SOFTWARE TOOL ARTICLE

bwimage: A package to describe image patterns in natural structures [version 1; peer review: 1 approved with reservations]

Carlos Biagolini-Jr. 1, Regina H. Macedo2

1Programa de Pós-Graduação em Ecologia, Universidade de Brasília, Brasília, DF, 70910-900, Brazil
2Departamento de Zoologia, Universidade de Brasília, Brasília, DF, 70910-900, Brazil

Abstract
Currently R is the most popular software for data analyses among biologists. Here, we present bwimage, a package designed to describe patterns from black and white images. The package can be used for a wide range of applications. We implemented functions previously described in the literature to calculate parameters designed originally, but not exclusively, for vegetation structures. Additionally, we propose a new parameter: the aggregation index. We demonstrate applications for field work, providing examples that range from calculation of canopy openness, description of patterns in vertical vegetation structure, to patterns in bird nest structure. We provide advice and illustrated examples of how to produce high quality images for analyses.

Keywords
animal, ecology methods, field, image analyses, image processing, vegetation patterns

This article is included in the RPackage gateway.
Introduction
The facility to obtain high quality digital images creates the opportunity to measure natural variables using image analyses. Black and white pictures have frequently been used to understand patterns in field ecology, especially in plant biology studies\textsuperscript{1}. However, the use of plant image analyses software is not easily extended to other biological fields for several reasons. Free programs are uncommon and paid software normally has threshold algorithms that were specifically designed for vegetation pictures\textsuperscript{2}. Thus, a flexible method that would allow the application of such analyses to other subjects would be welcome. For example, despite the relatively well reported descriptions of bird nests and egg morphology in Del Hoyo and collaborators\textsuperscript{3} (but see Xiao, Hu\textsuperscript{4}), there are no well-established approaches to estimate nest wall openness patterns.

Currently, R software\textsuperscript{5} allows users to migrate from data processing based on combinations of different software (with the possibility of having costly licensing, software-specific files, incompatibility between operating systems and lack of updates) to a free, single cross-platform software. Here, we introduce bwimage, a package for R that can be used to analyze patterns in black and white images from natural structures. We provide data examples for applications and descriptions of routines for processing of black and white images.

Methods
Implementation
Bwimage’s analysis of images is based on the transformation from a picture (“jpeg” and “png” files are allowed) to a binary matrix (Figure 1). For each pixel, the intensity of red, green and blue is averaged and compared to a threshold. If the average intensity is less than the threshold (default is 50%) the pixel will be set as black, otherwise it will be white. Beyond RGB intensity in PNG images, the alpha channel is used to set transparent pixels, i.e. alpha channel values above the threshold (default is 50%) will set the pixel as transparent. In the data matrix, the value one represents black pixels, zero represents white pixels and NA represents transparent pixels. For high resolution files, i.e. numbers of pixels in width and height, we suggest reducing the resolution to create a smaller matrix, as this strongly reduces GPU usage and time necessary to run analyses. However, by reducing resolution, the accuracy of data description will also be lowered. Figure 2 compares different resamplings from a

![Figure 1. General approach for image analysis in the bwimage package. A) An image of a natural structure is obtained with digital photography; here we used an image from a canopy. B) The image is converted into a binary matrix, functions threshold_color (to a single image) or threshold_image_list (for two or more images). In the data matrix the value one represents black pixels, zero represents white pixels and NA represents transparent pixels.](image.png)
figure of 2500x2500 pixels. If the user is not acquainted with scale and threshold processing and/or images were captured under different light conditions, we recommend the scale and application of threshold algorithms in a native image editor software, such as GIMP, and subsequent usage of the resulting images with the bwimage package.

Several metrics can be performed with the functions presented in Table 1. We implemented functions to calculate parameters designed originally, but not exclusively, for vegetation structures (described by Zehm et al. 2003) and propose a new parameter: the aggregation index. The aggregation index is a standardized estimation of the average proportion of same-color pixels around each image pixel. First, the proportion of same-color neighboring pixels (SCNP) is calculated (marginal lines and columns are excluded). Next, the SCNP for all pixels are averaged; then, given the proportion of black and white pixels, number of pixels in height and width, and location of transparent pixels (when present), the maximum and minimum possible aggregation indexes are calculated. Finally, the observed aggregation is standardized to a scale where the minimum possible value is set at zero and the maximum value is set at one (Figure 3).

**Operation**

Bwimage is written in the R programming language, and can be run on Windows, Mac OS X, and Linux systems. The package is available at the CRAN repository, and the development releases are available at Github (see Software availability). The bwimage CRAN page documents package dependencies. Input images must be in one of the following formats PNG, JPG, or JPEG.
**Table 1.** Main functions of the bwimage package.

<table>
<thead>
<tr>
<th>Functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image processes</strong></td>
<td></td>
</tr>
<tr>
<td>threshold_color</td>
<td>Convert an image into a matrix</td>
</tr>
<tr>
<td>threshold_image_list</td>
<td>Convert several images into a list of matrices</td>
</tr>
<tr>
<td>image_information</td>
<td>Summary of image information</td>
</tr>
<tr>
<td><strong>Image analyses</strong></td>
<td></td>
</tr>
<tr>
<td>denseness_total</td>
<td>Proportion of black pixels in relation to all pixels</td>
</tr>
<tr>
<td>denseness_row</td>
<td>Proportion of black pixels in rows subsets</td>
</tr>
<tr>
<td>denseness_column</td>
<td>Proportion of black pixels in column subsets</td>
</tr>
<tr>
<td>hole_row</td>
<td>Description of white pixel continuous sequences in rows</td>
</tr>
<tr>
<td>hole_column</td>
<td>Description of white pixel continuous sequences in columns</td>
</tr>
<tr>
<td>light_gap</td>
<td>Left and right distances from first black pixel to image edge</td>
</tr>
<tr>
<td>heigh_maximum</td>
<td>Higher black pixel in all image</td>
</tr>
<tr>
<td>altitudinal_profile</td>
<td>Description of higher black pixel in image sections</td>
</tr>
<tr>
<td>heigh_propotion</td>
<td>Height below which a given proportion of vegetation denseness occurs</td>
</tr>
<tr>
<td>topline</td>
<td>Line running along the crest of highest black pixel</td>
</tr>
<tr>
<td>aggregation_index</td>
<td>Pixel aggregation estimator</td>
</tr>
</tbody>
</table>

**Use cases**

Canopy openness is one of the most essential ecological parameters for a field ecologist. In the bwimage package, canopy openness can be calculated based on a single picture. To illustrate, we demonstrate below how to analyze a canopy image with the bwimage package. The photo was taken with a digital camera placed in the ground, perpendicular to the ground. Canopy closure can be calculated by estimating the total amount of vegetation in the canopy. Canopy openness is equal to one minus the canopy closure. For this example, we used the original image from Figure 1. The original image file is provided as Underlying data.

```r
canopy_matrix <- threshold_color("canopy.JPG", compress_method="proportional", compress_rate=0.1)
1-denseness_total(canopy_matrix)
[1] 0.1297333
```

Several metrics to describe vertical vegetation complexity can be performed by the bwimage package (see Table 1). Here we provide examples based on an image (Figure 2A) from a vegetation plot of 30x100cm. The original image file is provided as Underlying data. On the 100cm side of this plot we placed a panel of 100x100 cm, covered with white cloth, and perpendicular to the ground. A plastic canvas of 50x100cm was used to cover the vegetation along a narrow strip in front of a camera positioned on a tripod at a height of 55 cm. A photograph of the portion of standing vegetation against the white cloth was taken.

```r
vegetation_matrix <- threshold_color(bush)
denseness_total(vegetation_matrix,height_size = 100, width_size = 100)
[1] 0.115248
topline(vegetation_matrix)
topline
  785.6
heigh_propotion_test(vegetation_matrix,proportion=0.75,height_size=100)
Height below which 0.75 of the vegetation denseness is located
31.2
```
Variation in eggs and nest morphology provide relevant information concerning bird life history that has frequently been used to answer ecological\cite{10-12} and evolutionary questions\cite{13-15}. Here we analyze examples that address the quantification nest wall openness and the aggregation of nest wall holes, using a nest of the blue-black grassquit (\textit{Volatinia jacarina}) deposited in the museum collection Coleção Ornitológica Marcelo Bagno, at Universidade de Brasília (register number COMB-N682). Figure 4 describes how to produce a high-quality image to describe patterns in bird nest wall openness. The original image file used is provided as Underlying data\cite{16}.

---

**Figure 3. Demonstration of aggregation index calculation.** Imagens represent photos of 6x6 pixel, with 50% black and 50% white pixels. For the aggregation index calculation, first, the proportion of same-color neighboring pixels (SCNP) is calculated (marginal lines and columns are excluded). Then, SCNP are averaged. In these examples, the average SCNP is 0.75 (A), 0.5 (B), 0 (C) and 0.875 (D). Next, the maximum and minimum possible aggregation indexes are calculated. In images with 50% black and white pixels, the minimum aggregation will be pixels distributed in chessboard style (C), and the highest aggregation will be the aggregation of all same color pixels on each image side (D). Thus, by scaling the aggregation by the minimum and maximum possible aggregation, the final aggregation index is: 0.857 (A), 0.571 (B), 0 (C) and 1 (D).
Conclusions
The bwimage package provides accessible and simple methods for ecologists and field researchers to describe patterns from black and white digital images. It is a flexible method that allows the application of image analyses to an exceptionally broad range of research subjects. Bwimage’s analysis is based on a simple computational routine based on the transformation of a picture (“jpeg” or “png” files) into a binary matrix, followed by the analysis itself. Several metrics can be calculated by the bwimage package. We implemented functions previously described in the literature, and additionally, we proposed a new parameter: the aggregation index, which generates a standardized estimate of the average proportion of same-color pixels around each image pixel. The application of this methods is exceptionally broad.

Data availability
Underlying data
Figshare: Canopy of Royal poinciana. https://doi.org/10.6084/m9.figshare.8429117.v2

Figshare: Image from a vegetation plot of 30x100cm. https://doi.org/10.6084/m9.figshare.8429882.v2

Figshare: Blue-black grassquit (Volatinia jacarina) nest. https://doi.org/10.6084/m9.figshare.8432018.v1

Data are available under the terms of the Creative Commons Zero “No rights reserved” data waiver (CC0 1.0 Public domain dedication).

Software availability
- Software available from: https://CRAN.R-project.org/package=bwimage
- Source code available from: https://github.com/biagolini/bwimage
- Archived source code at time of publication: https://doi.org/10.5281/zenodo.3266299

- License: GPL-3

---

**Grant information**

CB received graduate fellowships from Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Acknowledgments**

We are grateful to Universidade de Brasília for logistic support and Marcelo Antônio de Assis Silva for technical advice on how to obtain a high-quality image of the bird nest used in the manuscript.

---

**References**


Open Peer Review

Current Peer Review Status: ?

Version 1

Reviewer Report 13 August 2019

https://doi.org/10.5256/f1000research.21722.r51562

© 2019 Chianucci F. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Francesco Chianucci
The Council for Agricultural Research and Economics (CREA) - Research Centre for Forestry and Wood, Arezzo, Italy

The topic is interesting. However, I think the package makes simple things which many other similar packages do, and omit some intermediate steps required to refine the analysis.

First, the authors should provide a list of thresholding (either single or dual) methods to binarize images. The single Otsu, the minimum algorithm, the two-corner method are some of the algorithm which should be developed. The package could also use some dependancies from other packages to implement these thresholding. For instance, the rtiff package contains the function autoThreshold which implement the Ridler-Calvard thresholding. I also suggest take a look at the caiman package which contains tool for canopy image analysis, see also below.

Second, users can also process single channel image or decide to use a single channel (for example, the blue channel is frequently used for canopy images). So the user should have flexibility on choosing the image feature to process (setting the channel, setting the threshold).

Canopy images are usually fish eye images, and therefore they require correction for lens projection. In addition, canopy openness should be weighted for zenith angle in fish eye image. Similarly, gap fraction is required for zenith ring. Thus, the packages should allow to work with fish eye (circular image), namely correct for lens distortion, setting the circular inner mask, extract information for zenith angle ranges (inner rings). I suggest to take a look in the caiman package to inherit its functions.

An interesting attribute from the packages is the identification of row or column gaps (holes), namely continuous sequences of white pixels. In my view, a very interesting output would be the identification and the labelling of all holes in the image, along with their summary statistics (e.g. number of holes, size of each hole in number of pixels, average, sd and so on). I think the package will improve strongly if authors can implement such function. This would allow the extraction of canopy attributes from images.1

References

**Is the rationale for developing the new software tool clearly explained?**
Yes

**Is the description of the software tool technically sound?**
Yes

**Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?**
Partly

**Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?**
Partly

**Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?**
Partly

**Competing Interests:** No competing interests were disclosed.

**Reviewer Expertise:** Forestry

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 16 Oct 2019

Carlos Biagolini-Jr., Universidade de Brasilia, Brasilia, Brazil

Dear Dr. Francesco Chianucci,

We appreciate all of your helpful comments, and we are certain that they have improved significantly the original version of the Package. In the new version (1.2) of bwimage package published in CRAN we: i) introduced two new functions (stretch and compress functions) to provide an additional tool to distort images. These functions are applications of algorithms for mapping images from circle to square, and vice versa, adapted from Lambers (2016 *J of Computer Graphics Techniques*); and ii) included the possibility of users processing single channel images.

We considered incorporating other packages for threshold in bwimage dependencies; however, we decided to avoid a cross-dependency of packages because it implies in a heavy load package, and brings possible compatibility issues with R updates. Thus, we chose to create a tutorial (shared on [https://www.r-bloggers.com/using-bwimage-r-package-to-describe-patterns-in-images-of-natural-structures/](https://www.r-bloggers.com/using-bwimage-r-package-to-describe-patterns-in-images-of-natural-structures/)) to provide an overview of bwimage package and demonstrate examples of how to apply threshold algorithms from the package autothresholddr, following its application in bwimage...
package. A comparison of estimation of vegetation density from a bush image submitted to different thresholding algorithms is provided. Note: the current version of autothresholdr package (1.3.5) provide 17 threshold algorithms by function auto_thresh, covering a wide range of applications.

We do not mention in the previous version of this article, but bwimage version 1.0 already had a function (hole_section_data) to summarize holes statistics (i.e. number of holes, mean hole size, sd, minimum and maximum size). This function was designed to be used inside a loop or apply-family functions. By combination of hole_section_data and loop function, users can potentially collect a summary of holes statistics. The size of each section is obtained by the hole_section function, which returns size and map of each hole. We add information about this function in manuscript table 1. We also provide an example of how to apply hole_section_data for a set of 12 images in the above-mentioned tutorial.

With my best regards.
Sincerely,
Carlos Biagolini-Jr

**Competing Interests:** No competing interests were disclosed.