure 2B).

```
# Loading necessary package
library(biopixR)

# Importing the image
beads <- importImage("figures/beads2.jpg")

# Plot original image
beads |> plot(axes = FALSE)

# Converting the image to grayscale
beads <- grayscale(beads)

# Detecting objects in the image using edge detection
objects <-
  objectDetection(beads,           # Image to process
                  method = 'edge',    # Method for object detection
                  alpha = 1,          # Threshold adjustment factor
                  sigma = 0)         # Smoothing factor

# Displaying internal visualization of object detection with marked contours
# and centers
objects$marked_objects |> plot(axes = FALSE)

# Adding text annotations at the centers of detected objects
text(objects$centers$mx,        # x-coordinates of object centers
     objects$centers$my,        # y-coordinates of object centers
     objects$centers$value,    # Text to display (value of the object center)
     col = "green",            # Color of the text
     cex = 1.5)
```

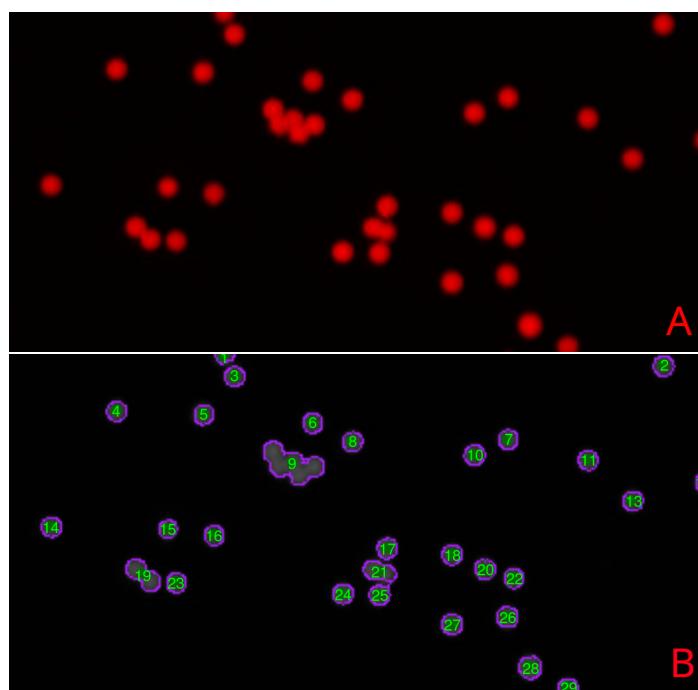


Figure 2: Microbead Detection using biopixR: **A)** The original image shows red fluorescent microbeads, with the majority appearing as isolated, round, spherical objects. Some microbeads are clustered together or overlapping, forming aggregated structures, while others are partially captured within the image frame. **B)** In the grayscale microbead image, edges of the microbeads are highlighted in purple, and the labeling ID (value) is displayed at the center of each object in green.

4.2 pliman: an R package for plant image analysis

pliman is designed to analyze plant images, particularly leaves and seeds, to help identify disease states, lesion shapes, and quantifying objects. It supports various functions, including image transformation, binarization, segmentation, and detailed analysis, all facilitated by a detailed vignette.⁶ A key feature of pliman is its automation of quantitative feature extraction (Figure 3 and 4), which traditionally requires manual, time-consuming, and error-prone methods. The features of this package are versatile, encompassing a range of segmentation strategies, the analysis of shape and contour characteristics of leaves and seeds, the counting of objects, and the quantification of disease states from leaf images. While the primary focus is on plant imaging, the techniques used are applicable to other fields such as cellular imaging. This cross-applicability is further emphasized by the package's batch processing capabilities, which allow for autonomous analysis of multiple images, critical for high-throughput phenotyping tasks (Olivoto 2022).

```
# Loading necessary package
library(pliman)

# Import requires EBImage:
# Importing the main image
beads <- EBImage::readImage("figures/beads2.jpg")

# Importing additional images for background and foreground
foreground <- EBImage::readImage("figures/foreground.jpg")
background <- EBImage::readImage("figures/background.jpg")

# Displaying the microbead image
EBImage::display(beads)

# Combining the foreground and background images and arranging them in 2 rows
pliman::image_combine(foreground, background, nrow = 2, col = "transparent")
```

⁶<https://tiagoolivoto.github.io/pliman/index.html>, accessed 07/11/2024

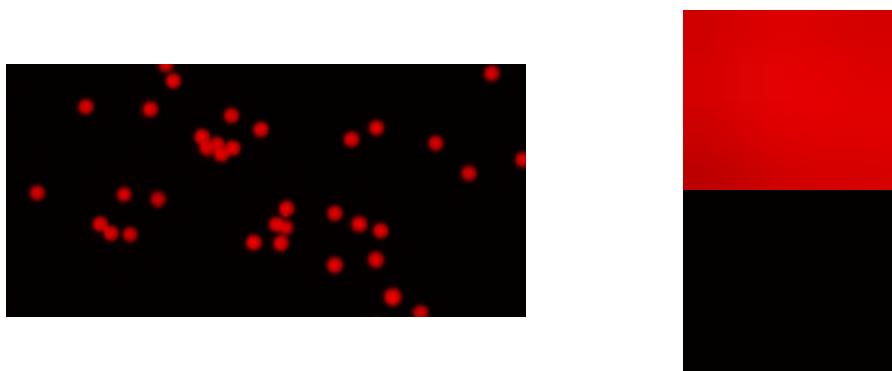


Figure 3: Preparing Segmentation using `pliman`: The image comprises two sections. On the left, an image of microbeads is displayed. On the right, a cropped view from the same image illustrates two states for segmentation: the microbead (foreground) in red, and the background is shown in black, emphasizing the clear division needed for segmentation analysis.

```
# Performing segmentation based on provided background and foreground images
analyze_objects(
  img = beads,                      # Main image of microbeads
  background = background,           # Background sample image
  foreground = foreground,           # Foreground sample image
  marker = "id",                   # Displaying enumeration
  contour_col = "yellow"           # Color for the contour of the segmented objects
)
```

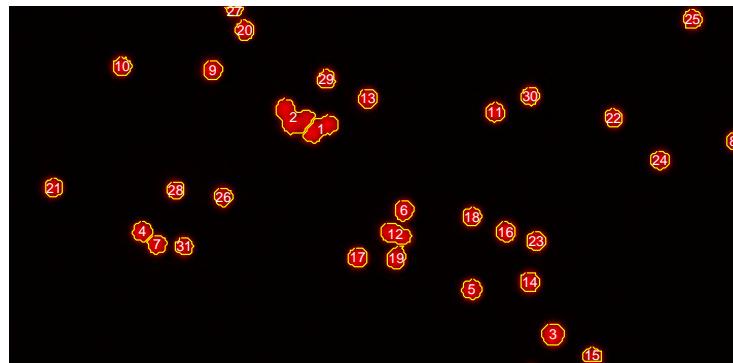


Figure 4: Segmentation Results using `pliman`: The image depicts the segmentation results obtained via the `pliman` `analyze_objects()` function. It displays the contours of the segmented objects, outlined in yellow. Each distinct object within the segmentation is numbered, facilitating its identification.

4.3 FIELDimageR: an R package for the analysis of drone-captured images

The FIELDimageR package, is an R package designed for the specific purpose of analyzing drone-captured images from agricultural field trials. The package offers a variety of functions for the ROI selection, the extraction of foregrounds (Figure 5), watershed segmentation, quantification and shape analysis (Matias, Caraza-Harter, and Endelman 2020). The developers have applied this package to analyze pollen, which visually resembles cells under a microscope. This suggests that FIELDimageR may be applicable for use in microbiological image analysis. For the spatial analysis the package utilizes the terra package (Matias, Caraza-Harter, and Endelman 2020).⁷

To showcase the functionalities of the FIELDimageR package and its parallels with biological applications, the same microbead image is subjected to analysis. The image is initially transformed

⁷<https://github.com/OpenDroneMap/FIELDimageR>, accessed 05/07/2024

into a ‘SpatRaster’ object and then segmented using an intensity threshold (Figure 5). The microbeads are correctly identified as the foreground objects by the `fieldMask()` function. Subsequently, a distinct labeling ID is assigned to each microbead, as illustrated by a color gradient. Moreover, the contours of each individual object are displayed (Figure 6). The results of the segmentation and the extraction of shape-related information are presented in the interactive leaflet interface (Figure 7). Presenting information like cluster ID, size, perimeter and width of the detected objects.

```
# Loading necessary packages
library(FIELDimageR)
library(FIELDimageR.Extra)
library(terra)
library(sf)
library(leafsync)
library(mapview)

# Using the same image as imported in the previous example
# Creating a SpatRaster object using the 'terra' package
EX.P <- rast("figures/beads2.jpg")
EX.P <- imgLAB(EX.P)

#> [1] "3 layers available"

# Removing background based on a vegetation index
EX.P.R1 <-
  fieldMask(
    mosaic = EX.P,      # Input SpatRaster object
    index = "BIM",      # Index representing vegetation
    cropValue = 5,       # Threshold value for the index
    cropAbove = F        # Indicates to remove values below the threshold
  )

# Displaying the original, background, and foreground images
EX.P.R1$newMosaic
```



Figure 5: Displaying the original, background, and foreground Images: The original image (left) shows the fluorescent microbeads. The middle image displays the background in white and all objects detected by segmentation in black (FALSE). The right image shows only the foreground (microbeads) after detection through segmentation using the `fieldMASK()` function.

```
# Labeling of all microbeads
EX.P.Total <- fieldCount(mosaic = EX.P.R1$mask, plot = T)
```

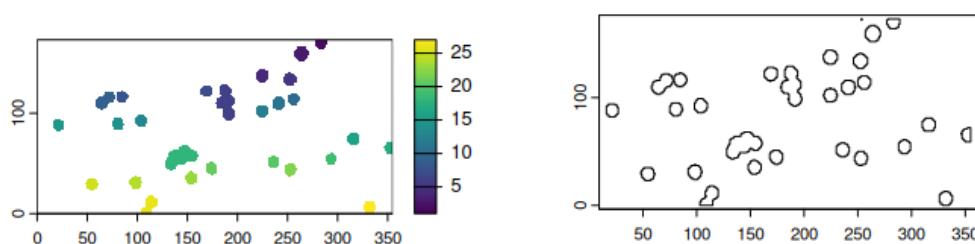


Figure 6: Labeling of Microbeads: The `fieldCount()` function is used to label individual microbeads. This function utilizes the mask produced in the previous section to identify the objects. The left image displays the labeling with a color gradient indicating distinct objects. On the right, the object contours are shown. The output of the function includes more than just the labeling value (named ID in this package); it also provides information on area, perimeter, width, and geometry of the detected objects.

```
# Combining the 'FIELDimageR.Extra', 'mapview' and 'leafsync' to create an
# interactive view
m1 <- fieldView(EX.P, r = 1, g = 2, b = 3)
m2 <- mapview(EX.P.Total)
sync(m1, m2)
```

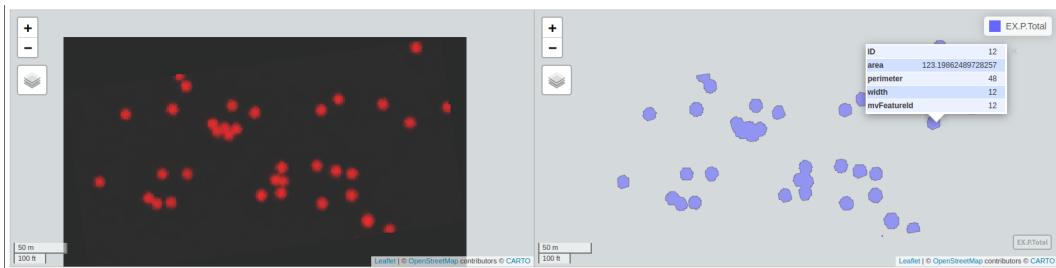


Figure 7: Displaying Results with an Interactive leaflet Tool: The tool displays the original image on the left. For comparison, the cursor is mirrored to the corresponding image (only visible in HTML format). The left image provides detailed information interactively. Hovering over the objects reveals their labeling ID. Performing a left-click opens a detailed window providing information for the individual object, such as area, perimeter, width, and shape. The packages FIELDimageR.Extra, mapview, and leafsync are used to create the interactive display.

In summary, packages such as EBImage and biopixR provide direct pipelines for the extraction of features from images, including shape, size, radius, and perimeter, as well as texture information through the calculation of Haralick texture features (Haralick, Shanmugam, and Dinstein 1973; Pau et al. 2010; Brauckhoff, Kieffer, and Rödiger 2024). The biopixR package employs the imager and magick packages for image processing (Brauckhoff, Kieffer, and Rödiger 2024), whereas pliman and FIELDimageR rely on EBImage for direct image analysis, with FIELDimageR also utilizing terra and raster for spatial data exploration (Matias, Caraza-Harter, and Endelman 2020; Olivoto 2022). In comparison to the other packages discussed in this section, biopixR facilitates the process of object detection by eliminating the necessity for the generation of masks or the provision of representative sample images of the foreground and background. Nevertheless, in contrast to the other packages, biopixR lacks the functionality of watershed segmentation for the enhanced handling of touching objects (Figure 2B and Figure 4) (Matias, Caraza-Harter, and Endelman 2020; Olivoto 2022; Brauckhoff, Kieffer, and Rödiger 2024).

5 Decoding complexity - clustering, classification and annotation

The automation of measuring cellular phenomena and the effects of compounds, which started in the late 1990s, is now increasingly significant owing to the progress of machine learning (ML) algorithms and computing power. These advancements are enhancing the field of bioinformatics' accessibility to these techniques. Consequently, they are being more commonly employed with the aim of gaining novel biological insights (Murphy 2014; Moen et al. 2019; Weiss et al. 2022). One of the latest methods of image analysis involves comparing the morphological characteristics of cells from captured images with pre-classified training data that represents a specific state (Moen et al. 2019). Bioimage informatics methods aim to generate fully automated models for biological systems (Murphy 2014).

A major challenge in handling new data sets is the need to label images, which is critical to assigning meaning to the objects within them. This is particularly important in medical imaging, where expert knowledge is essential for accurate labeling (Boom et al. 2012; Weiss et al. 2022). In ML, two common techniques that can be used to categorize data into distinct groups are clustering and classification. Clustering, an unsupervised learning method, is used to discover underlying structures or patterns in unlabeled data by assessing similarities between data points (Mostafa and Amano 2019). Classification, a form of supervised learning, involves building a model from previously labeled training data to make predictions about new data (Mostafa and Amano 2019; Kumar Dubey, Gupta, and Jain 2022). This requires prior labeling of the data to determine the characteristics of each group, a process known as annotation. However, manual annotation is time-consuming and labor-intensive, requiring significant human effort to identify relevant details in an image (Yao et al. 2016; Weiss et al. 2022). Because images often require multi-label annotation - the assignment of multiple semantic concepts to a single image - there has been a growing demand for automated image annotation systems that aim to reduce the burden of manual labeling and increase the efficiency of data processing (Nasierding, Tsoumacas, and Kouzani 2009).

To effectively analyze complex image data sets, researchers require advanced pattern recognition techniques that can extract meaningful biological insights from these images. This enables them to transform visual data into actionable scientific knowledge (Behura 2021). Some of the most widely used clustering algorithms for this purpose include:

- k-means: is a centroid-based algorithm that partitions n observations into k clusters by minimizing within-cluster sum of squares. It does require specifying the number of clusters beforehand (Struyf, Hubert, and Rousseeuw 1996).
- Partitioning Around Medoids (PAM): a k-means relative, seeks to identify k representative objects from the data set, which are robust representations of the clusters' center and are also referred to as medoids. Clusters are formed by assigning each object to its nearest medoid, with the objective of optimizing within-cluster similarity (Kaufman and Rousseeuw 1990; Van der Laan, Pollard, and Bryan 2003).
- c-means: also known as Fuzzy C-Means (FCM), extends the concept of k-means to allow each data point to belong to more than one cluster (Bezdek, Ehrlich, and Full 1984).
- Density-Based Spatial Clustering of Applications with Noise (DBSCAN): is a density-based clustering algorithm that groups together points that are closely packed together and separates them from points that lie alone in low-density regions. It does not require specifying the number of clusters beforehand (Ester et al. 1996; Schubert et al. 2017).
- Self-Organizing Maps (SOM): are a type of neural network architecture that systematically organizes input features into a spatially coherent representation. This method can be utilized for clustering based on various object features, thereby facilitating the discovery of patterns within these objects (Kohonen 1990, 2013).

5.1 pixelclasser: a simplified support vector machine approach for pixel classification

The `pixelclasser` package is a tool for classifying image pixels into user-defined color categories using a simplified version of the Support Vector Machine (SVM) technique. It includes functions that allow users to visualize image pixels, define classification rules, classify pixels, and store the resulting information.⁸ Users must provide a test set that captures the variation between categories, as the package requires manual placement of rules for each category - automatic rule construction methods are not included. In addition, `pixelclasser` provides quality control of the classifications and comes with a detailed vignette to facilitate the use of this classification tool.⁹ The classification on the pixel-level can be used for image segmentation via pixel clustering.

5.2 biopixR: pattern recognition of shape- and texture-related features

The `biopixR` package incorporates two unsupervised ML clustering algorithms: SOM and PAM. PAM organizes a distance matrix into clusters, identifying medoids as robust representatives of each cluster, typically specified with a predefined number of groups (k) (Kaufman and Rousseeuw 1990; Van der Laan, Pollard, and Bryan 2003; Park and Jun 2009). This approach clusters Haralick texture features extracted from multiple images within a directory, thereby enabling image classification based on these features (Haralick, Shanmugam, and Dinstein 1973). The optimal number of clusters (k) is automatically determined using silhouette analysis (Rousseeuw 1987; Brauckhoff, Kieffer, and Rödiger 2024). SOM is used to cluster object features related to object shape and intensity, thereby facilitating the identification of patterns within these characteristics (Brauckhoff, Kieffer, and Rödiger 2024).

The capacity for pattern recognition within the `biopixR` package is demonstrated by the clustering of shape-related and pixel-intensity information from an example image of microbeads (Figure 8A). The image depicts both single and aggregated microbeads, wherein the former exhibit a round, spherical shape, while the latter appear more oval. The extracted features and the corresponding cluster are depicted in Figure 8B, which showcases the identification of patterns within these objects based on their shape characteristics.

```
# Load the 'biopixR' package
library(biopixR)

# Import an image from the specified path
img <- importImage("figures/beads.png")
```

⁸<https://github.com/ropensci/pixelclasser>, accessed 07/11/2024

⁹<https://cloud.r-project.org/web/packages/pixelclasser/vignettes/pixelclasser.html>, accessed 07/11/2024

```
# Set seed for reproducibility
set.seed(123)

# Extract shape features from the image
result <- shapeFeatures(
  img,
  alpha = 0.8,
  sigma = 0.7,
  xdim = 2,
  ydim = 1,
  SOM = TRUE,
  visualize = FALSE
)

# Define colors for plotting points based on classes
colors <- c("darkgreen", "darkred")

# Plot the image without axes and add colored points representing the classes
img |> plot(axes = FALSE)
with(result,
  points(
    result$x,
    result$y,
    col = colors[factor(result$class)],
    pch = 19,
    cex = 1.2
  ))
text(c(471), c(354), c("A"), col = "darkred", cex = 5)

# Create a data frame with various shape features and the pixel-intensity
df <- data.frame(
  size = result$size,
  intensity = result$intensity,
  perimeter = result$perimeter,
  circularity = result$circularity,
  eccentricity = result$eccentricity,
  radius = result$mean_radius,
  aspectRatio = result$aspect_ratio
)

# Min-Max Normalization Function
min_max_norm <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}

# Applying the function to each column
df_normalized <- as.data.frame(lapply(df, min_max_norm))

# Create a boxplot of the normalized data
boxplot(
  df_normalized,
  ylab = "normalized values",
  xaxt = "n",
  cex.lab = 1.25,
  cex.axis = 1.25
)

# Add axis ticks and diagonal labels
axis(1, at = 1:ncol(df), labels = FALSE) # Add axis ticks but no labels
text(
  cex = 1.2,
  x = seq_len(ncol(df_normalized)),
  y = -0.07,
  labels = colnames(df_normalized),
```

```
adj = 0,  
srt = -45,  
xpd = TRUE  
)  
  
# Highlight specific rows based on class  
highlight_rows <-  
  which(result$class == 2) # Example row indices to highlight  
  
# Add points for the specific rows  
# Adding points for each column  
for (col in 1:ncol(df_normalized)) {  
  points(  
    rep(col, length(highlight_rows)),  
    df_normalized[highlight_rows, col],  
    col = "red",  
    pch = 19,  
    cex = 1.5  
  )  
}  
  
text(c(0.5),  
  c(0.98),  
  c("B"),  
  col = "darkred",  
  cex = 5)
```

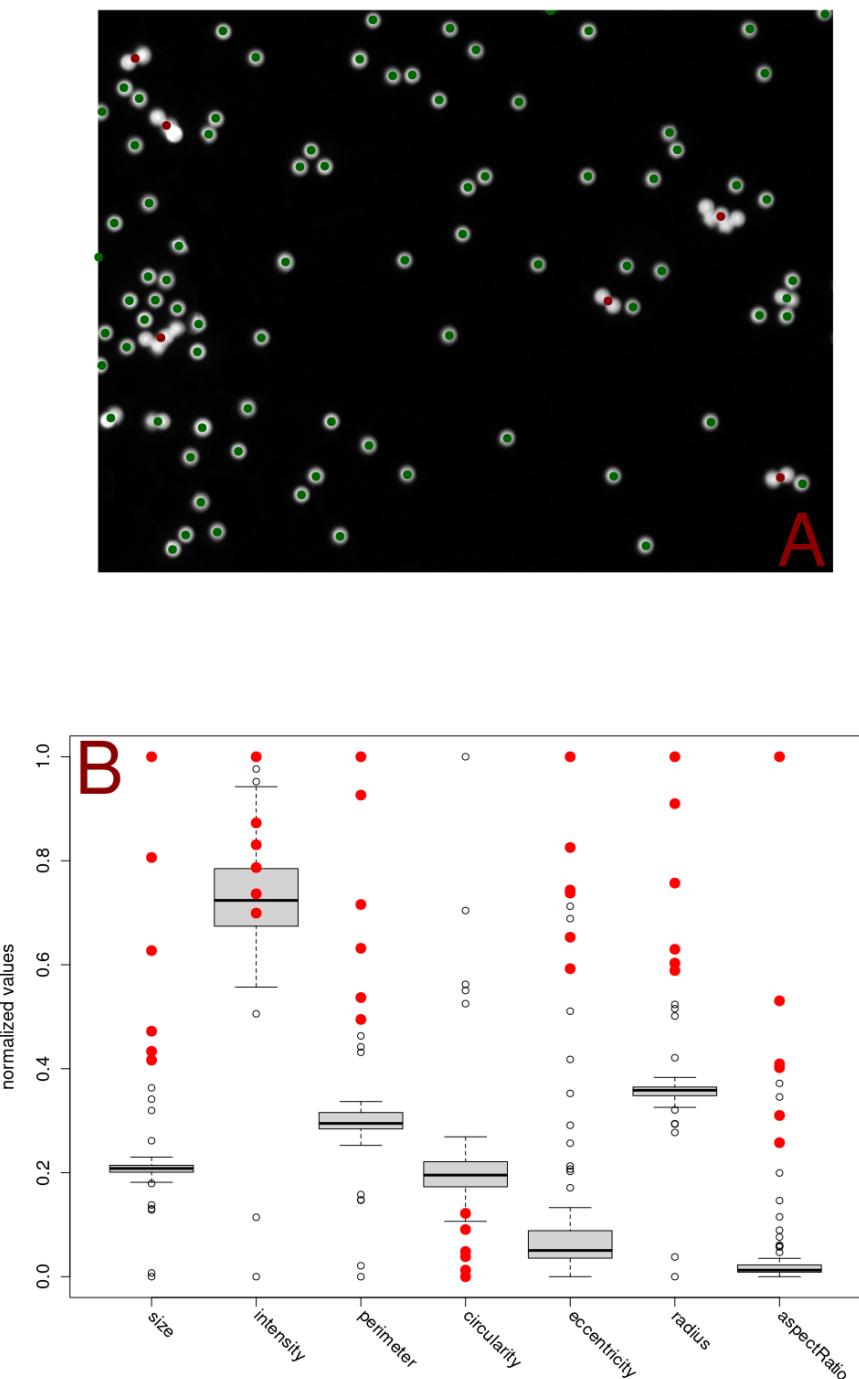


Figure 8: Clustering Microbeads Based on Shape and Intensity Features: A) The utilization of Self-Organizing Maps (SOM) enables the clustering of microbeads into two distinct groups based on shape and intensity features extracted using the `shapeFeatures()` function. This method enables the precise clustering of microbeads according to a range of properties, including intensity, area, perimeter, circularity, radius, and aspect ratio. This facilitates a deeper understanding of the morphological variations observed in the microbeads. B) The attributes utilized as input for the SOM algorithm are illustrated in this plot. To ensure comparability, the different parameters have been normalized using a min-max normalization procedure. The points highlighted in red represent the microbeads that are also highlighted in red in Figure A. Notably, these highlighted points differ from the most commonly occurring values in all attributes except for the intensity.

6 Harmonizing visions - techniques and approaches in image registration

The process of image registration plays a pivotal role in the analysis of medical images, as it enables the comparison of multiple images representing different conditions (Jenkinson and Smith 2001). This process, which can be described as image alignment, entails aligning a series of images within a single coordinate system, thereby ensuring consistency across images (Peng 2008; Rittscher 2010). A variety of techniques are employed in image registration, including mutual information registration, spline-based elastic registration, and invariant moment feature-based registration, among others (Peng 2008). These methods are of particular significance in the field of medical imaging, where they are employed to enhance the analysis of images obtained by techniques such as computed tomography (CT) and magnetic resonance imaging (MRI) (Sonka and Fitzpatrick 2000).

6.1 RNiftyReg: interface for the ‘NiftyReg’ image registration tools

The RNiftyReg package provides an interface to the ‘NiftyReg’ image registration library, which supports both linear and non-linear registration in two and three dimensions (J. Clayden et al. 2023). This package has been utilized in research on brain connectivity (J. D. Clayden, Dayan, and Clark 2013), and it includes a comprehensive README that introduces its features and capabilities.¹⁰

7 Jack of all trades - general purpose R packages for broad-spectrum analysis

Five principal image processing packages for R offer a broad range of algorithms and capabilities for complete image analysis, rendering them suitable as general-purpose tools. These packages are imager, magick, EBImage, OpenImageR and SimpleITK. This section will introduce each of these key packages and their roles in image analysis.

7.1 imager: wrapper for the ‘CImg’ C++ image processing library

The imager R package, created by Barthelmé and Tschumperlé (2019), integrates the functionality of the ‘CImg’ library, developed by David Tschumperlé, into R.¹¹ This allows users to edit and create images. The package uses two primary data structures: raster images, known as *cimg*, and pixel sets, referred to as *pixelset*. These structures, encoded as four-dimensional numeric or logical arrays, permit the execution of basic R functions such as `plot()`, `print()`, or `as.data.frame()`, as well as the processing of hyperspectral images and videos (Barthelmé and Tschumperlé 2019). The 4D arrays encompass two spatial dimensions (width and height), one temporal or depth dimension, and one color dimension (Simon Barthelme 2024). imager offers over 100 standard commands for tasks such as loading, saving, resizing, and denoising of images.¹² The imager package supports the file formats JPEG, PNG, and BMP and is available on CRAN (Simon Barthelme 2024).

7.2 EBImage: image processing and analysis for biological imaging data in R

The EBImage package, established in 2006, is one of the oldest image processing tools available in R and can be accessed via the Bioconductor repository. It is primarily written in R and C/C++ (Andrzej Oleś 2017). EBImage provides a suite of general tools for image processing and analysis, particularly excelling in microscopy-based cell assays. It features specialized commands for cell segmentation and the extraction of quantitative data from images (Pau et al. 2010). The package employs the RGB color system for color detection, which is based on pixel intensities. The incorporation of the EBImage package into the R workflow facilitates the automation and objectivity of the image analysis procedure (Heineck et al. 2019). Images in EBImage are managed as an extension of R’s base array, specifically the package-specific Image class. As images are treated as multidimensional arrays, algebraic operations are possible. This class structure includes various slots, with the `.data` slot holding the numeric pixel intensity array and the `colorMode` slot managing the image’s color information. Adjusting the `colorMode` setting changes the image’s rendering mode (Andrzej Oleś 2017; Heineck et al. 2019). Typically, the first two dimensions of an image carry spatial information, while additional dimensions are variable and can represent color channels, time points, replicas, or depth. EBImage also features an interactive display interface through GTK+, and offers a set of functions for automated image-based phenotyping

¹⁰<https://github.com/jonclayden/RNiftyReg>, accessed 07/11/2024

¹¹<https://github.com/asgr/imager>, accessed 07/11/2024

¹²<https://asgr.github.io/imager/>, accessed 07/11/2024

in biology, including cell segmentation, feature extraction, statistical analysis, and visualization (Pau et al. 2010). It supports a range of file formats, including JPEG, PNG, and TIFF, and can handle additional formats through integration with the ‘ImageMagick’ image-processing library (Pau et al. 2010; Andrzej Oleś 2017).

7.3 `magick`: advanced image processing in R using ‘ImageMagick’

This package is built upon ‘Magick++’, the C++ API for the ‘ImageMagick’ image processing library.¹³ The R package provides access to ‘ImageMagick’ functionalities, enabling both basic and complex image manipulations directly in R. Notably, images in `magick` are automatically displayed in the RStudio console, creating a dynamic and interactive editing environment. The wide variety of functions made available through this package are impressive. The possibilities range from functions that are rather ‘just for fun’, such as implosion or introduction of noise, to more advanced processing techniques, including different segmentation techniques, edge detection, and a toolbox for morphology operations. The `magick` package is compatible with a diverse range of image formats and encompasses the functionalities required for format conversion. This includes the conversion to the formats supported by the `EBImage` package. It also handles multiple frames, facilitating the creation and processing of animated graphics. Each operation in `magick` creates a new, altered version of the image, preserving the original (Ooms 2024a).¹⁴ Recent developments include the introduction of a `shiny` application that enables users to interactively perform basic image processing tasks such as blurring and edge detection.¹⁵ The `magick` package is compatible with a range of popular file formats, including PNG, BMP, TIFF, PDF, SVG, and JPEG, and is available through the CRAN repository (Ooms 2024a).¹⁶

7.4 `OpenImageR`: a general-purpose image processing library

`OpenImageR` is a lesser known but highly versatile general-purpose image processing library that integrates both the R and C++ programming languages. This package offers a comprehensive array of functions for preprocessing, filtering, and feature extraction. Images are treated as two- or three-dimensional objects, represented by matrices, data frames, or arrays, with the third dimension representing color information. The functionalities within `OpenImageR` are organized into three main categories: basic functions, which include importing, displaying, cropping, and thresholding; filter functions, which feature augmentation and various edge detection algorithms; and image recognition, which incorporates functions from the ‘ImageHash’ Python library. In recent updates, a number of new features have been incorporated, including Gabor feature extraction, which was originally developed in MATLAB and based on code by Haghigiat, Zonouz, and Abdel-Mottaleb (2015). The most recent version incorporates image segmentation techniques that utilize superpixels and clustering. Images can be visualized through the `shiny` application or the `grid` package. `OpenImageR` is capable of handling a multitude of image formats, including PNG, TIFF, and JPG (Mouselimis et al. 2023).¹⁷ ¹⁸

7.5 `SimpleITK`: a streamlined wrapper for ITK in biomedical image analysis

The following section will introduce a prominent tool in biomedical image analysis, the wrapper for the Insight Segmentation and Registration Toolkit (ITK), known as `SimpleITK` (Rittscher 2010). `SimpleITK` represents a streamlined version of the original ITK, an open-source C++ library that features a wide array of imaging algorithms and frameworks (Lowekamp et al. 2013; Yaniv et al. 2017). This library has been in development for approximately two decades and is particularly favored in the medical image analysis community (Lowekamp et al. 2013; Beare, Lowekamp, and Yaniv 2018). The objective of `SimpleITK` is to simplify the accessibility of ITK algorithms by reducing their complexity, thereby making these sophisticated tools more approachable for a broader audience (Lowekamp et al. 2013). Adapted for the R programming language through SWIG, `SimpleITK` offers over 250 image processing algorithms that function across various scripting and prototyping environments (Lowekamp et al. 2013; Yaniv et al. 2017; Beare, Lowekamp, and Yaniv 2018). In contrast to other general-purpose image processing packages, which treat images as mere arrays, `SimpleITK` treats images as objects within a physical space, thereby providing a set of metadata about image and voxel geometry in world coordinates (Lowekamp et al. 2013; Yaniv et al. 2017; Beare, Lowekamp, and Yaniv 2018). This nuanced representation is of particular importance for specific medical imaging

¹³<https://imagemagick.org/script/magick++.php>, accessed 07/11/2024

¹⁴<https://www.imagemagick.org/Magick++/ImageDesign.html>, accessed 07/11/2024

¹⁵<https://georgestagg.github.io/shinymagick/>, accessed 07/11/2024

¹⁶<https://imagemagick.org/>, accessed 07/11/2024

¹⁷<https://github.com/mlampros/OpenImageR>, accessed 07/11/2024

¹⁸<https://mlampros.github.io/OpenImageR/index.html>, accessed 07/11/2024

applications. Additionally, SimpleITK incorporates metadata such as the origin, pixel spacing, and a matrix defining the physical orientation of image axes (Yaniv et al. 2017). However, the complexity of the underlying ITK library may impede customization and necessitate familiarity with C++. Another challenge for R developers arises from the fact that the documentation is also based on C++ (Beare, Lowekamp, and Yaniv 2018). To facilitate the learning process, Yaniv et al. (2017) has developed a series of Jupyter notebooks that provide an introduction to the package and its capabilities for both Python and R users. These notebooks serve as educational tools and a resource for research, providing full coverage of the entire spectrum of image analysis processes (Beare, Lowekamp, and Yaniv 2018).¹⁹ In combination with R, SimpleITK enables detailed image processing and facilitates the subsequent statistical evaluation of quantified data. The software is compatible with a range of digital image formats, including JPEG, BMP, PNG, and TIFF, and is capable of analyzing 2D and 3D images (Beare, Lowekamp, and Yaniv 2018). The package is obtained through the GitHub repository.²⁰

In summary, these packages and their associated libraries offer a vast array of algorithms that can be accessed in R. This includes features from the 'CImg', 'ImageMagick' and ITK libraries, along with the diverse algorithms encoded in the EBImage package. These flexible packages provide the foundation for the development of numerous tailored applications.

8 Exploring the facets of complexity - multiplexed imaging in R

Multiplexed imaging is a crucial technology for analyzing complex biological processes at the single-cell level, especially in tissue-based cancers and autoimmune diseases (C. Harris, Wrobel, and Vandekar 2022). This technique enables the simultaneous assessment of multiple protein and DNA molecules, overcoming limitations that hinder advancements in understanding biological interactions and phenomena (Gerdes et al. 2013; Goltsev et al. 2018). Multiplex imaging is the result of a multiplex experiment, in which multiple species (Aherne et al. 2024), biomolecules (Damond et al. 2019), or cell types (J. H. Creed et al. 2021) are labeled with different probes, dyes, or antibodies simultaneously. This technique allows for the differentiation of components within the resulting image (Eling et al. 2020). In comparison to standard immunofluorescence experiments, the number of distinct targets is significantly increased, reaching up to 50 different target molecules (Damond et al. 2019; Einhaus et al. 2023). This can be used to distinguish between species in a biofilm (Aherne et al. 2024), or to obtain an overview of the biomarker distribution or tissue composition in a sample (Damond et al. 2019; Yang et al. 2020). The technique has the capacity to reveal the positions and interactions of individual cells, provide insight into the activities of biomolecules, and holds the potential for the reconstruction of the three-dimensional tissue architecture of a given sample (C. R. Harris et al. 2022; Cho, Kim, and Park 2023; Zhao and Germain 2023). Several imaging techniques are used to obtain detailed insights into the spatial interactions between cells, including Co-Detection by indEXing (CODEX) (Goltsev et al. 2018), Multiplex Ion Beam Imaging (MIBI) (Angelo et al. 2014), and Multiplexed Immunofluorescence Imaging (MxIF) (Gerdes et al. 2013; C. Harris, Wrobel, and Vandekar 2022; Feng et al. 2023). These methods generate vast amounts of imaging data, often terabytes across hundreds of slides, which necessitates sophisticated image analysis pipelines (C. R. Harris et al. 2022).

8.1 mxnorm: normalize multiplexed imaging data

Managing technical variability within these pipelines is crucial, and intensity normalization is one approach to address this issue (C. R. Harris et al. 2022). The R package `mxnorm` addresses this by providing tools for implementing, evaluating, and visualizing various normalization techniques (C. Harris 2023). These tools aid in measuring technical variability and evaluating the efficacy of various normalization methods. They enable users to apply customized methods to improve image consistency by reducing technical variations while preserving biological signals. `mxnorm` provides an analysis pipeline for multiplex images, incorporating normalization algorithms inspired by the ComBat paper, the `fda` package, and the `tidyverse` framework (C. Harris, Wrobel, and Vandekar 2022). For researchers who want to effectively standardize multiplexed imaging data, these features make `mxnorm` a powerful resource (C. Harris 2023).

8.2 DIMPLE: manipulation and exploration of multiplex images

To assess patient outcomes, understand disease mechanisms, and develop effective cancer therapies, the DIMPLE R package is designed to extract critical information from the tumor microenvironment (TME). DIMPLE facilitates quantification and visualization of cellular interactions within the TME using

¹⁹<https://github.com/InsightSoftwareConsortium/SimpleITK-Notebooks>, accessed 07/11/2024

²⁰<https://github.com/SimpleITK/SimpleITKInstaller>, accessed 07/11/2024

spatial data. It also enables correlation of these interactions and phenotypic data with patient outcomes through sophisticated statistical modeling. DIMPLE provides researchers with an extensive toolkit to analyze cellular interactions and transform raw multiplex imaging data into actionable biological insights, potentially identifying prognostic indicators for cancer research and therapy development. To support the analysis process a shiny application is provided (Masotti et al. 2023).²¹

8.3 cytomapper: visualization of multiplex images and cell-level information

The cytomapper package is designed to visualize multiplexed read-outs and cell-level information obtained by multiplex imaging technologies (Nils Eling, Nicolas Damond, Tobias Hoch 2020). It offers various functions to view pixel-level information across multiple channels and display expression data for individual cells. Additionally, cytomapper includes features to gate cells based on their expression values, enhancing the analysis of complex data sets. It is compatible with data from various multiplex imaging technologies and requires single-cell read-outs, multi-channel TIFF stacks, and segmentation masks. The cytomapper package is a versatile tool for researchers working with advanced imaging data sets to explore cellular behaviors and properties (Eling et al. 2020).

8.4 SPIAT: analyzing spatial properties of tissues

The SPIAT package, standing for Spatial Image Analysis of Tissues, is among the most comprehensive tools for multiplex image analysis (Anna Trigos 2022). Developed with compatibility for multiplex imaging technologies like CODEX and MIBI, SPIAT facilitates the analysis of spatial data by using X and Y coordinates of cells, their marker intensities, and phenotypes. It features six analysis modules that support a variety of functions including visualization, cell co-localization, distance measurements between cell types, categorization of the immune microenvironment in relation to tumor areas, analysis of cellular neighborhoods and clusters, and quantification of spatial heterogeneity (Yang et al. 2020; Anna Trigos 2022). To use SPIAT, images must be pre-segmented and cells phenotyped, typically using external software like HALO and InForm to prepare the correct input format (Yang et al. 2020). The package provides a shiny application that assists the user in formatting spatial data from the aforementioned sources in a manner that ensures compatibility with the functions of the SPIAT package.²² SPIAT is designed to be user-friendly, making complex spatial analysis accessible to researchers with varying computational skills (Feng et al. 2023).

8.5 Seurat: spatially resolved transcriptomics (SRT)

Spatially resolved transcriptomics (SRT) is a commonly used approach for the quantification of gene expression levels in tissue sections while preserving positional information (Larsson et al. 2023). The Seurat package (Hao et al. 2024) is a package for spatial transcriptomics and multiplexed imaging analysis. It shares some similarities with the SPIAT and spatialTIME packages. For assays with cell segmentation, Seurat facilitates the visualization of individual cell boundaries or centroids, thereby enabling more precise mapping of molecular signals to cells. In contrast to other reviewed packages, Seurat's unique feature is its integration of spatial and molecular data for spatial data analysis. In particular, it enables the joint analysis of spatially-resolved gene expression data alongside traditional single-cell RNA-seq, allowing researchers to map cell types and states within their native tissue context, along with metadata. Notably, Seurat supports the analysis and visualization of spatial omics data at both single-cell and subcellular resolution. Seurat deliberately supports a broad range of spatial technologies, including the Akoya CODEX/Phenocycler™ platform and sequencing-based platforms such as Visium Spatial Gene Expression, 10x Genomics and Slide-seq. To achieve these capabilities, Seurat offers statistical methods to identify genes or features with spatially structured expression patterns, which facilitate the uncovering of region-specific biological processes. Since its first publication in 2015 (Satija et al. 2015) its functionality has expanded to include support for image-based spatial transcriptomics (highly multiplexed imaging technologies). Seurat uses image data (e.g., raw, masked, processed images, 10X Genomics Visium Image).

8.6 spatialTIME: spatial analysis of Vectra immunofluorescence data

The spatialTIME package has been designed for the analysis of immunofluorescence data with the objective of identifying spatial patterns within the TME. The package appears to be designed to work

²¹<https://github.com/nateosher/DIMPLE>, accessed 07/11/2024

²²<https://github.com/TrigosTeam/SPIAT-shiny>, accessed 07/11/2024

with data acquired by the Vectra Polaris™ imaging system.²³ It facilitates the spatial analysis of multiplex immunofluorescence data, enabling spatial characterization and architectural reconstruction. Additionally, the package includes a shiny application, iTIME, which offers a user-friendly point-and-click interface that mirrors many of the capabilities found in spatialTIME (J. H. Creed et al. 2021).²⁴ The package also comes with a detailed vignette to help users get started with its features (J. Creed et al. 2024).

In summary, R offers a range of tools for analyzing multiplex imaging data. However, it is important to note that these packages, except for the cytomapper package, require image preprocessing and use the resulting data frames as input for analysis.

9 Tracing the dance - R packages for analyzing cellular movement dynamics

Cellular migration is essential for various physiological and pathological functions, including development, immune responses, wound healing, and tumor progression (Bise et al. 2011; Yamada and Sixt 2019; Hossian and Mattheolabakis 2020), making it a crucial field in disciplines such as neuroscience, oncology, and regenerative medicine (Kaiser and Bruinink 2004; Hu, Becker, and Willits 2023). To gain insight into these biological processes, researchers can track cell movement by manually tracing their positions in sequential images for 2D coordinates or by incorporating the z coordinate for 3D analysis (Hu, Becker, and Willits 2023). By studying cell migration at multiple levels - from the molecular components and the behavior of individual cells to the dynamics of cell populations - researchers can unravel the complex interactions that influence the movement of cells (Maheshwari and Lauffenburger 1998). Such wide studies are crucial in advancing our understanding of phenomena such as cancer metastasis, which could lead to new therapeutic strategies (Um et al. 2017).

9.1 celltrackR: analyzing motion in two or three dimensions

The celltrackR package is intended for analyzing motion in two or three dimensions, primarily using data from time-lapse microscopy or x-y-(z) coordinates. It is useful in both biological settings for tracking cells and in non-biological contexts for object tracking (Johannes Textor 2024). Additionally, the package provides a web user interface to facilitate the analysis process.²⁵ The package contains standard analytical tools, such as mean square displacement and autocorrelation, as well as algorithms for simulating artificial tracks using various models, such as Brownian motion and the Beauchemin model of lymphocyte migration (Johannes Textor 2024). Furthermore, celltrackR provides a complete pipeline for track analysis, including data management, quality control, and methods for detecting tracking errors, such as track interpolation and drift correction (Wortel et al. 2021). The package is well-documented providing detailed vignettes that guide users through the migration analysis process (Johannes Textor 2024).

10 Mapping the unseen - exploring spatial properties in bioimage data

In this section, we explore the use of R tools for analyzing spatial properties in applications such as transcriptomics. One notable package is the MoleculeExperiment package ("MoleculeExperiment" 2024), which can be used to analyze molecular data within image-based data sets. This package builds upon other popular packages like EBImage, focusing on raster analysis, and terra (Hijmans 2024) for handling geographic information systems (GIS) tasks. Raster or gridded data are spatial data structures that divide regions into rectangles called cells or pixels, storing one or more values. These grids contrast with vector data representing points, lines, and polygons in GIS contexts. Each pixel represents an area on a surface, making color image rasters unique due to their multiple bands containing reflectance values for specific colors or light spectra.

The terra package (formerly known as raster/sp) offers fast operations through optimized back-end C++ code. Users can perform various raster tasks such as creating objects, executing spatial/geometric functions like re-projections and resampling, filtering, and conducting calculations. Functions within the package facilitate extracting essential statistics from entire SpatRaster data sets, including mean values, maximum values, value ranges, or counts of NA cells. In addition to these analytical capabilities, terra provides functionality for visualizing data and interacting with rasters,

²³https://web.archive.org/web/20250125194642/https://www.akoyabio.com/wp-content/uploads/2021/11/Vectra_Polaris_Product_Note_with_MOTiF_Akoya.pdf, accessed 07/14/2025

²⁴<https://fridleylab.shinyapps.io/iTIME/>, accessed 07/11/2024

²⁵<https://github.com/ingewortel/celltrackR>, accessed 07/11/2024

enhancing user experience when working with gridded spatial information. This versatility makes the package an essential tool in analyzing transcriptomic data within image-based data sets using R tools (Hijmans 2020).

11 Numbers game - simplifying scientific image data representation

The R environment offers multiple additional tools for the extraction of information from data, with a particular focus on the extraction of measuring points in scientific diagrams. This task is of particular significance when data is available exclusively in image format, for instance from publications or other sources.

11.1 digitize: use data from published plots or images

The `digitize` package is a well-established and mature tool that simplifies importing data from digital images by providing a user-friendly interface for calibration and point location. It leverages the `readbitmap` package to read various bitmap formats such as BMP, JPEG, PNG, and TIFF. When reading these image files, `digitize` relies on the magic number embedded within each file rather than solely relying on the file extension. For seamless integration with JPEG and PNG images, this package depends on external libraries like '`libjpg`' and '`libpng`' (Poisot 2011). Interestingly, the packages can be used for other purposes as well. For example, Figure 9 demonstrates that the `digitize` package can quantify certain structures in images. This example illustrates how fluorescent objects in an image can be identified by their position and subsequently quantified by their number.

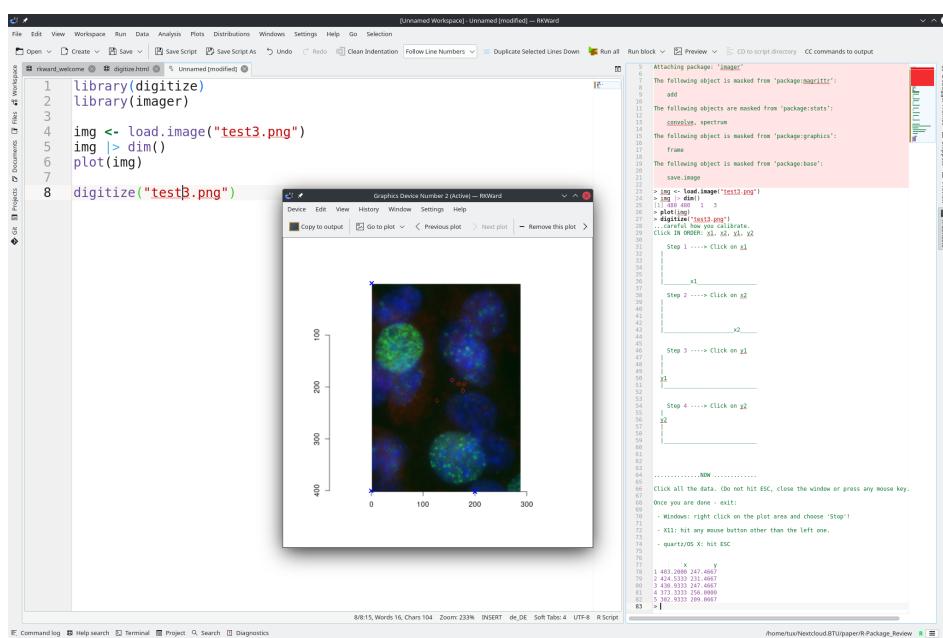


Figure 9: Counting using `digitize`: The figure provided to `digitize`, consists of cells with DNA damage (similar to Rödiger et al. (2018)). The nucleus is colored with DAPI (blue) and the γ H2AX histone, a marker for DNA double strand breaks, is stained with a specific antibody. The `digitize` package is used to interactively extract the coordinates (shown in the console) by using the cursor to define the region of interest (blue cross) and tag the objects within it (red circles). In the screenshot it is displayed how `digitize` is invoked in RKWard (0.7.5z+0.7.6+devel3, Linux, TUXEDO OS 2, (Rödiger et al. 2012)).

11.2 juicr: extraction of numerical data from scientific images

`juicr` is a tool designed to automate the extraction of numerical data from scientific images. It offers users a Tk graphical user interface (GUI) that simplifies point-and-click manual extraction with advanced features such as image zooming, calibration capabilities, and classification options. Additionally, `juicr` provides semi-automated tools for fine-tuning extraction attempts. To ensure optimal performance, this package depends on the `EBImage` package, which must be installed and

loaded prior to utilization. Once data is extracted using `juicr`, users can choose to save their results in various formats including comma-separated values (CSV) files or postscript (EPS) files for easy import into other software. Moreover, extractions can also be saved as fully-embedded and standalone HTML files, that preserve all extraction details, setup configurations, and image modifications. These HTML files provide a means of storing data while ensuring long-term accessibility and replicability for future reference and analysis purposes (Lajeunesse 2021).

11.3 `image2data`: transforming images into data sets

In recent years, the conversion of images into data sets has emerged as an essential tool in various fields such as computer vision, healthcare, and geospatial analysis. The `image2data` R package provides functionality to convert images into data sets (Caron and Dufresne 2022). The primary function `image2data()` takes an image file with extensions like `.png`, `.tiff`, `.jpeg` or `.bmp` as input and converts it into a data set. Each row of the resulting data set represents a pixel (or subject), while columns represent variables such as x-coordinate, y-coordinate, and hex color code. The `image2data()` function offers methods for reducing data sets, yielding results akin to pixelated images with adjustable precision values. Higher precision leads to more data points, while lower precision yields fewer. This example showcases a pixilated representation of a pixel-based image in PNG format, highlighting its unique visual attributes. Users have the ability to customize and modify various elements by adjusting their corresponding hex color codes for precise control over hues, saturation levels, and brightness.

```
# Loading the required packages
library(image2data)
library(data.table)

# Path to the image file
image <- "figures/test3.png"
img <- EBImage::readImage(image)

# Subsampling the image data
beads_subsample <- image2data(
  path = image,
  reduce = .2,
  seed = 42,
  showplot = FALSE
) |> as.data.table() # Converting the result to a data.table

# Display a part of the subsampled data
beads_subsample

#>          x      y      g
#>      <num>    <num>  <char>
#>  1:  0.1022393 -0.9263444 #2F5C61
#>  2: -0.1022393  0.4006978 #121D11
#>  3:  1.2449136 -0.5213380 #121B10
#>  4:  0.4871401 -1.6588028 #151E1C
#>  5: -0.3548305 -1.5381626 #0D1B0D
#>  ---
#> 23151: -1.1486884  1.1159219 #352B5E
#> 23152: -0.6074216  0.1508003 #252E60
#> 23153:  1.4975048  0.5988925 #14180B
#> 23154: -1.3651952  0.2025032 #2A306B
#> 23155:  0.3428023 -0.3231434 #112048

EBImage::display(img)

# Plotting the subsampled data
plot(beads_subsample$x,           # x-coordinates
     beads_subsample$y,           # y-coordinates
     col = beads_subsample$g,     # Color based on hex code extracted by image2data()
     pch = 19,                   # Plotting character (solid circle)
```

```
xlab = "",  
ylab = "")
```

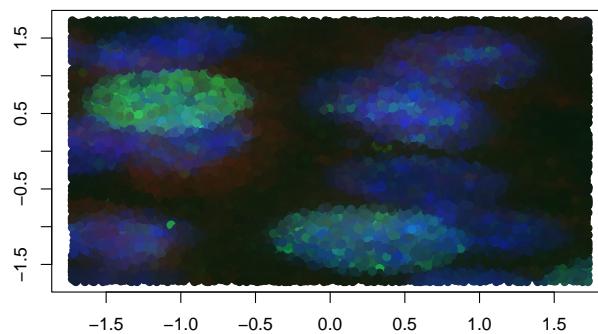
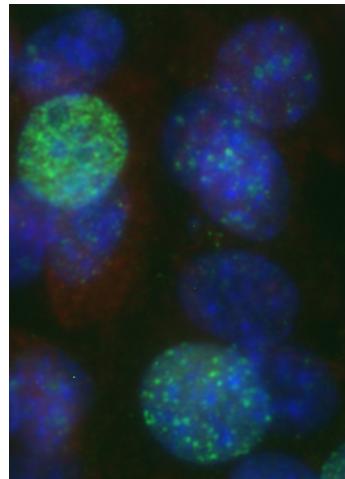


Figure 10: Application Example of the `image2data` Package: The image displays nuclei stained with DAPI (blue) and a quantitative marker for DNA double strand breaks, was labeled with a specific antibody (green). The `image2data` package extracted 20% of the pixels from the original image (top), creating a table with `x | y` coordinates and corresponding hex color codes. This data was then used to reassemble the image using R's base plot (bottom).

12 Engaging insights - interactive approaches to image analysis

The analysis and processing of images to extract useful information can be a challenging endeavor. Consequently, the implementation of interactive approaches accompanied by immediate visual feedback regarding parameter alterations represents a significant aid in simplifying image analysis. Therefore, this section will focus on interactive tools and functions from packages that facilitate the exploration of images and the extraction of useful insights.

12.1 `cytomapper`: a shiny application for hierarchical gating and visualization of multiplex images

The `cytomapper` package, designed for processing multiplex images, includes a shiny application that facilitates the hierarchical gating of cells using specific markers and allows for the visualization of selected cells. The graphical user interface (GUI) of this shiny application is designed to assist in the process of cell labeling. Furthermore, the data from the selected cells can be saved as a `SingleCellExperiment`, thereby enabling various downstream processing methods (Eling et al. 2020; Nils Eling, Nicolas Damond, Tobias Hoch 2020). The `cytomapper` package offers comparable functionality for feature extraction as described in the beginning, providing an algorithm for extracting morphological and intensity features from multiplex images (Nils Eling, Nicolas Damond, Tobias Hoch 2020).

12.2 colocr: interactive ROI selection in image analysis through shiny app

The `colocr` package, which facilitates the exploration of fluorescent microscopic images, features a GUI accessible through a shiny app. This GUI can be invoked locally or accessed online. The process of image analysis frequently necessitates the input of manual labor, particularly in the selection of ROIs. This package streamlines the process of selecting ROIs by semi-automating it, thereby allowing users to review and interactively select one or more ROIs. Moreover, the app offers the option to interactively adjust parameters such as threshold, tolerance, denoising, and hole filling, thereby enhancing user control and precision in image analysis by providing immediate feedback (Ahmed, Lai, and Kim 2019; Ahmed 2020).²⁶

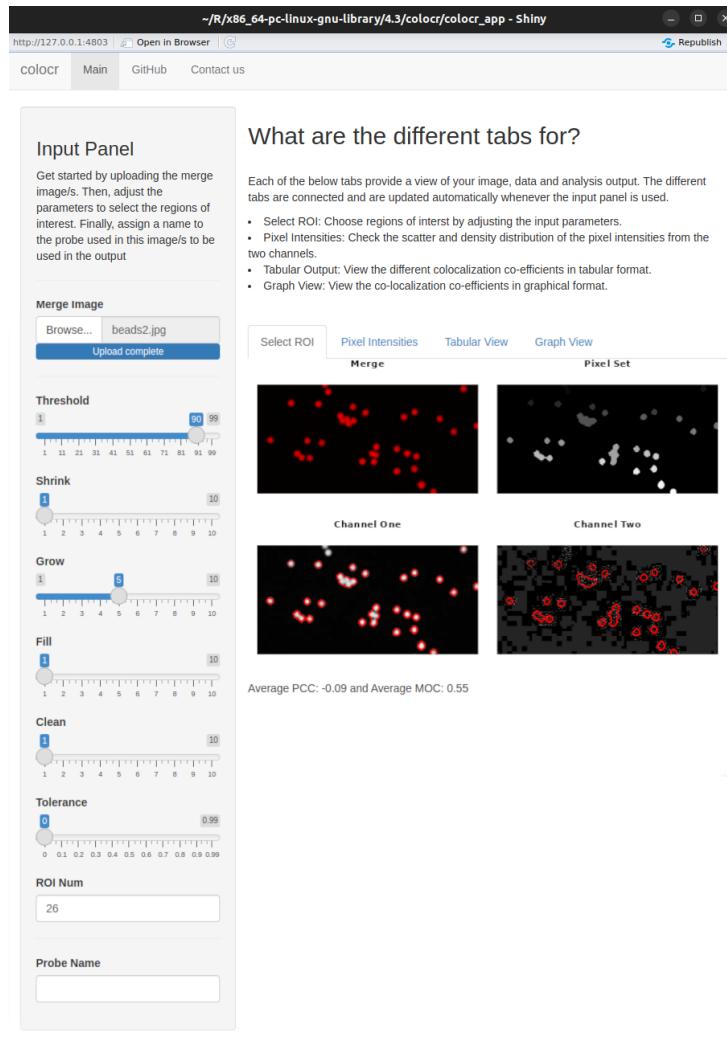


Figure 11: Shiny Application of the `colocr` Package: The figure depicts an interactive image analysis graphical user interface (GUI), invoked locally from the RStudio integrated development environment (IDE). It comprises multiple sliders for real-time parameter adjustments and supports the selection of multiple distinct regions of interest (ROIs). Users can interactively select ROIs and extract characteristics such as pixel intensity. Furthermore, the tool offers functionalities to compute co-localization, providing comprehensive analysis capabilities. Available at: https://mahshaaban.shinyapps.io/colocr_app2/ or run: `colocr::colocr_app()`.

12.3 magick: shiny and Tcl/Tk tools for interactive image exploration

A basic demo version of an interactive web interface for the `magick` R package is available via a shiny app. While it remains a demonstration version and does not encompass all the functionalities of the full package, it is not suitable for in-depth analysis of large-scale imaging data. In contrast, the app provides fundamental tools for image processing, including blurring, imploding, rotating, and

²⁶https://mahshaaban.shinyapps.io/colocr_app2/, accessed 07/11/2024

more. This tool is designed to facilitate basic image processing tasks in an interactive environment.²⁷ Additionally, a distinct package is available that provides the functionality of `magick` in an interactive manner. This package, called `magickGUI`, was developed by Ochi (2023). The interactive features are based on the Tcl/Tk wrapper for R and include functions for thresholding, edge detection, noise reduction, and many more.

12.4 `biopixR`: interactive Tcl/Tk function for feature extraction

In the `biopixR` package, the `tcltk` package — which enables Tcl/Tk integration in R — was employed to create an interactive function. This function initiates the launch of a GUI that streamlines the process of feature extraction by facilitating object detection and enabling users to select between edge detection and thresholding for segmentation. The GUI displays the currently detected edges (when using edge detector) or all detected coordinates (when using threshold) and the object centers within an image. The application includes sliders that allow users to adjust parameters and magnify the image. This interactive function is designed to facilitate the parameter selection process, as the chosen parameters affect the quality of image segmentation (Brauckhoff, Kieffer, and Rödiger 2024).

13 Tailored tools - specialized R packages for image processing

In contrast to the previously mentioned general-purpose tools, some packages have been designed with a specific focus on particular research areas. These specialized tools address the unique challenges encountered in those fields and offer versatile solutions for analyzing the data collected in those domains. While a complete survey of the available packages is outside the scope of this article, a concise overview of the most pertinent packages and their applications will be presented.

13.1 `fslr`: analysis of neuroimage data

The `fslr` package serves as a wrapper for the FSL software, enabling the use of the 'FMRIB' Software Library within the R environment. The FSL software is a widely utilized tool for the analysis and processing of neuroimaging data, including MRI. The package employs the use of NIfTI images to facilitate the execution of processing tasks, thereby introducing capabilities such as brain extraction and tissue segmentation, which were previously unavailable in R (Muschelli et al. 2015; Muschelli 2022).

13.2 `colocr`: co-localization analysis of fluorescence microscopy images

A common application derived from fluorescence microscopy, which is extensively utilized in biological research, is co-localization analysis. This analysis assesses the distribution of signals across different color channels to determine whether the positioning of objects is correlated (Dunn, Kamocka, and McDonald 2011; Ahmed, Lai, and Kim 2019). The objective of this software is to streamline the analysis process by providing tools for loading images, selecting regions of interest, and calculating co-localization statistics (Ahmed, Lai, and Kim 2019; Ahmed 2020). It incorporates methods outlined by Dunn, Kamocka, and McDonald (2011).²⁸

CRAN offers a list of packages tailored to medical image analysis, accompanied by detailed descriptions of their applications. This list can be accessed via the following URL:

<https://cran.r-project.org/web/views/MedicalImaging.html>

Moreover, the Bioconductor repository contains a number of packages focused on single-cell analysis, as detailed by Amezquita et al. (2019). The Bioconductor project is an initiative dedicated to the collaborative development and the use of scalable software for computational biology and bioinformatics. Its objective is to reduce the entry barriers to interdisciplinary research and to improve the remote reproducibility of scientific findings (Gentleman et al. 2004). Other packages identified during the course of our research, though not explored in depth, are acknowledged in the forthcoming summary:

²⁷<https://github.com/jeroen/shinymagick>, accessed 07/11/2024

²⁸<https://github.com/ropensci/colocr>, accessed 07/11/2024

Table 1: Overview of R packages for tailored applications in image processing. This table summarizes key aspects such as general application, repository (Repo) hosting (CRAN, Bioconductor (Bioc), GitLab), linked libraries, and package dependencies. It also includes information on licensing and current status. The current status is divided into the date of first publication on the corresponding repository (*). Active repository status is indicated by a circle, with the date of the latest update (^). Some packages that are no longer maintained are marked as archived (†).

	Application	Repo	based on	License	Status
adimpro by Polzehl and Tabelow (2007)	Adaptive Smoothing	CRAN	Image Magick	GPL (≥ 2)	*2006-10-27 °2023-09-06
phenopix by Filippa et al. (2016)	Vegetation phenology	CRAN	jpeg	GPL-2	*2017-06-16 °2024-01-19
gitter by Wagih and Parts (2014)	Pinned Microbial Cultures	CRAN-archived	EBIImage	LGPL	*2013-06-29 †2020-01-16
TCIApathfinder by Russell et al. (2018)	Cancer Imaging	CRAN	Rnifti	MIT	*2017-08-20 °2019-09-21
SPUTNIK by Inglesi et al. (2018)	Mass Spectrometry Imaging	CRAN	imager	GPL (≥ 3)	*2018-02-19 °2024-04-16
SAFARI by Fernández et al. (2022)	Shape analysis	CRAN	EBIImage	GPL (≥ 3)	*2021-02-25
pavo by Maia et al. (2019)	Spectral and Spatial analysis	CRAN	magick & imager	GPL (≥ 2)	*2012-12-05 °2023-09-24
miet by Combès (2020)	Magnetic Resonance images	gitlab	Rnifti	MIT	*2019-09-06 °2023-12-20
scalpel by Petersen, Simon, and Witten (2017)	Calcium imaging	CRAN	-	GPL (≥ 2)	*2017-03-14 °2021-02-03
ProFit by Robotham et al. (2016)	Galaxy images	CRAN-archived	EBIImage	LGPL-3	*2016-09-29 †2022-08-08
fsbrain by Schäfer and Ecker (2020) & Schaefer (2024)	Neuroimaging	CRAN	magick	MIT	*2019-10-30 °2024-02-03
geomorph by Adams and Otárola-Castillo (2013)	Geometric morphometric shape analysis	CRAN	jpeg	GPL (≥ 3)	*2012-10-26 °2024-03-05
imbibe	Medical images	CRAN	Rnifti	BSD-3-clause	*2020-10-26 °2022-11-09
opencv by Ooms and Wijffels (2024)	edge, body, face detection	CRAN	OpenCV	MIT	*2019-04-01 °2023-10-29
DRIP	jump regression, denoising, deblurring	CRAN	-	GPL (≥ 2)	*2015-09-22 °2024-02-05
imagefluency by Stefan Mayer (2024)	image statistics based on fluency theory	CRAN	magick & Open-ImageR	GPL-3	*2019-09-27 °2024-02-22
mand by Kawaguchi (2021)	Neuroimaging	CRAN	imager	GPL-2 GPL-3	*2020-05-06 °2023-09-12
recolorize by Weller et al. (2024)	Segmentation	CRAN	imager	CC BY 4.0	*2021-12-07
MaxContrastProjection by Jan Sauer (2017)	maximum contrast projection	Bioc	EBIImage	Artistic-2.0	*2017-04-25 †2020-04-28

14 Combining forces - making use of the open-source approach

The majority of the aforementioned packages are designed to encompass all facets of image analysis, including preprocessing, quantification, and visualization. This integration is typically achieved through the utilization of one or more general-purpose packages (Table 1 and 2). The combination of

existing packages or libraries with new code facilitates the development of specialized packages. R, as a package-based language, provides a convenient means of combining these specialized packages to meet the specific needs of the individual user. The following section illustrates the combination of packages to perform statistical analysis on quantified image data.

14.1 **biopixR** and **countfitteR**: quantitative analysis of DNA double strand breaks

DNA double strand breaks (DSBs) represent a particularly severe form of DNA damage, frequently resulting in apoptotic cell death in the absence of repair. The extent of DNA damage can be quantified through immunofluorescence staining, which employs antibodies against the phosphorylated histone protein H2AX (γ H2AX). The staining process results in the formation of γ H2AX foci, which serve as a quantitative representation of the number of DNA DSBs. It has been proposed that the number of DNA DSBs is indicative of the efficacy of an anti-tumor agent, thereby enabling the assessment of individual patient responses to therapies and the evaluation of the general cytotoxic effects of treatments *in vivo*. This enables more precise modulation of therapy according to the patients individual needs (Rödiger et al. 2018; Ruhe et al. 2019; J. Schneider et al. 2019).

In the following example, the **biopixR** package was employed to quantify DNA double-strand breaks, resulting in an output of foci per cell (Figure 12). To achieve this objective, the green fluorescent foci were extracted by applying the `objectDetection()` function to the green color channel of the image (Figure 12A). The result of the foci extraction is illustrated in Figure 12B using the `changePixelColor()` function, whereby each of the distinct foci is highlighted in a different color. The DAPI-stained nuclei were extracted through the application of thresholding on the blue color channel. Subsequently, the resulting data frame was subjected to size filtering in order to eliminate any detected noise. The final quantification of foci per cell was achieved by comparing the coordinates of nuclei and foci in the obtained data frames. This result can then be further analyzed using the **countfitteR** package, which provides an automated evaluation of distribution models for count data' (Michał Burdukiewicz 2019; Chilimoniuk et al. 2021). The resulting distribution is presented in Figure 13.

```
# Load the 'biopixR' package
library(biopixR)

# Import image from specified path
DSB_img <- importImage("figures/tim_242602_c_s3c1+2+3m4.tif")

# Extract the blue color channel representing the nuclei and
# the green color channel representing yH2AX foci
core <- as.cimg(DSB_img[, , , 3])
yH2AX <- as.cimg(DSB_img[, , , 2])

# Process the nuclei: thresholding, labeling, and converting to a data frame
cores <-
  threshold(core) |> label() |> as.data.frame() |> subset(value > 0)

# Calculate the center and size for the nuclei
DT <- as.data.table(cores)
cores_center <-
  DT[, list(mx = mean(x),
            my = mean(y),
            size = length(x)), by = value]

# Filter the nuclei based on size, to discard noise
cores_clean <-
  sizeFilter(cores_center,
             cores,
             lowerlimit = 150,
             upperlimit = Inf)

# Detect objects yH2AX foci in green color channel
DSB <- objectDetection(yH2AX, alpha = 1.1, sigma = 0)

# Function to compare coordinates from two data frames and count matches
compareCoordinates <- function(df1, df2) {
  # Create a single identifier for each coordinate pair
```

```
df1$coord_id <- paste(round(df1$mx), round(df1$my), sep = ",")  
df2$coord_id <- paste(df2$x, df2$y, sep = ",")  
  
# Find matches by checking if coordinates from df2 exist in df1  
matches <- df2$coord_id %in% df1$coord_id  
  
# Convert df2 to a data table and add a column indicating matches  
DT <- data.table(df2)  
DT$DSB <- matches  
  
# Summarize the results  
result <-  
  DT[, list(count = length(which(DSB == TRUE))), by = value]  
  
return(result)  
}  
  
# Compare coordinates between detected DSB centers and cleaned nuclei coordinates  
count <- compareCoordinates(DSB$centers, cores_clean$coordinates)  
  
# Extract the count column for further analysis  
to_analyze <- count[, 2]
```

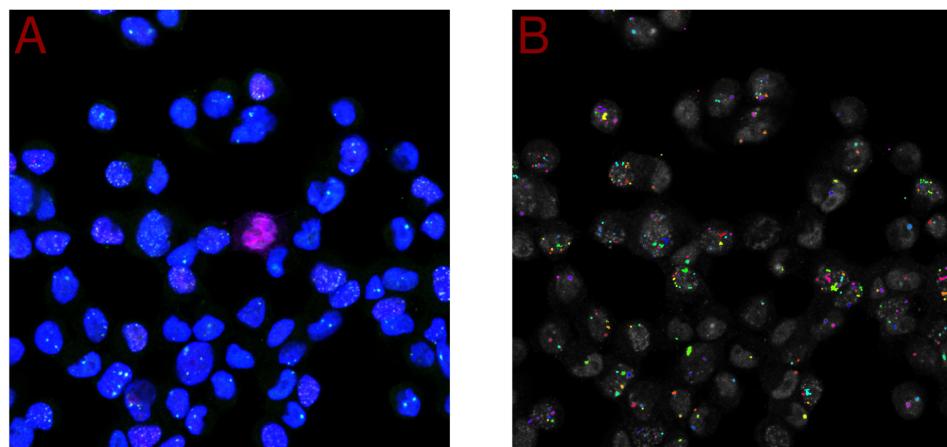


Figure 12: Quantification of DNA Double Strand Breaks: **A)** The image displays cells with nuclei stained using DAPI. The quantitative marker for DNA double strand breaks, γ H2AX, targeted with a specific antibody, is visible as green fluorescent foci. The experimental procedure follows the method described by Rödiger et al. (2018). **B)** The γ H2AX foci are quantified using the biopixR package. The detected foci are highlighted in different colors using the changePixelColor() function.

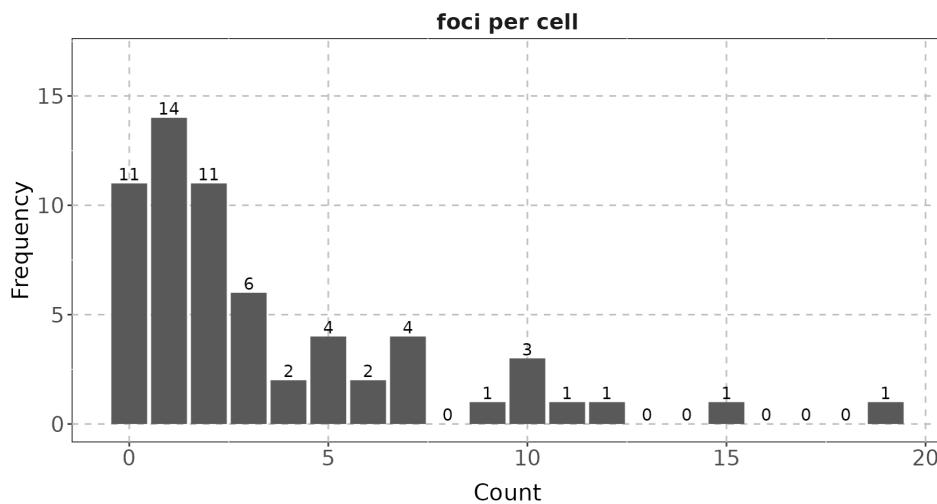


Figure 13: Analyzing Count Data with the countfitteR Package: The data representing the number of foci per cell obtained from the biopixR analysis were imported into the interactive shiny interface of the countfitteR package. This package analyzed the distribution and summarized the results. One outcome is illustrated in this figure, which shows the frequency distribution of a specific count of foci per cell.

15 Exploring the blank spot - z-stack imaging in R

Z-stack imaging refers to the capture of images that possess a third dimension, specifically image depth, which enables the spatial capture of molecules or the reconstruction of the three-dimensional architecture of tissues. One method for achieving z-stacking involves capturing multiple two-dimensional images at uniform intervals over the depth of an object by changing the focal plane. The individual 2D images are then reconstructed to create a 3D model (Trivedi and Mills 2020; Kim et al. 2022).

The only packages currently available in the R programming language for dealing with z-stack imaging are spatialTIME and MaxContrastProjection. However, the spatialTIME package, necessitates preprocessing and is therefore unable to handle the images directly (J. H. Creed et al. 2021). The other package, MaxContrastProjection, has unfortunately been removed from Bioconductor. The package is capable of performing maximum contrast projection, whereby the z-stacks of a 3D image are merged into a 2D image (Jan Sauer 2017). To the best of our knowledge, these are the only packages in R that address the topic of z-stack imaging.

16 Scaling new heights - high throughput analysis in the era of small and big data?

The exponential growth of data, which reached levels of zettabytes (10^{21} bytes) as early as 2012 (Sagiroglu and Sinanc 2013), is accompanied by a significant increase in image generation due to advancements in imaging technologies such as microscopy. High-resolution images produced in a single experiment can result in data sets exceeding terabytes (Peng et al. 2012; Eliceiri et al. 2012). This surge in data generation across various fields has initiated the era of Big Data, which presents considerable challenges in the handling and interpretation of massive data sets (Cui, Schwarz, and Datcu 2015). In automated microscopy, the rapid acquisition of large image volumes facilitates extensive screening processes but complicates the conversion of image stacks into actionable information and discoveries. Resulting in a critical need for analytical pipelines that can efficiently identify regions of interest, compute relevant features, and perform statistical analysis, ensuring reproducibility and reliability (Wollman and Stuurman 2007).

The extraction of quantitative information from images is a common practice, but it is becoming increasingly complex and error-prone when performed manually. This complexity requires the implementation of high-throughput methods capable of autonomously processing multiple images (Olivoto 2022). These developments are crucial not only in specialized fields such as immunohistochemistry, fluorescence *in situ* hybridization (Ollion et al. 2013), drug discovery, and cell biology (Shariff et al. 2010), but also in promoting a data-driven approach to biological research, thereby accelerating tasks and enhancing research productivity (Rittscher 2010).

The R programming language has limitations in handling large data sets. Since R places temporary copies of data in the random access memory (RAM) to access objects, it can lead to memory overload when processing data sets that exceed the available RAM. Additionally, R uses RAM to store generated data, so large lists of imported images can easily overwhelm the RAM. Moreover, R typically executes code on a single thread, not utilizing the full capabilities of the central processing unit (CPU). Several packages address issues such as file-based access and parallel computing, thereby enhancing R's capability to handle big data. One approach is to combine R with the 'Hadoop' library (Prajapati 2013; Oussous et al. 2018). Another effective method for managing big data is the use of the HDF5, which efficiently manages data storage and access, provides multicore reading and writing, and is well-suited for organizing complex data collections. The `cytomapper` package utilizes HDF5 to optimize file management (Nils Eling, Nicolas Damond, Tobias Hoch 2020; Folk et al. 2011; Koranne 2011).

Other packages, such as `pliman`, `biopixR`, and `FIELDimageR`, include features for optimized batch processing, such as parallel processing, by utilizing the `foreach` package for multi-core processing (Olivoto 2022; Brauckhoff, Kieffer, and Rödiger 2024; Matias, Caraza-Harter, and Endelman 2020). However, these packages are not fully optimized for big data. The `biopixR` package simplifies image processing by providing a pipeline that scans entire directories and verifies image uniqueness using Message Digest 5 (MD5) sums. It enables the application of specific filters to batches of images and generates an RMarkdown log file detailing the operations performed. The results are saved in a manageable CSV format, enhancing the efficiency of handling whole image directories (Brauckhoff, Kieffer, and Rödiger 2024).

In conclusion, while R offers a range of options for handling big data, these options are not widely implemented in image processing packages. Consequently, the optimization and creation of workflows capable of handling big data is left to the end-user.

17 Summary

In conclusion, we present a summary of the major R packages previously discussed. This summary provides an overview of the general applications, published repositories, and licensing information associated with these packages. Furthermore, it includes a list of the dependencies or libraries that these packages rely on. The status column indicates both the initial publication date and the date of the most recent update, thereby demonstrating the ongoing commitment to maintaining these packages (Table 2).

Table 2: Summary of key characteristics of major R packages for image processing. The table details general applications, repository (Repo) sources (CRAN, Bioconductor (Bioc), and GitHub), primary package or library dependencies, and licensing information. The status column indicates the date of first publication (*) and the most recent update (°) for each package.

	Application	Repo	based on	License	Status
imager by Barthelmé and Tschumperlé (2019)	general purpose	CRAN	Cimg	LGPL-3	*2015-08-26 °2024-04-26
magick by Ooms (2024b)	general purpose	CRAN	Image Magick	MIT	*2016-07-24 °2024-02-18
EBImage by Pau et al. (2010)	general purpose	Bioc	-	LGPL	*2006-04-27 °2024-05-01
biopixR by Brauckhoff, Kieffer, and Rödiger (2024)	bioimages	CRAN	imager & magick	LGPL (≥ 3)	*2024-03-25 °2024-11-11
pliman by Olivoto (2022)	plant images	CRAN	EBImage	GPL (≥ 3)	*2021-05-15 °2023-10-14
mxnorm by C. Harris, Wrobel, and Vandekar (2022)	multiplex images	CRAN	-	MIT	*2022-02-22 °2023-05-01
DIMPLE by Masotti et al. (2023)	multiplex images	GitHub	-	MIT	*2023-09-07
cytomapper by Eling et al. (2020)	multiplex images	Bioc	EBImage	GPL (≥ 2)	*2020-10-28 °2024-05-01
SPIAT by Yang et al. (2020)	spatial data	Bioc	Spatial Experiment	Artistic-2.0	*2022-11-02 °2024-05-01

	Application	Repo	based on	License	Status
spatialTIME by J. H. Creed et al. (2021)	spatial data	CRAN	-	MIT	*2021-05-14 °2024-03-11
celltrackR by Wortel et al. (2021)	motion analysis	CRAN	-	GPL-2	*2020-03-31 °2024-03-26
FIELDimageR by Matias, Caraza-Harter, and Endelman (2020)	agricultural field trails	GitHub	EBImage	GPL-3	*2019-11-01 °2024-05-03
fslr by Muschelli et al. (2015)	MRI of the brain	CRAN	FMRIB library	GPL-3	*2014-06-13 °2022-08-25
colocR by Ahmed, Lai, and Kim (2019)	fluorescence microscopy	CRAN	imager & magick	GPL-3	*2019-05-31 °2020-05-08
imageseg by Jürgen Niedballa et al. (2022b)	image segmentation	CRAN	magick	MIT	*2021-12-09 °2022-05-29
SimpleITK by Beare, Lowekamp, and Yaniv (2018)	general purpose	GitHub	Simple ITK	Apache 2.0	*2015-11-16 °2020-09-17
pixelclassifier by Real (2024)	image segmentation	CRAN	jpeg & tiff	GPL-3	*2021-10-21 °2023-10-18
OpenImageR	general purpose	CRAN	Rcpp	GPL-3	*2016-07-09 °2023-07-08
RniftyReg	image registration	CRAN	Rcpp & Rnifti	GPL-2	*2010-09-06 °2023-07-18

The packages outlined in Table 2 are examined in terms of their individual dependencies. A minimal number of dependencies is essential for ensuring long-term stability and functionality. The packages are organized according to their dependencies and imports, which were extracted from the DESCRIPTION files to facilitate the identification of similarities between the packages. The relationships between the packages are illustrated in the form of a dendrogram (Figure 14).

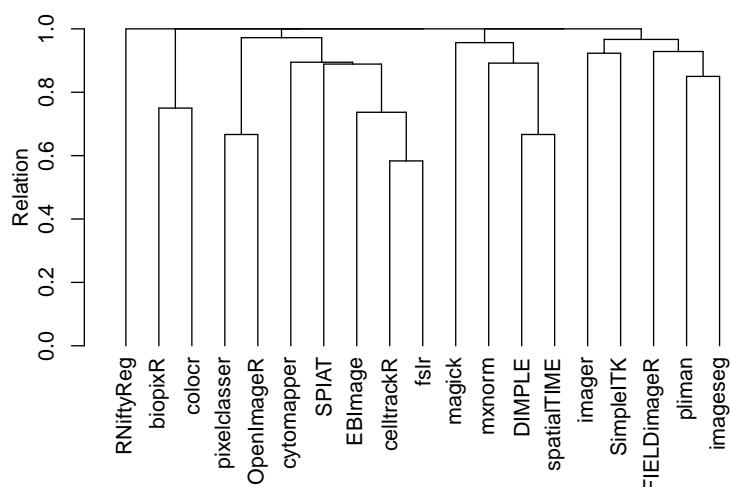


Figure 14: Dendrogram of Hierarchically Clustered Package Dependencies: The dendrogram depicts the outcomes of a hierarchical clustering of various image analysis packages, based on their named dependencies and imports, as extracted from their respective DESCRIPTION files. Each branch represents a distinct package, and the proximity between branches reflects the degree of similarity in their dependencies and imports. The required distance matrix was calculated using the binary method, also known as Jaccard distance. To perform the hierarchical clustering, the complete linkage clustering method was employed (R Core Team 2023).

18 Conclusion

The Tables 1 and 2 highlight an array of R packages employed within bioimage informatics. These tools cater to diverse applications such as adaptive smoothing, vegetation phenology analysis, microbial culture imaging, cancer imaging, mass spectrometry imaging, shape analysis, spectral and spatial analysis, magnetic resonance image processing, calcium imaging, galaxy image analysis, neuroimaging, geometric morphometric shape analysis, medical image processing, edge detection, body and face recognition, jump regression, denoising, and deblurring.

Many of these packages rely on common image processing libraries such as ‘ImageMagick’ and ‘CImg’ or specialized libraries like ‘RNifti’ for neuroimaging data and OpenCV for computer vision tasks. Some notable examples include *adimpro*, *gitter*, *SAFARI*, *pavo*, *rental*, *scalpel*, *ProFit*, and *fsbrain*.

The majority of these packages are hosted on CRAN, which serves as the primary repository for R packages. Notably, one package, *rental*, is hosted on GitLab, indicating that some packages may also be developed and distributed through alternative platforms. R is an open-source, free, and cross-platform programming language that extends these values to its packages (R Core Team 2023). The CRAN Repository Policy states that package authors “should make all reasonable efforts to provide cross-platform portable code,” typically requiring packages to run on at least two major R platforms.²⁹ Similarly, the standard tests employed by Bioconductor encompass evaluations on all major platforms, including Linux, macOS, and Windows.³⁰ Thus, it can be concluded that the majority of packages in these repositories are compatible across multiple platforms.

The most commonly used license in this domain is the GNU General Public License (GPL), particularly versions 2 and 3. Other licenses employed include the Lesser GNU General Public License (LGPL), MIT, Apache License 2.0, and others. The prevalence of open-source licenses reflects the collaborative nature of R package development. It’s essential to ensure compatibility when combining code from different packages with varying licenses; otherwise, legal considerations might arise.

As previously outlined, the most fundamental image processing packages in R are *imager*, *magick*, *EBImage*, *OpenImageR*, and *SimpleITK*. Primarily, *imager*, *magick*, and *EBImage* form the foundation for the majority of the specialized packages reviewed. These packages support various formats, with JPEG and PNG being the most common and supported by all five packages. BMP and TIFF are also widely supported, while PDF and SVG formats are exclusively supported by *magick*.

Table 3: Supported File Formats by Main Image Processing Packages

	imager	magick	EBImage	OpenImageR	SimpleITK
JPEG	+	+	+	+	+
PNG	+	+	+	+	+
BMP	+	+	-	-	+
TIFF	-	+	+	+	+
PDF	-	+	-	-	-
SVG	-	+	-	-	-

The ongoing development of new code by the R community significantly enhances the capabilities of image analysis, fostering both growth and adaptability within the community. This ensures that R remains well-equipped to address emerging challenges effectively. The result is a diverse range of image processing packages, including versatile general-purpose tools and specialized pipelines designed for intricate analyses of biological images. This extensive array of tools in R not only demonstrates the versatility and applicability of these packages across different scientific disciplines but also solidifies R’s position as an invaluable resource for researchers interested in leveraging image analysis to uncover novel insights. This review provides a concise overview of the current landscape of image processing packages available in R, emphasizing the pivotal role these tools play in advancing scientific research and discovery. The comprehensive toolkit, R, empowers researchers to drive forward innovations and enrich the scientific community. Finally, it is noteworthy that 92 % of the 38 discovered packages are active in their respective repositories and thus considered up to date. Furthermore, 66 % of these packages have been actively maintained with updates in the past 1.5 years. Among the identified packages, 14 provide users with GUIs or interactive functions. These packages include: *FIELDimageR*, *cytomapper*, *colocr*, *biopixR*, *EBImage*, *magick*, *imager*, *pavo*, *pliman*, *imagefluency*, *geomorph*, *fsbrain*, *scalpel*, and *adimpro*. The majority of the 38 packages identified

²⁹<https://cran.r-project.org/web/packages/policies.html>, accessed 06/10/2024

³⁰<https://contributions.bioconductor.org/bioconductor-package-submissions.html>, accessed 06/10/2024

during the research can be considered autonomous, offering all the necessary features for extensive image data analysis, including image import, processing, and visualization. However, some packages related to multiplex imaging necessitate preprocessing, rendering them unable to provide a complete analysis within the R environment.

All mentioned packages are open source and available either on CRAN, Bioconductor or GitHub.

Predicting the future is challenging, yet here we provide some opinions on trends in bioimage informatics, which ultimately will also be seen in R. Publications and conferences in the fields of image processing and computer vision show that advances are driven by artificial intelligence (AI), deep learning (particularly Convolutional Neural Networks (CNNs), Large Language Models (LLMs) and Vision Transformer models (VTs)), and data visualization (Ye et al. 2024; Belcher et al. 2023; Rabbani et al. 2021; Hameed, Abdulhussain, and Mahmmod 2021; Velden et al. 2022). One example of deep learning is `imageseg`, which is using a CNN (U-Net and U-Net++ architectures) for general purpose image segmentation (Jürgen Niedballa et al. 2022a). Another development is the deeper integration of R with advanced deep learning frameworks, which will enable users to build and deploy models, with applications like image classification, segmentation and object detection. An example for such integration is `e11mer`, which makes various LLMs accessible from R for output streaming, tool calling and structured data extraction.

The question arises: Is AI merely a buzzword, or is it here to stay? Given that AI is grounded in science and we already see applications in R, the latter is more probable. Consequently, R bioimage packages will be developed that combine image data with other multimodal data types, such as text and sensor data. Generative AI and advanced visualization techniques are also one topic due to the availability of generative models like diffusion models and Generative Adversarial Networks (GANs). These technologies open new possibilities for image augmentation and enhanced data visualization. It is important that such technologies stick to one of R's strengths, which is explainability, in particular on transparent, understandable and explainable AI (xAI).

18.1 Session Info

```
#> R version 4.3.3 (2024-02-29)
#> Platform: x86_64-pc-linux-gnu (64-bit)
#> Running under: Ubuntu 24.04.2 LTS
#>
#> Matrix products: default
#> BLAS:   /usr/lib/x86_64-linux-gnublas/libblas.so.3.12.0
#> LAPACK: /usr/lib/x86_64-linux-gnulapack/liblapack.so.3.12.0
#>
#> locale:
#> [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
#> [3] LC_TIME=de_DE.UTF-8      LC_COLLATE=en_US.UTF-8
#> [5] LC_MONETARY=de_DE.UTF-8    LC_MESSAGES=en_US.UTF-8
#> [7] LC_PAPER=de_DE.UTF-8      LC_NAME=C
#> [9] LC_ADDRESS=C            LC_TELEPHONE=C
#> [11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C
#>
#> time zone: Europe/Berlin
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats4     stats      graphics  grDevices  utils      datasets  methods
#> [8] base
#>
#> other attached packages:
#> [1] RNiftyReg_2.8.4          OpenImageR_1.3.0
#> [3] pixelclasser_1.1.1       SimpleITK_2.4.0
#> [5] imageseg_0.5.0           colocr_0.1.1
#> [7] fslr_2.25.3             neurobase_1.32.4
#> [9] oro.nifti_0.11.4         celltrackR_1.2.1
#> [11] spatialTIME_1.3.4-5     SPIAT_1.4.2
#> [13] SpatialExperiment_1.12.0 cytomapper_1.14.0
#> [15] SingleCellExperiment_1.24.0 SummarizedExperiment_1.32.0
#> [17] Biobase_2.62.0          GenomicRanges_1.54.1
#> [19] GenomeInfoDb_1.38.8     IRanges_2.36.0
#> [21] S4Vectors_0.40.2        BiocGenerics_0.48.1
#> [23] MatrixGenerics_1.14.0   matrixStats_1.4.1
#> [25] DIMPLE_1.0.0           mxnorm_1.0.3
#> [27] kableExtra_1.4.0        data.table_1.16.2
#> [29] image2data_1.0.1       mapview_2.11.2
#> [31] leafsync_0.1.0          sf_1.0-19
#> [33] terra_1.7-83           FIELDimageR.Extra_0.0.5
#> [35] nlme_3.1-164            FIELDimageR_0.6.2
#> [37] pliman_3.0.0           EBImage_4.44.0
#> [39] biopixR_1.2.0          magick_2.8.5
#> [41] imager_1.0.2           magrittr_2.0.3
#> [43] knitr_1.49              cluster_2.1.6
#>
#> loaded via a namespace (and not attached):
#> [1] spatstat.sparse_3.1-0   fs_1.6.5          bitops_1.0-9
#> [4] lubridate_1.9.3        httr_1.4.7          RColorBrewer_1.1-3
#> [7] tools_4.3.3            R6_2.5.1          HDF5Array_1.30.1
#> [10] rhdf5filters_1.14.1   withr_3.0.2        sp_2.1-4
#> [13] readbitmap_0.1.5     gridExtra_2.3     leaflet_2.2.2
#> [16] leafem_0.2.3          cli_3.6.3          spatstat.explore_3.3-3
#> [19] nnls_1.6               mvtnorm_1.3-2     spatstat.data_3.1-4
#> [22] proxy_0.4-27          systemfonts_1.1.0 R.utils_2.12.3
#> [25] svglite_2.1.3         parallelly_1.39.0 maps_3.4.2.1
#> [28] readxl_1.4.3          rstudioapi_0.17.1 generics_0.1.3
#> [31] tensorflow_2.16.0     spatstat.random_3.3-2 crosstalk_1.2.1
#> [34] dplyr_1.1.4            Matrix_1.6-5       ggbeeswarm_0.7.2
#> [37] DescTools_0.99.58      abind_1.4-8        R.methodsS3_1.8.2
#> [40] lifecycle_1.0.4        whisker_0.4.1     yaml_2.3.10
```

```
#> [43] rhdf5_2.46.1           SparseArray_1.2.4      recipes_1.1.0
#> [46] grid_4.3.3            promises_1.3.0        shinydashboard_0.7.2
#> [49] crayon_1.5.3          lattice_0.22-5       haven_2.5.4
#> [52] zeallot_0.1.0         pillar_1.10.1        tcltk_4.3.3
#> [55] rjson_0.2.23          boot_1.3-30          gld_2.6.6
#> [58] future.apply_1.11.3   codetools_0.2-19     glue_1.8.0
#> [61] spatstat.univar_3.1-1 rjtools_1.0.13       yesno_0.1.3
#> [64] vctrs_0.6.5           png_0.1-8           spam_2.11-0
#> [67] cellranger_1.1.0     gtable_0.3.6        gower_1.0.1
#> [70] xfun_0.49             exactextractr_0.10.0 S4Arrays_1.2.1
#> [73] mime_0.12              iterators_1.0.14    survival_3.5-8
#> [76] timeDate_4041.110    lava_1.8.0          hardhat_1.4.0
#> [79] units_0.8-5           satellite_1.0.5     fields_16.3
#> [82] ipred_0.9-15          vigor_0.4.7         hunspell_3.0.5
#> [85] svgPanZoom_0.3.4      colorspace_2.1-1     KernSmooth_2.23-22
#> [88] rpart_4.1.23          nnet_7.3-19         DBI_1.2.3
#> [91] raster_3.6-30         compiler_4.3.3      Exact_3.3
#> [94] tidyselect_1.2.1      DelayedArray_0.28.0 expm_1.0-0
#> [97] xml2_1.3.6            scales_1.3.0        bookdown_0.41
#> [100] keras_2.15.0         tfruns_1.5.3        classInt_0.4-10
#> [103] fuzzyjoin_0.1.6     stringr_1.5.1       goftest_1.2-3
#> [106] tiff_0.1-12          spatstat.utils_3.1-1 digest_0.6.37
#> [109] fftwtools_0.9-11    XVector_0.42.0      minqa_1.2.8
#> [112] rmarkdown_2.29       pkgconfig_2.0.3     RNifti_1.7.0
#> [115] htmltools_0.5.8.1   lme4_1.1-35.5       jpeg_0.1-10
#> [118] base64enc_0.1-3     htmlwidgets_1.6.4    fastmap_1.2.0
#> [121] rlang_1.1.5          BiocParallel_1.36.0 shiny_1.9.1
#> [124] jsonlite_1.8.9      RCurl_1.98-1.16    R.oo_1.27.0
#> [127] ModelMetrics_1.2.2.2 Rhdf5lib_1.24.2     GenomeInfoDbData_1.2.11
#> [130] dotCall64_1.2         viridis_0.6.5       munsell_0.5.1
#> [133] Rcpp_1.0.13-1        pROC_1.18.5        reticulate_1.40.0
#> [136] stringi_1.8.4       MASS_7.3-60.0.1     rootSolve_1.8.2.4
#> [139] zlibbioc_1.48.2      listenv_0.9.1       plyr_1.8.9
#> [142] parallel_4.3.3       deldir_2.0-4        forcats_1.0.0
#> [145] lmom_3.2            tensor_1.5          stars_0.6-7
#> [148] splines_4.3.3       igraph_2.1.1        hms_1.1.3
#> [151] locfit_1.5-9.10     evaluate_1.0.1      spatstat.geom_3.3-4
#> [154] reshape2_1.4.4       nloptr_2.1.1        BiocManager_1.30.25
#> [157] bmp_0.3               tidyR_1.3.1         foreach_1.5.2
#> [160] httpuv_1.6.15       future_1.34.0       purrr_1.0.2
#> [163] polyclip_1.10-7     xtable_1.8-4        ggplot2_3.5.1
#> [166] lwgeom_0.2-14        viridisLite_0.4.2    e1071_1.7-16
#> [169] later_1.3.2          beeswarm_0.4.0      class_7.3-22
#> [172] tibble_3.2.1         globals_0.16.3      mapedit_0.6.0
#> [175] timechange_0.3.0
```

19 Funding

This review was partially funded by the project Rubin: NeuroMiR (03RU1U051A, federal ministry of education and research, Germany).

20 Conflict of interest

The authors declare no conflict of interest.

21 Acknowledgements

We would like to express our gratitude to Dr. Coline Kieffer for providing the microbead images used in this review. We thank Robert M Flight at codeberg.org for reading and improving the manuscript.

References

- Adams, Dean C., and Erik Otárola-Castillo. 2013. "Geomorph: An R Package for the Collection and Analysis of Geometric Morphometric Shape Data." Edited by Emmanuel Paradis. *Methods in Ecology and Evolution* 4 (4): 393–99. <https://doi.org/10.1111/2041-210x.12035>.
- Aherne, Olivia, Martina Mørch, Roberto Ortiz, Oonagh Shannon, and Julia R. Davies. 2024. "A Novel Multiplex Fluorescent-Labeling Method for the Visualization of Mixed-Species Biofilms in Vitro." Edited by Aude A. Ferran. *Microbiology Spectrum*, May. <https://doi.org/10.1128/spectrum.00253-24>.
- Ahmed, Mahmoud. 2020. "Colocr: Conduct Co-Localization Analysis of Fluorescence Microscopy Images." *Comprehensive R Archive Network*. <https://doi.org/10.32614/CRAN.package.colocr>.
- Ahmed, Mahmoud, Trang Huyen Lai, and Deok Ryong Kim. 2019. "Colocr: An R Package for Conducting Co-Localization Analysis on Fluorescence Microscopy Images." *PeerJ* 7 (July): e7255. <https://doi.org/10.7717/peerj.7255>.
- Amezquita, Robert A., Aaron T. L. Lun, Etienne Becht, Vince J. Carey, Lindsay N. Carpp, Ludwig Geistlinger, Federico Marini, et al. 2019. "Orchestrating Single-Cell Analysis with Bioconductor." *Nature Methods* 17 (2): 137–45. <https://doi.org/10.1038/s41592-019-0654-x>.
- Andrzej Oleś, Gregoire Pau. 2017. "EBIImage." Bioconductor. <https://doi.org/10.18129/B9.BIOC.EBIIMAGE>.
- Andrzej Oleś, John Lee. 2023. "RBioFormats." Bioconductor. <https://doi.org/10.18129/B9.BIOC.RBIOFORMATS>.
- Angelo, Michael, Sean C Bendall, Rachel Finck, Matthew B Hale, Chuck Hitzman, Alexander D Borowsky, Richard M Levenson, et al. 2014. "Multiplexed Ion Beam Imaging of Human Breast Tumors." *Nature Medicine* 20 (4): 436–42. <https://doi.org/10.1038/nm.3488>.
- Anna Trigos, Tianpei Yang, Yuzhou Feng. 2022. "SPIAT." Bioconductor. <https://doi.org/10.18129/B9.BIOC.SPIAT>.
- Austenfeld, Marcel, and Wolfram Beyschlag. 2012. "A Graphical User Interface for R in a Rich Client Platform for Ecological Modeling." *Journal of Statistical Software* 49 (June): 1–19. <https://doi.org/10.18637/jss.v049.i04>.
- Barthélémy, Simon, and David Tschumperlé. 2019. "Imager: An R Package for Image Processing Based on Clmg." *Journal of Open Source Software* 4 (38): 1012. <https://doi.org/10.21105/joss.01012>.
- Beare, Richard, Bradley LoweKamp, and Ziv Yaniv. 2018. "Image Segmentation, Registration and Characterization in R with SimpleITK." *Journal of Statistical Software* 86 (8). <https://doi.org/10.18637/jss.v086.i08>.
- Behura, Aradhana. 2021. "The Cluster Analysis and Feature Selection: Perspective of Machine Learning and Image Processing." *Data Analytics in Bioinformatics*. Wiley. <https://doi.org/10.1002/9781119785620.ch10>.
- Belcher, Byron T., Eliana H. Bower, Benjamin Burford, Maria Rosa Celis, Ashkaan K. Fahimipour, Isabela L. Guevara, Kakani Katija, et al. 2023. "Demystifying Image-Based Machine Learning: A Practical Guide to Automated Analysis of Field Imagery Using Modern Machine Learning Tools." *Frontiers in Marine Science* 10 (June). <https://doi.org/10.3389/fmars.2023.1157370>.
- Besson, Sébastien, Roger Leigh, Melissa Linkert, Chris Allan, Jean-Marie Burel, Mark Carroll, David Gault, et al. 2019. "Bringing Open Data to Whole Slide Imaging." In *Digital Pathology*, 3–10. Springer International Publishing. https://doi.org/10.1007/978-3-030-23937-4_1.
- Bezdek, James C., Robert Ehrlich, and William Full. 1984. "FCM: The Fuzzy c-Means Clustering Algorithm." *Computers & Geosciences* 10 (2–3): 191–203. [https://doi.org/10.1016/0098-3004\(84\)90020-7](https://doi.org/10.1016/0098-3004(84)90020-7).
- Bise, R., T. Kanade, Zhaozheng Yin, and Seung-il Huh. 2011. "Automatic Cell Tracking Applied to Analysis of Cell Migration in Wound Healing Assay." In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. <https://doi.org/10.1109/emb.2011.6091525>.
- Boom, Bastiaan J., Phoenix X. Huang, Jiyin He, and Robert B. Fisher. 2012. *Supporting Ground-Truth Annotation of Image Datasets Using Clustering: International Conference on Pattern Recognition*. Piscataway, NJ: IEEE. <https://ieeexplore.ieee.org/abstract/document/6460437>.
- Brauckhoff, Tim, Coline Kieffer, and Stefan Rödiger. 2024. "biopixR: Extracting Insights from Biological Images." *Journal of Open Source Software* 9 (102): 7074. <https://doi.org/10.21105/joss.07074>.
- Burdukiewicz, Michał. 2019. "countfitteR: Comprehensive Automatized Evaluation of Distribution Models for Count Data." CRAN: Contributed Packages. The R Foundation. <https://doi.org/10.32614/cran.package.countfitter>.
- Burdukiewicz, Michał, Andrej-Nikolai Spiess, Dominik Rafacz, Konstantin Blagodatskikh, and Stefan Rödiger. 2022. "PCRedux: A Quantitative PCR Machine Learning Toolkit." *Journal of Open Source Software* 7 (76): 4407. <https://doi.org/10.21105/joss.04407>.
- Caicedo, Juan C., Sam Cooper, Florian Heigwer, Scott Warchal, Peng Qiu, Csaba Molnar, Aliaksei S Vasilevich, et al. 2017. "Data-Analysis Strategies for Image-Based Cell Profiling." *Nature Methods*

- 14 (9): 849–63. <https://doi.org/10.1038/nmeth.4397>.
- Caron, Pier-Olivier, and Alexandre Dufresne. 2022. “Image2data: An R Package to Turn Images in Data Sets.” *The Quantitative Methods for Psychology* 18 (2): 186–95. <https://doi.org/10.20982/tqmp.18.2.p186>.
- Chessel, Anatole. 2017. “An Overview of Data Science Uses in Bioimage Informatics.” *Methods* 115 (February): 110–18. <https://doi.org/10.1016/j.ymeth.2016.12.014>.
- Chilimoniuk, Jarosław, Anna Erol, Stefan Rödiger, and Michał Burdukiewicz. 2024. “Challenges and Opportunities in Processing NanoString nCounter Data.” *Computational and Structural Biotechnology Journal* 23 (December): 1951–58. <https://doi.org/10.1016/j.csbj.2024.04.061>.
- Chilimoniuk, Jarosław, Alicja Gosiewska, Jadwiga Słowik, Romano Weiss, P. Markus Deckert, Stefan Rödiger, and Michał Burdukiewicz. 2021. “countfitteR: Efficient Selection of Count Distributions to Assess DNA Damage.” *Annals of Translational Medicine* 9 (7): 528–28. <https://doi.org/10.21037/atm-20-6363>.
- Cho, Wonjin, Sehun Kim, and Young-Gyun Park. 2023. “Towards Multiplexed Immunofluorescence of 3D Tissues.” *Molecular Brain* 16 (1). <https://doi.org/10.1186/s13041-023-01027-9>.
- Clayden, Jonathan D., Michael Dayan, and Chris A. Clark. 2013. “Principal Networks.” Edited by Pedro Antonio Valdes-Sosa. *PLoS ONE* 8 (4): e60997. <https://doi.org/10.1371/journal.pone.0060997>.
- Clayden, Jon, Marc Modat, Benoit Presles, Thanasis Anthopoulos, and Pankaj Daga. 2023. “RNiftyReg: Image Registration Using the ‘NiftyReg’ Library.” *Comprehensive R Archive Network*. <https://doi.org/10.32614/CRAN.package.RNiftyReg>.
- Combès, Benoit. 2020. “Miet: An R Package for Region of Interest Analysis from Magnetic Reasonance Images.” *Journal of Open Source Software* 5 (45): 1862. <https://doi.org/10.21105/joss.01862>.
- Creed, Jordan H, Christopher M Wilson, Alex C Soupir, Christelle M Colin-Leitzinger, Gregory J Kimmel, Oscar E Ospina, Nicholas H Chakiryan, et al. 2021. “spatialTIME and iTIME: R Package and Shiny Application for Visualization and Analysis of Immunofluorescence Data.” Edited by Anthony Mathelier. *Bioinformatics* 37 (23): 4584–86. <https://doi.org/10.1093/bioinformatics/btab757>.
- Creed, Jordan, Ram Thapa, Christopher Wilson, Alex Soupir, Oscar Ospina, Julia Wrobel, Brooke Fridley, and Fridley Lab. 2024. “spatialTIME: Spatial Analysis of Vectra Immunofluorescent Data.” *Comprehensive R Archive Network*. <https://doi.org/10.32614/CRAN.package.spatialTIME>.
- Cui, Shiyong, Gottfried Schwarz, and Mihai Datcu. 2015. “Remote Sensing Image Classification: No Features, No Clustering.” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 8 (11): 5158–70. <https://doi.org/10.1109/jstars.2015.2495267>.
- Diamond, Nicolas, Stefanie Engler, Vito R. T. Zanotelli, Denis Schapiro, Clive H. Wasserfall, Irina Kusmartseva, Harry S. Nick, et al. 2019. “A Map of Human Type 1 Diabetes Progression by Imaging Mass Cytometry.” *Cell Metabolism* 29 (3): 755–768.e5. <https://doi.org/10.1016/j.cmet.2018.11.014>.
- Dunn, Kenneth W., Małgorzata M. Kamocka, and John H. McDonald. 2011. “A Practical Guide to Evaluating Colocalization in Biological Microscopy.” *American Journal of Physiology-Cell Physiology* 300 (4): C723–42. <https://doi.org/10.1152/ajpcell.00462.2010>.
- Einhaus, Jakob, Alexander Rochwarger, Sven Mattern, Brice Gaudillièvre, and Christian M. Schürch. 2023. “High-Multiplex Tissue Imaging in Routine Pathology—Are We There Yet?” *Virchows Archiv* 482 (5): 801–12. <https://doi.org/10.1007/s00428-023-03509-6>.
- Eliceiri, Kevin W, Michael R Berthold, Ilya G Goldberg, Luis Ibáñez, B S Manjunath, Maryann E Martone, Robert F Murphy, et al. 2012. “Biological Imaging Software Tools.” *Nature Methods* 9 (7): 697–710. <https://doi.org/10.1038/nmeth.2084>.
- Eling, Nils, Nicolas Damond, Tobias Hoch, and Bernd Bodenmiller. 2020. “Cytomapper: An r/Bioconductor Package for Visualization of Highly Multiplexed Imaging Data.” Edited by Zhiyong Lu. *Bioinformatics* 36 (24): 5706–8. <https://doi.org/10.1093/bioinformatics/btaa1061>.
- Ester, Martin, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.” In *Kdd*, 96:226–31. 34. <https://dl.acm.org/doi/10.5555/3001460.3001507>.
- Feng, Yuzhou, Tianpei Yang, John Zhu, Mabel Li, Maria Doyle, Volkan Ozcoban, Greg T. Bass, et al. 2023. “Spatial Analysis with SPIAT and spaSim to Characterize and Simulate Tissue Microenvironments.” *Nature Communications* 14 (1). <https://doi.org/10.1038/s41467-023-37822-0>.
- Fernández, Esteban, Shengjie Yang, Sy Han Chiou, Chul Moon, Cong Zhang, Bo Yao, Guanghua Xiao, and Qiwei Li. 2022. “SAFARI: Shape Analysis for AI-Segmented Images.” *BMC Medical Imaging* 22 (1). <https://doi.org/10.1186/s12880-022-00849-8>.
- Filippa, Gianluca, Edoardo Cremonese, Mirco Migliavacca, Marta Galvagno, Matthias Forkel, Lisa Wingate, Enrico Tomelleri, Umberto Morra di Cella, and Andrew D. Richardson. 2016. “Phenopix: A R Package for Image-Based Vegetation Phenology.” *Agricultural and Forest Meteorology* 220 (April): 141–50. <https://doi.org/10.1016/j.agrformet.2016.01.006>.
- Folk, Mike, Gerd Heber, Quincey Koziol, Elena Pourmal, and Dana Robinson. 2011. “An Overview of

- the HDF5 Technology Suite and Its Applications." In *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*. EDBT/ICDT '11. ACM. <https://doi.org/10.1145/1966895.1966900>.
- Geithe, Christiane, Bo Zeng, Carsten Schmidt, Franziska Dinter, Dirk Roggenbuck, Werner Lehmann, Gregory Dame, Peter Schierack, Katja Hanack, and Stefan Rödiger. 2024. "A Multiplex Microchamber Diffusion Assay for the Antibody-Based Detection of microRNAs on Randomly Ordered Microbeads." *Biosensors and Bioelectronics: X* 18 (June): 100484. <https://doi.org/10.1016/j.biosx.2024.100484>.
- Gentleman, Robert C, Vincent J Carey, Douglas M Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, et al. 2004. "Bioconductor: Open Software Development for Computational Biology and Bioinformatics." *Genome Biology* 5 (10): R80. <https://doi.org/10.1186/gb-2004-5-10-r80>.
- Gerdes, Michael J., Christopher J. Sevinsky, Anup Sood, Sudeshna Adak, Musodiq O. Bello, Alexander Bordwell, Ali Can, et al. 2013. "Highly Multiplexed Single-Cell Analysis of Formalin-Fixed, Paraffin-Embedded Cancer Tissue." *Proceedings of the National Academy of Sciences* 110 (29): 11982–87. <https://doi.org/10.1073/pnas.1300136110>.
- Ghosh, Swarnendu, Nibaran Das, Ishita Das, and Ujjwal Maulik. 2019. "Understanding Deep Learning Techniques for Image Segmentation." *ACM Computing Surveys* 52 (4): 1–35. <https://doi.org/10.1145/3329784>.
- Giorgi, Federico M., Carmine Ceraolo, and Daniele Mercatelli. 2022. "The R Language: An Engine for Bioinformatics and Data Science." *Life* 12 (5): 648. <https://doi.org/10.3390/life12050648>.
- Goldberg, Ilya G, Chris Allan, Jean-Marie Burel, Doug Creager, Andrea Falconi, Harry Hochheiser, Josiah Johnston, Jeff Mellen, Peter K Sorger, and Jason R Swedlow. 2005. "The Open Microscopy Environment (OME) Data Model and XML File: Open Tools for Informatics and Quantitative Analysis in Biological Imaging." *Genome Biology* 6 (5). <https://doi.org/10.1186/gb-2005-6-5-r47>.
- Goltsev, Yury, Nikolay Samusik, Julia Kennedy-Darling, Salil Bhate, Matthew Hale, Gustavo Vazquez, Sarah Black, and Garry P. Nolan. 2018. "Deep Profiling of Mouse Splenic Architecture with CODEX Multiplexed Imaging." *Cell* 174 (4): 968–981.e15. <https://doi.org/10.1016/j.cell.2018.07.010>.
- Haase, Robert, Elnaz Fazeli, David Legland, Michael Doube, Siân Culley, Ilya Belevich, Eija Jokitalo, Martin Schorb, Anna Klemm, and Christian Tischer. 2022. "A Hitchhiker's Guide Through the Bio-Image Analysis Software Universe." *FEBS Letters* 596 (19): 2472–85. <https://doi.org/10.1002/1873-3468.14451>.
- Haghightab, Mohammad, Saman Zonouz, and Mohamed Abdel-Mottaleb. 2015. "CloudID: Trustworthy Cloud-Based and Cross-Enterprise Biometric Identification." *Expert Systems with Applications* 42 (21): 7905–16. <https://doi.org/10.1016/j.eswa.2015.06.025>.
- Hameed, Ibtihaal M., Sadiq H. Abdulhussain, and Basheera M. Mahmood. 2021. "Content-Based Image Retrieval: A Review of Recent Trends." Edited by D T Pham. *Cogent Engineering* 8 (1): 1927469. <https://doi.org/10.1080/23311916.2021.1927469>.
- Hao, Yuhan, Tim Stuart, Madeline H. Kowalski, Saket Choudhary, Paul Hoffman, Austin Hartman, Avi Srivastava, et al. 2024. "Dictionary Learning for Integrative, Multimodal and Scalable Single-Cell Analysis." *Nature Biotechnology* 42 (2): 293–304. <https://doi.org/10.1038/s41587-023-01767-y>.
- Haralick, Robert M., K. Shanmugam, and Its'Hak Dinstein. 1973. "Textural Features for Image Classification." *IEEE Transactions on Systems, Man, and Cybernetics SMC-3* (6): 610–21. <https://doi.org/10.1109/tsmc.1973.4309314>.
- Harris, Coleman. 2023. "Mxnorm: Apply Normalization Methods to Multiplexed Images." *Comprehensive R Archive Network*. <https://doi.org/10.32614/CRAN.package.mxnorm>.
- Harris, Coleman R, Eliot T McKinley, Joseph T Roland, Qi Liu, Martha J Shrubssole, Ken S Lau, Robert J Coffey, Julia Wrobel, and Simon N Vandekar. 2022. "Quantifying and Correcting Slide-to-Slide Variation in Multiplexed Immunofluorescence Images." Edited by Olga Vitek. *Bioinformatics* 38 (6): 1700–1707. <https://doi.org/10.1093/bioinformatics/btab877>.
- Harris, Coleman, Julia Wrobel, and Simon Vandekar. 2022. "Mxnorm: An R Package to Normalize Multiplexed Imaging Data." *Journal of Open Source Software* 7 (71): 4180. <https://doi.org/10.21105/joss.04180>.
- Heineck, Garrett C., Ian G. McNish, Jacob M. Jungers, Erin Gilbert, and Eric Watkins. 2019. "Using r-Based Image Analysis to Quantify Rusts on Perennial Ryegrass." *The Plant Phenome Journal* 2 (1): 1–10. <https://doi.org/10.2135/tppj2018.12.0010>.
- Hijmans, Robert J. 2020. "Terra: Spatial Data Analysis." CRAN: Contributed Packages. The R Foundation. <https://doi.org/10.32614/cran.package.terra>.
- . 2024. *Terra: Spatial Data Analysis*. <https://doi.org/10.32614/CRAN.package.terra>.
- Hossain, Mohammad D., and Dongmei Chen. 2019. "Segmentation for Object-Based Image Analysis (OBIA): A Review of Algorithms and Challenges from Remote Sensing Perspective." *ISPRS Journal of Photogrammetry and Remote Sensing* 150 (April): 115–34. <https://doi.org/10.1016/j.isprsjprs.2019.02.009>.
- Hossian, A K M Nawshad, and George Mattheolabakis. 2020. "Cellular Migration Assay: An in Vitro Technique to Simulate the Wound Repair Mechanism." In *Wound Regeneration*, 77–83. Springer US.

- https://doi.org/10.1007/978-1-0716-0845-6_8.
- Hu, Yang, Matthew L. Becker, and Rebecca Kuntz Willits. 2023. "Quantification of Cell Migration: Metrics Selection to Model Application." *Frontiers in Cell and Developmental Biology* 11 (May). <https://doi.org/10.3389/fcell.2023.1155882>.
- Inglese, Paolo, Gonçalo Correia, Zoltan Takats, Jeremy K Nicholson, and Robert C Glen. 2018. "SPUTNIK: An R Package for Filtering of Spatially Related Peaks in Mass Spectrometry Imaging Data." Edited by Robert Murphy. *Bioinformatics* 35 (1): 178–80. <https://doi.org/10.1093/bioinformatics/bty622>.
- Jähne, Bernd. 2002. *Digital Image Processing*. 5., rev. and extended ed. Engineering Online Library. Berlin: Springer. <https://doi.org/10.1088/0957-0233/13/9/711>.
- Jan Sauer, Bernd Fischer. 2017. "MaxContrastProjection." Bioconductor. <https://doi.org/10.18129/B9.BIOC.MAXCONTRASTPROJECTION>.
- Jenkinson, Mark, and Stephen Smith. 2001. "A Global Optimisation Method for Robust Affine Registration of Brain Images." *Medical Image Analysis* 5 (2): 143–56. [https://doi.org/10.1016/s1361-8415\(01\)00036-6](https://doi.org/10.1016/s1361-8415(01)00036-6).
- Johannes Textor, Jeffrey Berry, Katharina Dannenberg. 2024. "celltrackR: Motion Trajectory Analysis." *Comprehensive R Archive Network*. <https://doi.org/10.32614/CRAN.package.celltrackR>.
- Jude Hemanth, D., and J. Anitha. 2012. "Image Pre-Processing and Feature Extraction Techniques for Magnetic Resonance Brain Image Analysis." In *Computer Applications for Communication, Networking, and Digital Contents*, 349–56. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-35594-3_47.
- Kaiser, J.-P., and A. Bruinink. 2004. "Investigating Cell-Material Interactions by Monitoring and Analysing Cell Migration." *Journal of Materials Science: Materials in Medicine* 15 (4): 429–35. <https://doi.org/10.1023/b:jmsm.0000021115.55254.a8>.
- Kaufman, Leonard, and Peter J. Rousseeuw. 1990. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley Series in Probability and Statistics. Wiley. <https://doi.org/10.1002/9780470316801>.
- Kawaguchi, Atsushi. 2021. *Multivariate Analysis for Neuroimaging Data*. Milton: CRC Press. <https://doi.org/10.1201/9780429289606>.
- Kim, David, Robert Burkhardt, Susan A. Alperstein, Hamza N. Gokozan, Abha Goyal, Jonas J. Heymann, Ami Patel, and Momin T. Siddiqui. 2022. "Evaluating the Role of z-stack to Improve the Morphologic Evaluation of Urine Cytology Whole Slide Images for High-grade Urothelial Carcinoma: Results and Review of a Pilot Study." *Cancer Cytopathology* 130 (8): 630–39. <https://doi.org/10.1002/cncr.22595>.
- Kohonen, T. 1990. "The Self-Organizing Map." *Proceedings of the IEEE* 78 (9): 1464–80. <https://doi.org/10.1109/5.58325>.
- . 2013. "Essentials of the Self-Organizing Map." *Neural Networks* 37 (January): 52–65. <https://doi.org/10.1016/j.neunet.2012.09.018>.
- Koranne, Sandeep. 2011. *Handbook of Open Source Tools*. 1st ed. SpringerLink. Boston, MA: Springer US. <https://doi.org/https://doi.org/10.1007/978-1-4419-7719-9>.
- Kumar Dubey, Ashutosh, Umesh Gupta, and Sonal Jain. 2022. "Medical Data Clustering and Classification Using TLBO and Machine Learning Algorithms." *Computers, Materials & Continua* 70 (3): 4523–43. <https://doi.org/10.32604/cmc.2022.021148>.
- Lajeunesse, Marc J. 2021. *Juicr: Automated and Manual Extraction of Numerical Data from Scientific Images*. <https://doi.org/10.32614/CRAN.package.juicr>.
- Larsson, Ludvig, Lovisa Franzén, Patrik L Ståhl, and Joakim Lundeberg. 2023. "Semla: A Versatile Toolkit for Spatially Resolved Transcriptomics Analysis and Visualization." *Bioinformatics* 39 (10): btad626. <https://doi.org/10.1093/bioinformatics/btad626>.
- Leigh, Roger, David Gault, Melissa Linkert, Jean-Marie Burel, Josh Moore, Sébastien Besson, and Jason R. Swedlow. 2016. "OME Files - an Open Source Reference Library for the OME-XML Metadata Model and the OME-TIFF File Format," November. <https://doi.org/10.1101/088740>.
- Linkert, Melissa, Curtis T. Rueden, Chris Allan, Jean-Marie Burel, Will Moore, Andrew Patterson, Brian Loranger, et al. 2010. "Metadata Matters: Access to Image Data in the Real World." *Journal of Cell Biology* 189 (5): 777–82. <https://doi.org/10.1083/jcb.201004104>.
- Lowe kamp, Bradley C., David T. Chen, Luis Ibáñez, and Daniel Blezek. 2013. "The Design of SimpleITK." *Frontiers in Neuroinformatics* 7. <https://doi.org/10.3389/fninf.2013.00045>.
- Maheshwari, Gargi, and Douglas A. Lauffenburger. 1998. "Deconstructing (and Reconstructing) Cell Migration." *Microscopy Research and Technique* 43 (5): 358–68. [https://doi.org/10.1002/\(sici\)1097-0029\(19981201\)43:5%3C358::aid-jemt2%3E3.0.co;2-d](https://doi.org/10.1002/(sici)1097-0029(19981201)43:5%3C358::aid-jemt2%3E3.0.co;2-d).
- Maia, Rafael, Hugo Gruson, John A. Endler, and Thomas E. White. 2019. "Pavo2: New Tools for the Spectral and Spatial Analysis of Colour in R." Edited by Robert B. O'Hara. *Methods in Ecology and Evolution* 10 (7): 1097–107. <https://doi.org/10.1111/2041-210x.13174>.
- Masotti, Maria, Nathaniel Osher, Joel Eliason, Arvind Rao, and Veerabhadran Baladandayuthapani. 2023. "DIMPLE: An R Package to Quantify, Visualize, and Model Spatial Cellular Interactions from Multiplex Imaging with Distance Matrices." *Patterns* 4 (12): 100879. <https://doi.org/10.1016/j.patter.2023.100879>.

- 1016/j.patter.2023.100879.
- Matias, Filipe Inácio, Maria V. Caraza-Harter, and Jeffrey B. Endelman. 2020. "FIELDimageR: An R Package to Analyze Orthomosaic Images from Agricultural Field Trials." *The Plant Phenome Journal* 3 (1). <https://doi.org/10.1002/ppj2.20005>.
- Moen, Erick, Dylan Bannon, Takamasa Kudo, William Graf, Markus Covert, and David Van Valen. 2019. "Deep Learning for Cellular Image Analysis." *Nature Methods* 16 (12): 1233–46. <https://doi.org/10.1038/s41592-019-0403-1>.
- "MoleculeExperiment." 2024. *Bioconductor*. <http://bioconductor.org/packages/MoleculeExperiment/>.
- Moore, Josh, Chris Allan, Sébastien Besson, Jean-Marie Burel, Erin Diel, David Gault, Kevin Kozlowski, et al. 2021. "OME-NGFF: A Next-Generation File Format for Expanding Bioimaging Data-Access Strategies." *Nature Methods* 18 (12): 1496–98. <https://doi.org/10.1038/s41592-021-01326-w>.
- Moore, Josh, Daniela Basurto-Lozada, Sébastien Besson, John Bogovic, Jordão Bragantini, Eva M. Brown, Jean-Marie Burel, et al. 2023. "OME-Zarr: A Cloud-Optimized Bioimaging File Format with International Community Support." *Histochemistry and Cell Biology* 160 (3): 223–51. <https://doi.org/10.1007/s00418-023-02209-1>.
- Mostafa, Samih M., and Hirofumi Amano. 2019. "Effect of Clustering Data in Improving Machine Learning Model Accuracy." *Journal of Theoretical and Applied Information Technology* 97 (21): 2973–81. <https://www.jatit.org/volumes/Vol97No21/7Vol97No21.pdf>.
- Mouselimis, Lampros, Sight Machine, Johannes Buchner, Mohammad Haghhighat, Radhakrishna Achanta, and Oleh Onyshchak. 2023. "OpenImageR: An Image Processing Toolkit." *Comprehensive R Archive Network*. <https://doi.org/10.32614/CRAN.package.OpenImageR>.
- Murphy, Robert F. 2014. "A New Era in Bioimage Informatics." *Bioinformatics* 30 (10): 1353–53. <https://doi.org/10.1093/bioinformatics/btu158>.
- Muschelli, John. 2022. "Fslr: Wrapper Functions for 'FSL' ('FMRIB' Software Library) from Functional MRI of the Brain ('FMRIB')." *Comprehensive R Archive Network*. <https://doi.org/10.32614/CRAN.package.fslr>.
- Muschelli, John, Elizabeth Sweeney, Martin Lindquist, and Ciprian Crainiceanu. 2015. "Fslr: Connecting the FSL Software with R." *The R Journal* 7 (1): 163. <https://doi.org/10.32614/rj-2015-013>.
- Nasierding, Gulisong, Grigoris Tsoumacas, and Abbas Z. Kouzani. 2009. "Clustering Based Multi-Label Classification for Image Annotation and Retrieval." In *2009 IEEE International Conference on Systems, Man and Cybernetics*. IEEE. <https://doi.org/10.1109/icsmc.2009.5346902>.
- Niedballa, Juergen, Jan Axtner, Leibniz Institute for Zoo, and Wildlife Research. 2022. "Imageseg: Deep Learning Models for Image Segmentation." *Comprehensive R Archive Network*. <https://doi.org/10.32614/CRAN.package.imageseg>.
- Niedballa, Jürgen, Jan Axtner, Timm Fabian Döbert, Andrew Tilker, An Nguyen, Seth T. Wong, Christian Fiderer, Marco Heurich, and Andreas Wilting. 2022a. "Imageseg: An R Package for Deep Learning-Based Image Segmentation." *Methods in Ecology and Evolution* 13 (11): 2363–71. <https://doi.org/10.1111/2041-210X.13984>.
- . 2022b. "Imageseg: An R Package for Deep Learning-Based Image Segmentation." *Methods in Ecology and Evolution* 13 (11): 2363–71. <https://doi.org/10.1111/2041-210X.13984>.
- Nils Eling, Nicolas Damond, Tobias Hoch. 2020. "Cytomapper." *Bioconductor*. <https://doi.org/10.18129/B9.BIOC.CYTOMAPPER>.
- Ochi, Shota. 2023. "magickGUI: GUI Tools for Interactive Image Processing with 'Magick'." *Comprehensive R Archive Network*. <https://doi.org/10.32614/CRAN.package.magickGUI>.
- Olivoto, Tiago. 2022. "Lights, Camera, Pliman! An R Package for Plant Image Analysis." *Methods in Ecology and Evolution* 13 (4): 789–98. <https://doi.org/10.1111/2041-210X.13803>.
- Ollion, Jean, Julien Cochennec, François Loll, Christophe Escudé, and Thomas Boudier. 2013. "TANGO: A Generic Tool for High-Throughput 3D Image Analysis for Studying Nuclear Organization." *Bioinformatics* 29 (14): 1840–41. <https://doi.org/10.1093/bioinformatics/btt276>.
- Ooms, Jeroen. 2024a. *Magick: Advanced Graphics and Image-Processing in R*. *Comprehensive R Archive Network*. <https://docs.ropensci.org/magick/>.
- . 2024b. *Magick: Advanced Graphics and Image-Processing in R*. <https://ropensci.r-universe.dev/magick>.
- Ooms, Jeroen, and Jan Wijffels. 2024. *Opencv: Bindings to 'OpenCV' Computer Vision Library*. <https://ropensci.r-universe.dev/opencv>.
- Oussous, Ahmed, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, and Samir Belfkih. 2018. "Big Data Technologies: A Survey." *Journal of King Saud University - Computer and Information Sciences* 30 (4): 431–48. <https://doi.org/10.1016/j.jksuci.2017.06.001>.
- Park, Hae-Sang, and Chi-Hyuck Jun. 2009. "A Simple and Fast Algorithm for k-Medoids Clustering." *Expert Systems with Applications* 36 (2): 3336–41. <https://doi.org/10.1016/j.eswa.2008.01.039>.
- Pau, Grégoire, Florian Fuchs, Oleg Sklyar, Michael Boutros, and Wolfgang Huber. 2010. "EBImage—an R Package for Image Processing with Applications to Cellular Phenotypes." *Bioinformatics* 26 (7): 979–81. <https://doi.org/10.1093/bioinformatics/btq046>.
- Paul-Gilloteaux, Perrine. 2023. "Bioimage Informatics: Investing in Software Usability Is Essential."

- PLOS Biology 21 (7): e3002213. <https://doi.org/10.1371/journal.pbio.3002213>.
- Peng, Hanchuan. 2008. "Bioimage Informatics: A New Area of Engineering Biology." *Bioinformatics* 24 (17): 1827–36. <https://doi.org/10.1093/bioinformatics/btn346>.
- Peng, Hanchuan, Alex Bateman, Alfonso Valencia, and Jonathan D. Wren. 2012. "Bioimage Informatics: A New Category in Bioinformatics." *Bioinformatics* 28 (8): 1057–57. <https://doi.org/10.1093/bioinformatics/bts111>.
- Petersen, Ashley, Noah Simon, and Daniela Witten. 2017. "SCALPEL: Extracting Neurons from Calcium Imaging Data." arXiv. <https://doi.org/10.48550/ARXIV.1703.06946>.
- Poisot, Timoth\`ee. 2011. "The Digitize Package: Extracting Numerical Data from Scatterplots." *The R Journal* 3 (1): 25–26. http://journal.r-project.org/archive/2011-1/RJournal_2011-1_Poisot.pdf.
- Polzehl, Jörg, and Karsten Tabelow. 2007. "Adaptive Smoothing of Digital Images: The R Package Adimpro." *Journal of Statistical Software* 19 (1). <https://doi.org/10.18637/jss.v019.i01>.
- Prajapati, Vignesh. 2013. *Big Data Analytics with r and Hadoop*. Online-Ausg. Birmingham: Packt Publishing. https://api.pageplace.de/preview/DT0400.9781782163299_A24165845/preview-9781782163299_A24165845.pdf.
- R Core Team. 2023. "R: A Language and Environment for Statistical Computing." <https://www.R-project.org/>.
- Rabbani, A., A. M. Fernando, R. Shams, A. Singh, P. Mostaghimi, and M. Babaei. 2021. "Review of Data Science Trends and Issues in Porous Media Research With a Focus on Image-Based Techniques." *Water Resources Research* 57 (10): e2020WR029472. <https://doi.org/10.1029/2020WR029472>.
- Real, Carlos. 2024. *Pixelclasser: Classifies Image Pixels by Colour*. <https://github.com/ropensci/pixelclasser>.
- Ren, and Malik. 2003. "Learning a Classification Model for Segmentation." In *Proceedings Ninth IEEE International Conference on Computer Vision*. IEEE. <https://doi.org/10.1109/iccv.2003.1238308>.
- Rittscher, Jens. 2010. "Characterization of Biological Processes Through Automated Image Analysis." *Annual Review of Biomedical Engineering* 12 (1): 315–44. <https://doi.org/10.1146/annurev-bioeng-070909-105235>.
- Robotham, A. S. G., D. S. Taranu, R. Tobar, A. Moffett, and S. P. Driver. 2016. "ProFit: Bayesian Profile Fitting of Galaxy Images." *Monthly Notices of the Royal Astronomical Society* 466 (2): 1513–41. <https://doi.org/10.1093/mnras/stw3039>.
- Rödiger, Stefan, Alexander Böhm, and Ingolf Schimke. 2013. "Surface Melting Curve Analysis with R." *The R Journal* 5 (2): 37–53. <https://doi.org/10.32614/RJ-2013-024>.
- Rödiger, Stefan, Michał Burdukiewicz, Konstantin A. Blagodatskikh, and Peter Schierack. 2015. "R as an Environment for the Reproducible Analysis of DNA Amplification Experiments." *The R Journal* 7 (2): 127–50. <http://journal.r-project.org/archive/2015-1/RJ-2015-1.pdf>.
- Rödiger, Stefan, Michał Burdukiewicz, Konstantin Blagodatskikh, Michael Jahn, and Peter Schierack. 2015. "R as an Environment for Reproducible Analysis of DNA Amplification Experiments." *The R Journal* 7 (1): 127. <https://doi.org/10.32614/rj-2015-011>.
- Rödiger, Stefan, Thomas Friedrichsmeier, Prasenjit Kapat, and Meik Michalke. 2012. "RKWard: A Comprehensive Graphical User Interface and Integrated Development Environment for Statistical Analysis with R." *Journal of Statistical Software* 49 (9): 1–34. <https://doi.org/10.18637/jss.v049.i09>.
- Rödiger, Stefan, Marius Liefold, Madeleine Ruhe, Mark Reinwald, Eberhard Beck, and P. Markus Deckert. 2018. "Quantification of DNA Double-Strand Breaks in Peripheral Blood Mononuclear Cells from Healthy Donors Exposed to Bendamustine by an Automated γH2AX Assay—an Exploratory Study." *Journal of Laboratory and Precision Medicine* 3 (May): 47–47. <https://doi.org/10.21037/jlpm.2018.04.10>.
- Roesch, Elisabeth, Joe G. Greener, Adam L. MacLean, Huda Nassar, Christopher Rackauckas, Timothy E. Holy, and Michael P. H. Stumpf. 2023. "Julia for Biologists." *Nature Methods* 20 (5): 655–64. <https://doi.org/10.1038/s41592-023-01832-z>.
- Rousseeuw, Peter J. 1987. "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis." *Journal of Computational and Applied Mathematics* 20 (November): 53–65. [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7).
- Ruhe, Madeleine, Werner Dammermann, Stefan Lüth, Mandy Sowa, Peter Schierack, P. Markus Deckert, and Stefan Rödiger. 2019. "Effect of Cryopreservation on the Formation of DNA Double Strand Breaks in Human Peripheral Blood Mononuclear Cells." *Journal of Cellular Biotechnology* 4 (1–2): 67–73. <https://doi.org/10.3233/jcb-189006>.
- Russell, Pamela, Kelly Fountain, Dulcy Wolverton, and Debashis Ghosh. 2018. "TCIApathfinder: An R Client for the Cancer Imaging Archive REST API." *Cancer Research* 78 (15): 4424–26. <https://doi.org/10.1158/0008-5472.can-18-0678>.
- Sagiroglu, Seref, and Duygu Sinanc. 2013. "Big Data: A Review." In *2013 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE. <https://doi.org/10.1109/cts.2013.6567202>.
- Satija, Rahul, Jeffrey A. Farrell, David Gennert, Alexander F. Schier, and Aviv Regev. 2015. "Spatial

- Reconstruction of Single-Cell Gene Expression Data." *Nature Biotechnology* 33 (5): 495–502. <https://doi.org/10.1038/nbt.3192>.
- Schaefer, Tim. 2024. "Fsbrain: An R Package for the Visualization of Structural Neuroimaging Data." Zenodo. <https://doi.org/10.5281/ZENODO.10613234>.
- Schäfer, Tim, and Christine Ecker. 2020. "Fsbrain: An R Package for the Visualization of Structural Neuroimaging Data," September. <https://doi.org/10.1101/2020.09.18.302935>.
- Schneider, Caroline A., Wayne S. Rasband, and Kevin W. Eliceiri. 2012. "NIH Image to ImageJ: 25 Years of Image Analysis." *Nature Methods* 9 (7): 671–75. <https://doi.org/10.1038/nmeth.2089>.
- Schneider, Jens, Romano Weiss, Madeleine Ruhe, Tobias Jung, Dirk Roggenbuck, Ralf Stohwasser, Peter Schierack, and Stefan Rödiger. 2019. "Open Source Bioimage Informatics Tools for the Analysis of DNA Damage and Associated Biomarkers." *Journal of Laboratory and Precision Medicine* 4 (0): 1–27. <https://doi.org/10.21037/jlpm.2019.04.05>.
- Schubert, Erich, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. 2017. "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN." *ACM Transactions on Database Systems* 42 (3): 1–21. <https://doi.org/10.1145/3068335>.
- Shariff, Aabid, Joshua Kangas, Luis Pedro Coelho, Shannon Quinn, and Robert F. Murphy. 2010. "Automated Image Analysis for High-Content Screening and Analysis." *SLAS Discovery* 15 (7): 726–34. <https://doi.org/10.1177/1087057110370894>.
- Shirazi, Syed Hamad, Saeeda Naz, Muhammad Imran Razzak, Arif Iqbal Umar, and Ahmad Zaib. 2018. "Automated Pathology Image Analysis." In *Soft Computing Based Medical Image Analysis*, 13–29. Elsevier. <https://doi.org/10.1016/b978-0-12-813087-2.00026-9>.
- Simon Barthelme, Jan Wijffels, David Tschumperle. 2024. "Imager: Image Processing Library Based on 'CImg'." *Comprehensive R Archive Network*. <https://doi.org/10.32614/CRAN.package.imager>.
- Smith, Byron, Meyke HermSEN, Elizabeth Lesser, Deepak Ravichandar, and Walter Kremers. 2021. "Developing Image Analysis Pipelines of Whole-Slide Images: Pre- and Post-Processing." *Journal of Clinical and Translational Science* 5 (1): e38. <https://doi.org/10.1017/cts.2020.531>.
- Sonka, Milan, and J. Michael Fitzpatrick, eds. 2000. *Handbook of Medical Imaging*. SPIE Press Monograph PM80. Bellingham, Wash.: SPIE Press. <https://doi.org/https://doi.org/10.1117/3.831079>.
- Stefan Mayer. 2024. "Stm/Imagefluency: Imagefluency 0.2.5." Zenodo. <https://doi.org/10.5281/ZENODO.10652374>.
- Struyf, Anja, Mia Hubert, and Peter Rousseeuw. 1996. "Clustering in an Object-Oriented Environment." *Journal of Statistical Software* 1 (4). <https://doi.org/10.18637/jss.v001.i04>.
- Swedlow, Jason R., and Kevin W. Eliceiri. 2009. "Open Source Bioimage Informatics for Cell Biology." *Trends in Cell Biology* 19 (11): 656–60. <https://doi.org/10.1016/j.tcb.2009.08.007>.
- Swedlow, Jason R., Ilya G. Goldberg, and Kevin W. Eliceiri. 2009. "Bioimage Informatics for Experimental Biology." *Annual Review of Biophysics* 38 (1): 327–46. <https://doi.org/10.1146/annurev.biophys.050708.133641>.
- Trivedi, Maharshi M., and James K. Mills. 2020. "Centroid Calculation of the Blastomere from 3D z-Stack Image Data of a 2-Cell Mouse Embryo." *Biomedical Signal Processing and Control* 57 (March): 101726. <https://doi.org/10.1016/j.bspc.2019.101726>.
- Um, Eujin, Jung Min Oh, Steve Granick, and Yoon-Kyoung Cho. 2017. "Cell Migration in Micro-engineered Tumor Environments." *Lab on a Chip* 17 (24): 4171–85. <https://doi.org/10.1039/c7lc00555e>.
- Ushey, Kevin, JJ Allaire, and Yuan Tang. 2024. "Reticulate: Interface to 'Python'." <https://doi.org/10.32614/CRAN.package.reticulate>.
- Van der Laan, Mark, Katherine Pollard, and Jennifer Bryan. 2003. "A New Partitioning Around Medoids Algorithm." *Journal of Statistical Computation and Simulation* 73 (8): 575–84. <https://doi.org/10.1080/0094965031000136012>.
- Velden, Bas H. M. van der, Hugo J. Kuijf, Kenneth G. A. Gilhuijs, and Max A. Viergever. 2022. "Explainable Artificial Intelligence (XAI) in Deep Learning-Based Medical Image Analysis." *Medical Image Analysis* 79 (July): 102470. <https://doi.org/10.1016/j.media.2022.102470>.
- Wagih, Omar, and Leopold Parts. 2014. "Gitter: A Robust and Accurate Method for Quantification of Colony Sizes from Plate Images." *G3 Genes|Genomes|Genetics* 4 (3): 547–52. <https://doi.org/10.1534/g3.113.009431>.
- Weiss, Romano, Sanaz Karimijafarbigloo, Dirk Roggenbuck, and Stefan Rödiger. 2022. "Applications of Neural Networks in Biomedical Data Analysis." *Biomedicines* 10 (7): 1469. <https://doi.org/10.3390/biomedicines10071469>.
- Weller, Hannah I., Anna E. Hiller, Nathan P. Lord, and Steven M. Van Belleghem. 2024. "Recolorize: An R Package for Flexible Colour Segmentation of Biological Images." *Ecology Letters* 27 (2). <https://doi.org/10.1111/ele.14378>.
- Wollman, Roy, and Nico Stuurman. 2007. "High Throughput Microscopy: From Raw Images to Discoveries." *Journal of Cell Science* 120 (21): 3715–22. <https://doi.org/10.1242/jcs.013623>.
- Wortel, Inge M. N., Annie Y. Liu, Katharina Dannenberg, Jeffrey C. Berry, Mark J. Miller, and Johannes Textor. 2021. "CelltrackR: An R Package for Fast and Flexible Analysis of Immune Cell Migra-

- tion Data." *ImmunoInformatics* 1-2 (October): 100003. <https://doi.org/10.1016/j.immuno.2021.100003>.
- Yamada, Kenneth M., and Michael Sixt. 2019. "Mechanisms of 3D Cell Migration." *Nature Reviews Molecular Cell Biology* 20 (12): 738–52. <https://doi.org/10.1038/s41580-019-0172-9>.
- Yang, Tianpei, Volkan Ozcoban, Anu Pasam, Nikolce Kocovski, Angela Pizzolla, Yu-Kuan Huang, Greg Bass, et al. 2020. "SPIAT: An R Package for the Spatial Image Analysis of Cells in Tissues," May. <https://doi.org/10.1101/2020.05.28.122614>.
- Yaniv, Ziv, Bradley C. Lowekamp, Hans J. Johnson, and Richard Beare. 2017. "SimpleITK Image-Analysis Notebooks: A Collaborative Environment for Education and Reproducible Research." *Journal of Digital Imaging* 31 (3): 290–303. <https://doi.org/10.1007/s10278-017-0037-8>.
- Yao, Wei, Cornelius Octavian Dumitru, Otmar Loffeld, and Mihai Datcu. 2016. "Semi-Supervised Hierarchical Clustering for Semantic SAR Image Annotation." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 9 (5): 1993–2008. <https://doi.org/10.1109/jstars.2016.2537548>.
- Ye, Yilin, Jianing Hao, Yihan Hou, Zhan Wang, Shishi Xiao, Yuyu Luo, and Wei Zeng. 2024. "Generative AI for Visualization: State of the Art and Future Directions." *Visual Informatics* 8 (2): 43–66. <https://doi.org/10.1016/j.visinf.2024.04.003>.
- Zhao, Chen, and Ronald N Germain. 2023. "Multiplex Imaging in Immuno-Oncology." *Journal for ImmunoTherapy of Cancer* 11 (10): e006923. <https://doi.org/10.1136/jitc-2023-006923>.

Tim Brauckhoff
Brandenburg University of Technology Cottbus - Senftenberg
Institute of Biotechnology
Senftenberg, Germany
Eberhard Karls Universität Tübingen
Department of Computer Science
Tübingen, Germany
ORCID: 0009-0002-0142-7017
brauctile@disroot.org

Julius Rublack
University of Münster
Faculty of Biology
Münster, Germany
juliusrublack@web.de

Stefan Rödiger (corresponding author)
Brandenburg University of Technology Cottbus - Senftenberg
Institute of Biotechnology
Senftenberg, Germany
ORCID: 0000-0002-1441-6512
stefan.roediger@b-tu.de