Received: 26 April 2024 | Accepted: 22 October 2024

DOI: 10.1111/2041-210X.14451

Methods in Ecology and Evolution

APPLICATION

RStoolbox: An R package for remote sensing data analysis

Konstantin Müller[1] | **Jakob Schwalb-Willmann**[1] | **Martin Wegmann**[1] | **Benjamin Leutner**[2]

[1]Earth Observation Research Cluster, Department of Remote Sensing, University of Würzburg, Würzburg, Bavaria, Germany

[2]The Landbanking Group, Munich, Bavaria, Germany

Correspondence
Konstantin Müller
Email: konstantinfinn.mueller@gmx.de

Handling Editor: Carlos Alberto Silva

Abstract

1. The role of Satellite Remote Sensing in monitoring the Earth's surface is more important than ever, as it allows us to see changes in space, time, and across the electromagnetic spectrum. Therefore, it is crucial to not only gather data but also to analyse, visualize and present the findings.
2. RSTOOLBOX *R* package offers a suite of functions for (a) preprocessing, (b) analysis and (c) visualization of (multi-band) remote sensing data, implementing state-of-the-art methods such as unsupervised and supervised classification, or spectral unmixing or change vector analysis. Thereby, RSTOOLBOX enables various levels of users, from students to experts, to process and scientifically analyse different kinds of remote sensing data within a single programming environment.
3. To best integrate in pre-existing workflows, RSTOOLBOX is based on well-established data types for representing spatial data in *R* and inherits well-known packages popular within the spatial data science and remote sensing research communities.
4. To showcase the simple usage of RSTOOLBOX we provide multiple examples with sample data provided directly within the package.

KEYWORDS
earth observation, R programming language, remote sensing, spatial data analysis

1 | INTRODUCTION

The application of satellite remote sensing has shown continuous increment over the past decades, (Cracknell, 2018). The increasing availability of open-access data such as Landsat and Sentinel for local to global scale analyses lead to incremental growth of the importance of its scientific findings, (Radočaj et al., 2020). With global change, processes such as natural disasters, urban growth, land transformation, etc. became increasingly visible on the Earth's surface, within only the few decades Earth has been observed by remote sensing satellites, (Pricope et al., 2019). Since the vast majority of these data are freely available, there is a need for open-source tools and easy-to-apply methods to work with them and transform them into useful information. Data ranges from radio detection and ranging (RADAR) to multi-spectral or hyper-spectral data. Each data type has its specialized purpose and comes with specific requirements to correctly analyse the data (Bioucas-Dias et al., 2013; Richards, 2005). This results in many different specifications and requirements remote sensing software must be equipped with. Hence, well-implemented methods for data analysis and data processing are key to environmental research applications using remote sensing data.

In this regard, the *R* programming language (R Core Team, 2022) stands out from closed-source software, which treats methods and algorithms in a black box manner and does not allow any insight into the source code. The *R* language is designed for handling statistical, but also spatial data and has an ever-growing community around satellite imagery analysis. In 2010, the RASTER package (Hijmans, 2023a) was released built upon the SP package (Bivand et al., 2013; Pebesma & Bivand, 2005) for handling spatial data. Both packages' core dependency over the past years was RGDAL, Bivand et al. (2006–2023), first developed in 2006. However, RGDAL had been marked as deprecated and finally archived since 1 January 2024. Due to this, RASTER and SP. have been replaced by TERRA, Hijmans (2023b), and SF, Pebesma (2018); Pebesma and Bivand (2023), as successors of the foundation of processing spatial data. Thus, because of this significant shift in dependencies, we entirely refactored RSTOOLBOX. Both are state-of-the-art packages of *R* when it comes to processing spatial data, not depending on RGDAL. Hence, RSTOOLBOX inherits them as their core dependencies. Further core dependencies of RSTOOLBOX are CARET (Kuhn, 2008) for powerful machine-learning analysis and statistical modelling as well as GGPLOT2 (Hadley Wickham, 2016) for state-of-the-art visualizations of Earth Observation data in *R*. Additionally, RSTOOLBOX implements performance-critical core features of its methods in $C++$, ISO (2012) to speed up computation where possible. The functions then are integrated into the *R* environment utilizing the RCPP package published by Eddelbuettel and Balamuta (2018).

To conclude, RSTOOLBOX provides a powerful and very versatile pipeline of data-preparation, processing and post-visualization. It integrates seamlessly into other state-of-the-art packages and supplies complex algorithms, many of them implemented with C++ for efficiency, through easy-to-use functions.

2 | PACKAGE OVERVIEW

RSTOOLBOX provides a lot of functions needed for a solid base analysis while giving insights into its implementation since *R* and all its packages are open-source. We contribute methods to pre-process, analyse, and present or post-process data as follows:

1. *Preprocessing*: RSTOOLBOX provides various pre-processing methods to prepare remote sensing data for data analysis, as well as for explorative analysis. Those include radiometric conversions (Thuillier et al., 2003), topographic illumination correction (Riaño et al., 2003) as well as cloud and cloud-shadow masking. Here, we emphasize the ability of RSTOOLBOX to normalize and co-registering images for working with multi-source data if needed. Additionally, it includes histogram matching (Jia, 2006) to generate comparable image data across acquisitions, which is helpful for mosaicing. For radiometric processes, masking, as well as analysing and parsing QA-bands of satellite imagery, our package focuses primarily, yet not exclusively, on the Landsat mission (USGS, 2024).
2. *Analysis*: Various functions provide processing methods for remote sensing data. RSTOOLBOX provides a large array of established spectral indices, such as the normalized difference vegetation index (NDVI) etc. Moreover, we provide dimensionality reduction techniques such as principal component analysis (PCA) or tasselled cap transformations (TCT) with pre-defined parameters for multiple multispectral satellites (Baig et al., 2014; Crist, 1985; Huang et al., 2002; Ivits et al., 2008; Lobser & Cohen, 2007; Schoenert et al., 2014; Yarbrough et al., 2005). For subsequent analyses, RSTOOLBOX contributes functions for both fitting or training both unsupervised or supervised classification and regression models using machine learning techniques. Building on an efficient data pipeline, users can choose between many modelling algorithms implemented by CARET, such as random forest, xgboost or support vector machines. For sub-pixel analyses, RSTOOLBOX provides functions for spectral mixture analyses, for example using a multiple endmember spectral analysis (Spectral Unmixing) (Franc et al., 2005). Finally, our package offers functions to identify patterns: temporally by implementing a change vector analysis (CVA), spatially through fractional cover mapping, and spectrally with spectral angle mapping.
3. *Postprocessing and visualization*: For completeness, RSTOOLBOX also comes with additional import and export methods to load and save files in a *.RSTBX* format beyond the functions provided by for example, TERRA or STARS. Furthermore, we provide functions for reading and writing *.SLI* ENVIs binary spectral libraries format, Imaging and Lab (2024), as well as reading meta-data of remote sensing data. For visualization, RSTOOLBOX is built to seamlessly integrate into the workflow of state-of-art plotting libraries without conversion of the output data it returns to preserve a fast and easy workflow for example by the GGPLOT2 extensions ggR() and ggRGB().

Every method implemented has been carefully revised and implemented after the TIDYVERSE style guide, Hadley Wickham (2019). To get started quickly, we provide Landsat 5 TM and Sentinel 2A multi-spectral raster samples for direct copy-paste examples within the code listings in Section 3.

3 | SOFTWARE AND EXAMPLES

RSTOOLBOX functions are dedicated to allowing a seamless transition between the output of one method and the input of the next function using data pipes. This way, chains of functions can be used to analyse satellite imagery with just a few lines of code. In the following, we demonstrate different example workflows that could be used in terms of spectral analysis, supervised and unsupervised classification, and spectral unmixing.

3.1 | Spectral analysis

In the following code snippet, we perform a visual comparison of two spectral indices, the NDVI and the Kernel-Based NDVI (KNDVI). Both indices are calculated from the red and near-infrared band of the example Sentinel 2A scene directly included in RSTOOLBOX and accessible via the sen2 variable.

As shown in Listing 1, only a few lines of code are required to calculate spectral indices. Example data is loaded as variables into the user environment when the library is loaded. Here sen2, a TERRA SpatRaster, is passed to spectralIndices() to calculate two indices defined via the indices argument: NDVI and KNDVI. All spectral bands we needed for the calculation are provided by passing on the band names to the according band argument, here "B4" for red and "B8" for nir. For more information on the spectralIndices() function, we refer to the respective manual or package documentation (Leutner et al., 2024). The function implements more than 30 spectral indices equations. To conclude this simple spectral analysis, one may visualize the outcome. Hence, we follow up with creating a handy theme t we will use more often via GGPLOT2's theme method. Subsequently, we open up a simple plot with the ggR() function. Note how ggR() can inherit every grammar argument of GGPLOT2 as scale_fill_gradientn() and the previously created theme t. This solidifies the ability of RSTOOLBOX to fit into the state-of-the-art *R* environment to seamlessly integrate with widely used *R* packages such as GGPLOT2. Furthermore, plotting functions of RSTOOLBOX will automatically arrange multiple plots according to the layer argument setting to the number of layers plotted, here, sen2. To ensure that every plot will have the same value range to be visualized, we use the stretch argument with a linear stretch and follow it up with a custom colour scale of the VIRIDISLITE Garnier et al. (2023) package. Note that geom_raster is important, as it enables GGPLOT2 to recognize and extend our pre-plotted raster layer. Setting geom_raster to TRUE forces ggR() to map raster values to a GGPLOT2 fill scale instead of using an alpha scale. This way, scale_fill_* functions can be used for colouring. In addition, you can layer as many geom_raster layers on top of each other (ggLayer = TRUE), which allows for example to use transparency to visually combine data, such as a hillshade raster and elevation layer. We present the result of the visual comparison of the NDVI and KNDVI of Listing 1 in Figure 1.

Importantly, calculating spectral indices using spectralIndices() in RSTOOLBOX is on average 2.3 times faster than using standard band calculation with the SpatRaster class, as benchmarks of calculating all implemented indices on the Landsat 5 TM example scene with and without RSTOOLBOX show (tested on AMD Ryzen 7 PRO 7840U 3.3–5.1 GHz with 32 GB RAM). Finally, we provide a simple way of calculating custom spectral indices via our C++ integration. For this,

```r
# Load necessary libraries
library(RStoolbox)
library(viridisLite)
library(ggplot2)

# Calculating spectral indices
idx <- spectralIndices(sen2, red="B4",
  nir="B8", indices=c("NDVI", "KNDVI")
)

# Visualization including ggplot2
t <- theme(axis.text=element_blank(),
           axis.title=element_blank())

ggR(idx, layer=1:2, geom_raster=T,
  stretch="lin") + t +
  scale_fill_gradientn(colors=viridis(20))
```

LISTING 1 Code to easily set up a calculation of basic spectral indices utilizing RSTOOLBOX.

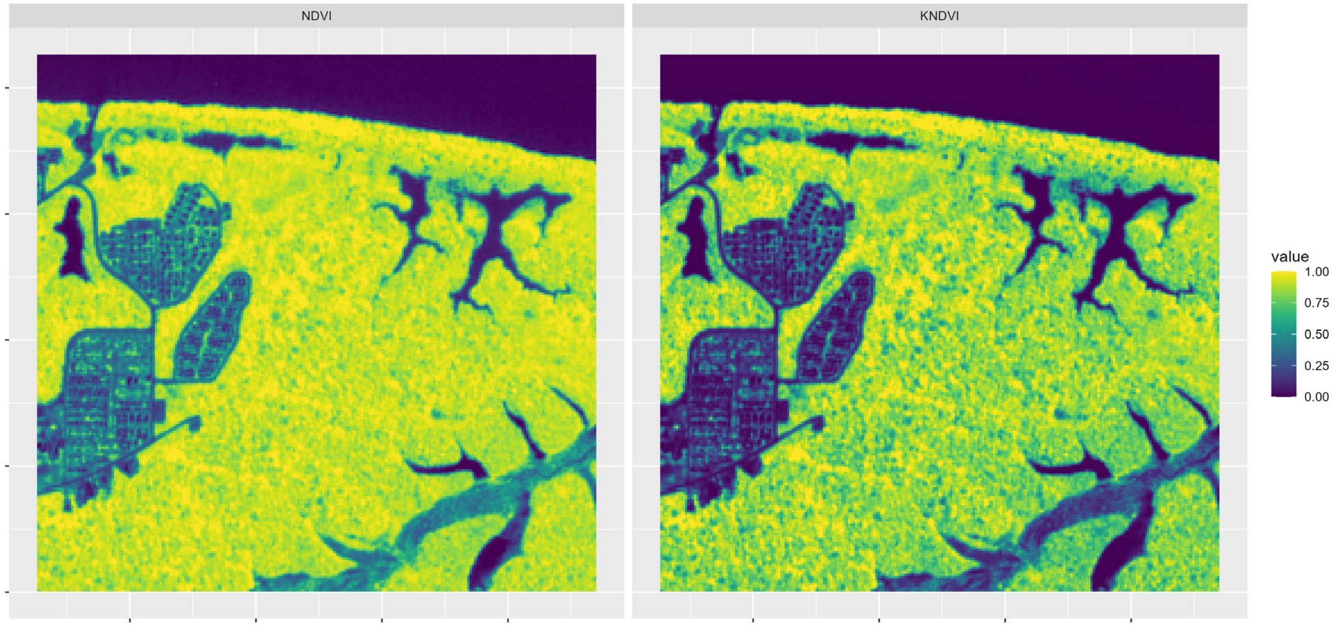

FIGURE 1 A visual comparison of the normalized difference vegetation index (NDVI) (left) and the Kernel-Based NDVI (KNDVI) (right) of the example Sentinel 2A multi-spectral scene provided in RSTOOLBOX.

we provide example code within the Supporting Information as well as the code utilized for our spectralIndices()-benchmark.

3.2 | Classification

In the following section, we present RSTOOLBOX's ability to perform both unsupervised and supervised classification as well as PCA to get a deeper understanding of the scene. For this, we execute the code snippet of Listing 2. We classify the Sentinel 2a sample scene with training polygons also included in RSTOOLBOX. Note that we adopt the previously created theme t and all libraries loaded until this point from Listing 1.

Within Listing 2, we load a file containing labelled polygons. Note that RSTOOLBOX provides both polygons and also a point dataset for training by default as spatial SF data frames. In practice, users would supply their own reference polygons. Subsequently, we conduct our initial analysis employing PCA. With this, one can reduce the dimensionality of a multi-band raster resulting in pixel values corresponding to the most significant source of variation in the image. Instances with high values on the resulting principal component have higher projections along the direction of maximum variance in the original data. This is done by calling rasterPCA() onto the Landsat 5 TM (lsat) data. Moreover, we encourage to print a report of the result via print(pca) to gain a deeper insight into the single components and eigenvalues computed. The resulting raster is accessed via pca$map. For further analysis, we perform both a supervised and unsupervised classification using RSTOOLBOX methods superClass() and unsuperClass() respectively. Note that for the supervised classification, the packages RANDOMFOREST and LATTICE are loaded automatically, both included within the suggested packages list of RSTOOLBOX. While unsupervised classification clusters pixels based purely on their similarity, supervised classification requires training data, such as polygons with known land-cover, and then fits a machine learning model that can map all pixels of the same land-cover. For the supervised approach we provide a lot of parameters to achieve high customizability of the modelling process. Those include the fitting algorithm, the number of samples, tuning variables, or optional export options. Here, we use the example Sentinel 2A image (sen2) as predictor dataset and polys as labels, containing the response variable, to train a random forest model, set via model = rf. The amount of trees is set to 100. Since RSTOOLBOX inherits the CARET machine learning package as a core dependency for this kind of analysis, many more parameters, tuning options and classifiers are available. Importantly, the responseCol attribute determines which feature should be aligned with the layers in the image. Finally, we set the trainingPartition to be 0.7 to split the samples of polys into 70 % for training and 30 % for validation. An extensive report on the model's performance and parameters can be viewed via printing sc$validation. For the unsupervised classification, we utilize the Kmeans algorithm, which requires setting the number of classes, knowing, that the scene has roughly four land-cover types: *water* (blue), *village* (green), *forest* (pink) and *dryout* (black). Note, that unsupervised classification has no notion of thematic classes (unlike supervised classification), hence the classes derived may or may not respond to the classes of the supervised classification. Finally, Listing 2 includes an example of visualizing the classification results. In the first part, we utilize the GRIDEXTRA dependency of RSTOOLBOX and arrange a true colour composite of our original image sen2 and the resulting PCA. Again red, green, and blue bands are provided by either the names or their indices. At the same time, we again use the theme form Listing 1 to it. After that, we use ggR() with the geom_raster attribute and again a colour scale to

```
# Loading additional library files
library(terra)
library(gridExtra)

# Loading the provided training files
polys <- readRDS(system.file(
  "external/trainingPolygons_sen2.rds",
  package="RStoolbox"))

# Performing Principal component analysis
pca <- rasterPCA(lsat)

# Performing (un-)supervised classification
sc <- superClass(sen2, polys,
  model="rf", responseCol="class",
  trainPartition=0.7, ntrees=100
)
uc <- unsuperClass(sen2, nClasses=4)

# Visualization
grid.arrange(
  ggRGB(lsat, r=3, g=2, b=1, stretch="sqrt") + t,
  ggRGB(pca$map, r=1, g=2, b=3, stretch="lin") + t, ncol=2)

grid.arrange(
  ggR(sc$map, geom_raster=T, forceCat=T) +
    scale_fill_identity(labels=levels(
      polys$class
  ), guide="legend", name="Classes") + t,
  ggR(uc$map, geom_raster=T, forceCat=T) +
    scale_fill_identity(labels=paste(
      "Class", c("A", "B", "C", "D")
    ), guide="legend", name="Classes") + t)
```

LISTING 2 Code for replicating a simple principle component analysis as well as a supervised and supervised classification using RSTOOLBOX's example data.

visualize higher and lower projections along the direction of maximum variance in the pca$map raster. Through RSTOOLBOX seamless inheritance of GGPLOT2, it also fits methods of GRIDEXTRA. The resulting plots are shown in Figure 2.

To conclude, we repeat our scheme of ggR() already used in Listing 1 to wrap the classification layers again in a GRIDEXTRA function call. Both classifications are plotted as facets using ggR() where we additionally set forceCat to TRUE to factorize the output. For completeness, we provide the GGPLOT2 method of scale_fill_identity() to customize the legend. We depict the result in Figure 3.

3.3 | Spectral unmixing

Apart from pixel-based classifications, RSTOOLBOX implements spectral unmixing to determine the fractions of different reflectance components on a sub-pixel scale based on their spectral signature. Listing 3 depicts a simple workflow to perform a multiple endmember spectral mixture analysis (MESMA) using non-negative least squares (NNLS) (Franc et al., 2005). This example employs the same libraries previously loaded in Listings 1 and 2.

First, endmembers, that is sets of spectral signatures, are extracted by cell number from the example Landsat 5 scene *lsat*. These endmembers represent three target classes, *forest*, *water* and *shortgrown vegetation*. Here, each of the three target classes is composed of three endmembers, that is optically selected reflectance values extracted from the Landsat scene by cell number. However, the user is free to choose the number of endmembers per class to unmix from. Endmembers are saved in variable *em*, with each row representing an endmember and columns representing the same spectral bands as in the Landsat image. An additional column named "class" contains the name of the class that the endmember belongs to. Next, the input Landsat 5 scene *lsat* is spectrally unmixed by calling the mesma() function. RSTOOLBOX implements a sequential coordinate-wise algorithm (SCA) based on Franc et al. (2005) to apply an NNLS regression for spectral unmixing. If the class column was missing, mesma() interpreted each row as a single class and thus performed a simple spectral mixture analysis (SMA). Since multiple endmembers per class are provided in this example, mesma() first picks a number of endmember combinations (determined by the optional argument n_models) drawn from *em* on which it computes individual SMA. Concurrent with the MESMA approach proposed by Roberts et al. (1998), mesma() then selects the best NNLS regression fit per pixel based on the lowest RMSE (Root Mean Square Error). To achieve best results, the user should adjust n_models in accordance to the number of endmembers per class provided so that as many endmember combinations as possible (with each endmember being used once) are computed. As a result, the function returns a SpatRaster, saved as *fracs*. It contains one layer per input class, each representing the estimated fraction (abundance) of the respective class per pixel, as shown in Figure 4. The last layer of the result is the RMSE layer, which the user can use to judge the MESMA fit per pixel. Overall, the workflow shows how users can quantify occurrences of materials or surfaces on a sub-pixel scale using RSTOOLBOX MESMA implementation.

4 | DISCUSSION

Remote sensing analysis is well-established in multiple fields, such as Geography, Biology or Archaeology, and is frequently used for environmental analysis. R packages such as RSTOOLBOX are important for making remote sensing methods accessible in these fields. The level of adoption is reflected by the usage statistics of RSTOOLBOX: Since its initial release, the package has been downloaded more than 169,000 times from CRAN (2024) and has been cited more than 260 times in articles from various disciplines, such as forest species mapping, heat exposure analysis and climate change research (Grabska et al., 2020; Muñiz-Castillo et al., 2019; Prăvălie et al., 2022). This indicates that RSTOOLBOX integrates well in research projects across many remote-sensing-related fields, as *R* has become a primary coding environment to implement methods and analyse data. Thereby, we provide many algorithms with a C++ integration for increased performance and reduced computational time on large datasets for future big-data scenarios. This goes along with modern, state-of-the-art packages like TERRA of SF utilizing C++ code for performance-critical

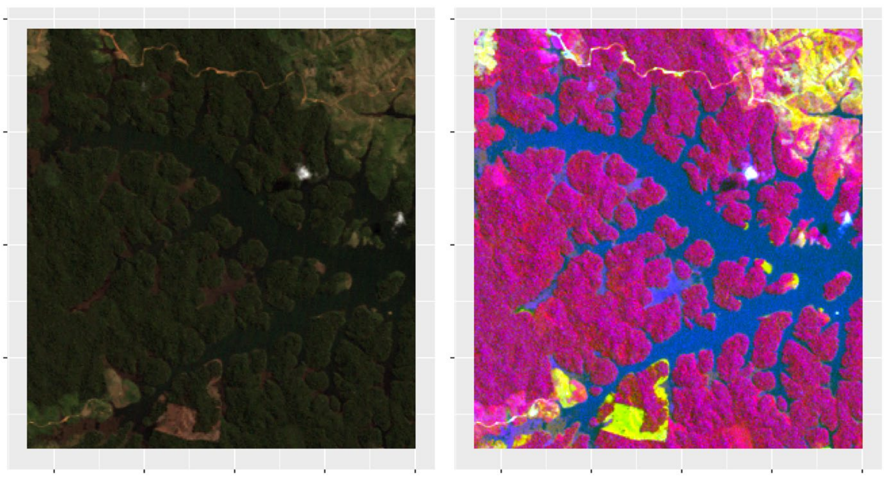

FIGURE 2 A true colour composite (left) of the example Landsat 5 TM multi-spectral scene of RSTOOLBOX and the principal component analysis.

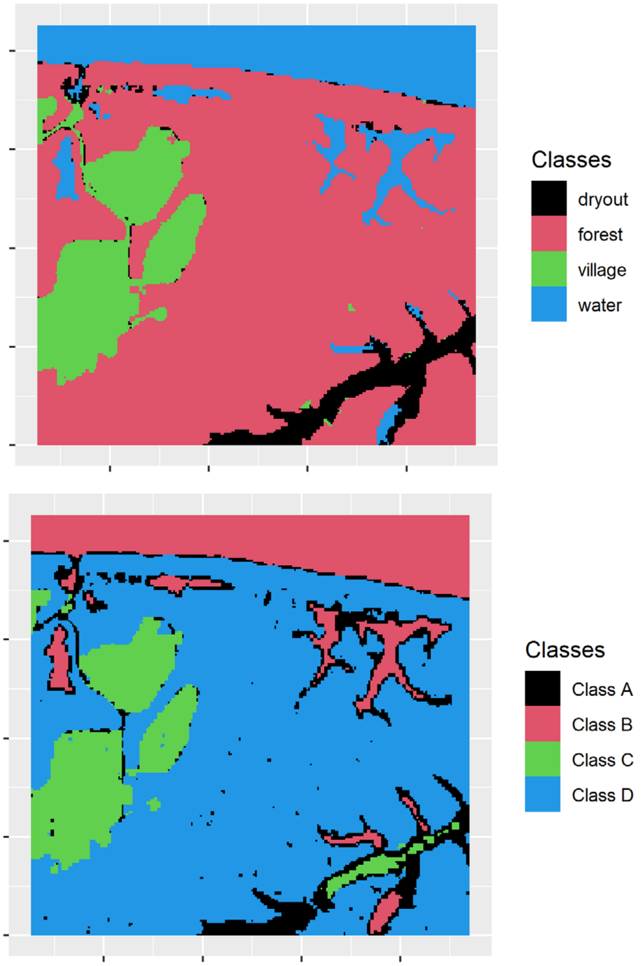

FIGURE 3 Supervised (top) and unsupervised (bottom) classification of the example Sentinel 2A multi-spectral image. The supervised version has been trained on the by RSTOOLBOX provided trainigPolygons_sen2.rds file.

core features. Moreover, *R* and remote sensing are also frequently used in teaching young scientists. RSTOOLBOX offers a less steep learning curve for the implementation of remote sensing methods than implementing them from scratch. RSTOOLBOX is therefore ideal for doing the first steps in remote sensing analysis. Even though RSTOOLBOX is already providing a wealth of well-established methods, there are still methods missing, especially in the realm of RADAR and LIDAR (light detection and ranging) analysis. It would enrich the package to implement such in the future. This could be, for example, achieved by integrating existing packages such as LIDR (Roussel et al., 2020; Roussel & Auty, 2024) as dependencies. A particular focus on algorithms for unmanned aircraft systems (UAS) data analysis would elevate the package's applicability since it would take into account the increasing usage of high-resolution systems in today's scientific data acquisition techniques. This development would need to address UAS image preprocessing, for example by integrating OpenDroneMap (ODM, 2020) to process single images into two and three-dimensional geographic data. This would include steps such as georeferencing, ortho-mosaicing and photogrammetry options like structure from motion (Hill, 2009; Matias et al., 2020; Westoby et al., 2012). Additionally, further functions focusing on hyperspectral remote sensing data will be implemented in forthcoming RSTOOLBOX updates in order to serve this field of remote sensing research, such as hyperspectral indices and preprocessing approaches (Roberts et al., 2018). Finally, a focus on deep learning algorithms for robust data analysis would greatly complement the overall power of RSTOOLBOX. This also opens possibilities for applying pre-trained or self-training models with already implemented methods of the toolbox. Likewise, the focus on the long-term historical Landsat archive and hence methods specific to the Landsat fleet need to be expanded to the Sentinel fleet by the European Space Agency (ESA). Further specific methods for other data providers could also be added, in case long-term availability is ensured.

5 | CONCLUSION

RSTOOLBOX package is been primarily developed to facilitate the application of earth observation methods within the *R* programming environment. Among frequently used remote sensing functionalities also methods to conduct more specific analysis are implemented, such as fractional cover analysis or spectral unmixing. This results in a broad range of possibilities to analyse remote sensing data to the required extent combined in one package.

Therefore, RSTOOLBOX is on the one hand supporting non-remote sensing experts to easily implement remote sensing in their analysis and on the other hand also provides methods that are more complex to implement and thus are potentially interesting for experienced remote sensing scientists as well. Hence, as shown in Section 3 we strongly focus on uniform data types and seamlessly interlocking methods for a clean and fast workflow. RSTOOLBOX empowers users to implement remote sensing methods in their research without the need to program all specific methods themselves. The relevance of this package in various research projects can be shown by the various references in diverse publications (Kamusoko, 2019; Lemenkova & Debeir, 2022; Zhiminaicela-Cabrera et al., 2020), as well as in the context of textbooks for training Geographic Information Systems and remote sensing (Wegmann et al., 2016, 2020). Overall, RSTOOLBOX aims to enable researchers and students to conduct complex analysis procedures within one single package all within *R*, fully open-source.

AUTHOR CONTRIBUTIONS

Bejnamin Leutner and Ned Horning are the original authors of the package. Benjamin Leutner developed, released and maintained the initial version accompanying a textbook on remote sensing applications using Open Source software Wegmann et al. (2016). Benjamin Leutner and Martin Wegmann tested it extensively within various postgraduate training courses. Jakob Schwalb-Willmann joined afterward and added further methodology for subsequent releases of the package. In 2023, the project underwent a major refactoring addressing critically needed adaptation due to essential updates and library changes, led by Konstantin Müller. Konstantin Müller led the writing of the manuscript, Jakob Schwalb-Willmann authored the chapter on spectral unmixing. All authors contributed critically to the drafts and gave final approval for publication.

```r
# Extraction of three endmembers per class
em <- rbind(
  data.frame(lsat[c(4155, 17018, 53134)], class="forest"),
  data.frame(lsat[c(22742, 25946, 38617)], class="water"),
  data.frame(lsat[c(4330, 1762, 1278)], class="shortgrown")
)

# Unmixing of Landsat 5 image using mesma
fracs <- mesma(lsat, em)

# Visualization of fractions and RMSE
plot(fracs)
```

LISTING 3 Code to perform a multiple endmember spectral mixture analysis (MESMA) with the Landsat 5 example scene of RSTOOLBOX.

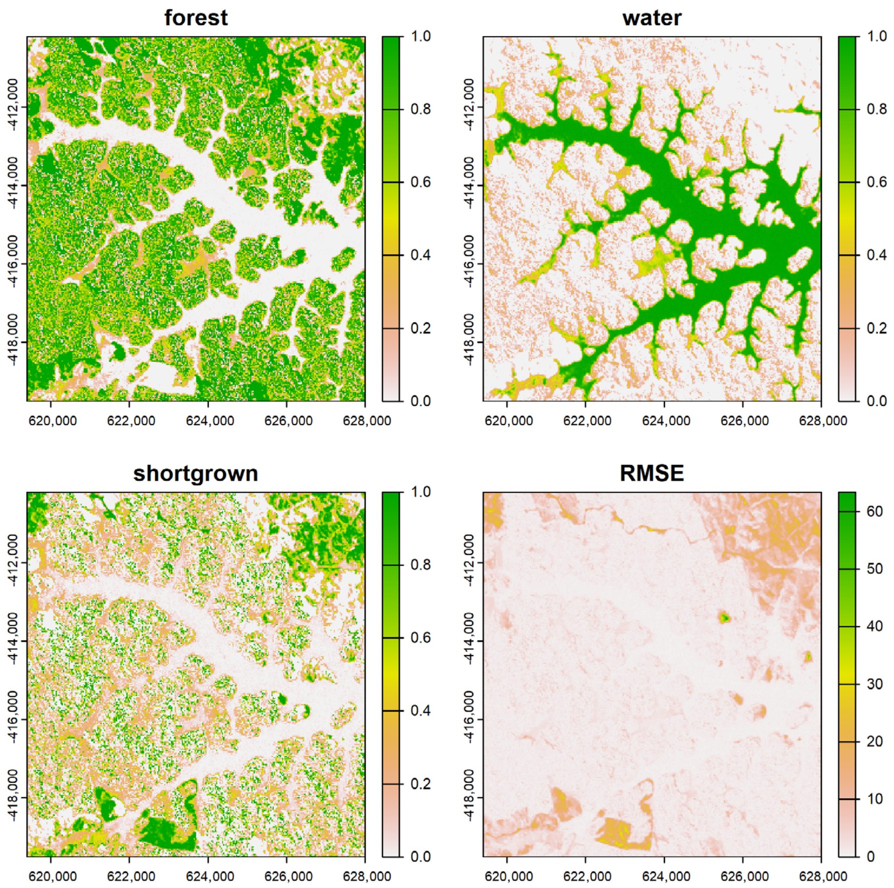

FIGURE 4 Fractions (abundances) per class, computed using mesma() on five endmembers per class. RMSE shows how the non-negative least squares fit varies across the scene.

FUNDING INFORMATION

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

CONFLICT OF INTEREST STATEMENT

The authors declare that there are no conflicts of interest related to the research, authorship or publication of this paper.

PEER REVIEW

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1111/2041-210X.14451.

DATA AVAILABILITY STATEMENT

RSTOOLBOX is published under the GNU General Public Licence >=3. The code is available at GitHub via https://github.com/bleutner/RStoolbox. Furthermore, RSTOOLBOX can be installed via the CRAN repository (https://cran.r-project.org/package=RStoolbox) using R's default package install command. Example data were self-made or collected from scenes freely available online in the Landsat/Sentinel mission archives. It is available directly from the GitHub repository or alternatively via https://doi.org/10.5281/zenodo.13983145 (Müller et al., 2024).

ORCID

Konstantin Müller https://orcid.org/0000-0001-6540-3124
Jakob Schwalb-Willmann https://orcid.org/0000-0003-2665-1509
Martin Wegmann https://orcid.org/0000-0003-0335-9601
Benjamin Leutner https://orcid.org/0000-0002-6893-2002

REFERENCES

Baig, M. H. A., Zhang, L., Shuai, T., & Tong, Q. (2014). Derivation of a tasselled cap transformation based on Landsat 8 at- satellite reflectance. *Remote Sensing Letters*, *5*, 423–431.

Bioucas-Dias, J. M., Plaza, A., Camps-Valls, G., Scheunders, P., Nasrabadi, N., & Chanussot, J. (2013). Hyperspectral remote sensing data analysis and future challenges. *IEEE Geoscience and Remote Sensing Magazine*, *1*, 6–36.

Bivand, R., Keitt, T., & Rowlingson, B. (2006–2023). *rgdal: Bindings for the 'geospatial' data abstraction library*. R package Version 1.6-7. https://CRAN.R-project.org/package=rgdal

Bivand, R. S., Pebesma, E., & Gomez-Rubio, V. (2013). *Applied spatial data analysis with R* (2nd ed.). Springer. https://asdar-book.org/

Cracknell, A. P. (2018). The development of remote sensing in the last 40 years. *International Journal of Remote Sensing*, *39*, 8387–8427. https://doi.org/10.1080/01431161.2018.1550919

CRAN. (2024). *Badges*. https://cranlogs.r-pkg.org/badges/grand-total/RStoolbox

Crist, E. P. (1985). A TM tasseled cap equivalent transformation for reflectance factor data. *Remote Sensing of Environment*, *17*, 301–306. https://www.sciencedirect.com/science/article/pii/0034425785901026

Eddelbuettel, D., & Balamuta, J. J. (2018). Extending R with C++: A brief introduction to Rcpp. *The American Statistician*, *72*, 28–36.

Franc, V., Hlavac, V., & Navara, M. (2005). Sequential coordinate-wise algorithm for the non-negative least squares problem. In A. Gagalowicz & W. Philips (Eds.), *Computer Analysis of Images and Patterns*, (pp. 407–414). Berlin, Heidelberg: Springer Berlin Heidelberg.

Garnier, S., Ross, N., Rudis, R., Camargo, P. A., Sciaini, M., & Scherer, C. (2023). *viridis(Lite)—Colorblind-friendly color maps for R*. ViridisLite package Version 0.4.2. https://sjmgarnier.github.io/viridis/

Grabska, E., Frantz, D., & Ostapowicz, K. (2020). Evaluation of machine learning algorithms for forest stand species mapping using Sentinel-2 imagery and environmental data in the Polish Carpathians. *Remote Sensing of Environment*, *251*, 112103. https://www.sciencedirect.com/science/article/pii/S0034425720304764

Hijmans, R. J. (2023a). *raster: Geographic data analysis and modeling*. R package Version 3.6-23. https://CRAN.R-project.org/package=raster

Hijmans, R. J. (2023b). *terra: Spatial data analysis*. R package Version 1.7-39. https://CRAN.R-project.org/package=terra

Hill, L. L. (2009). *Georeferencing: The geographic associations of information*. Mit Press.

Huang, C., Wylie, B., Yang, L., Homer, C., & Zylstra, G. (2002). Derivation of a tasseled cap transformation based on Landsat 7 at-satellite reflectance. *International Journal of Remote Sensing*, *23*, 1741–1748.

Imaging, D., & Lab, R. S. (2024). *ENVI spectral library format*. SLI. https://dirsig.cis.rit.edu/docs/new/sli.html

ISO. (2012). *ISO/IEC 14882:2011 Information technology—Programming languages—C++*. International Organization for Standardization. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=50372

Ivits, E., Lamb, A., Langar, F., Hemphill, S., & Koch, B. (2008). Orthogonal transformation of segmented SPOT5 images: Seasonal and geographical dependence of the tasselled cap parameters. *Photogrammetric Engineering and Remote Sensing*, *74*, 1351–1364. https://api.semanticscholar.org/CorpusID:6273816

Jia, X. (2006). *Remote sensing digital image analysis: An introduction*. Springer.

Kamusoko, C. (2019). *Image transformation*. Springer. https://doi.org/10.1007/978-981-13-8012-9_3

Kuhn, M. (2008). Building predictive models in R using the caret package. *Journal of Statistical Software*, *28*, 1–26. https://www.jstatsoft.org/index.php/jss/article/view/v028i05

Lemenkova, P., & Debeir, O. (2022). R libraries for remote sensing data classification by k-means clustering and NDVI computation in Congo River Basin, DRC. *Applied Sciences*, *12*, 12554. https://www.mdpi.com/2076-3417/12/24/12554

Leutner, B., Schwalb-Willmann, J., & Müller, K. (2024). *RStoolbox*. https://bleutner.github.io/RStoolbox/

Lobser, S., & Cohen, W. (2007). MODIS tasselled cap: Land cover characteristics expressed through transformed MODIS data. *International Journal of Remote Sensing*, *28*, 5079–5101.

Matias, F. I., Caraza-Harter, M. V., & Endelman, J. B. (2020). FIELDimageR: An R package to analyze orthomosaic images from agricultural field trials. *The Plant Phenome Journal*, *3*, e20005. https://doi.org/10.1002/ppj2.20005

Müller, K., Schwalb-Willmann, J., Wegmann, M., & Leutner, B. (2024). *Rstoolbox*. https://doi.org/10.5281/zenodo.13983145

Muñiz-Castillo, A. I., Rivera-Sosa, A., Chollett, I., Eakin, C. M., Andrade-Gómez, L., McField, M., & Arias-González, J. E. (2019). Three decades of heat stress exposure in Caribbean coral reefs: A new regional delineation to enhance conservation. *Scientific Reports*, *9*, 11013. https://doi.org/10.1038/s41598-019-47307-0

ODM. (2020). *A command line toolkit to generate maps, point clouds, 3D models and DEMs from drone, balloon or kite images*. https://github.com/OpenDroneMap/ODM

Pebesma, E. (2018). Simple features for R: Standardized support for spatial vector data. *The R Journal*, *10*, 439–446. https://doi.org/10.32614/RJ-2018-009

Pebesma, E., & Bivand, R. (2023). *Spatial data science: With applications in R*. Chapman and Hall/CRC. https://r-spatial.org/book/
Pebesma, E. J., & Bivand, R. (2005). Classes and methods for spatial data in R. *R News*, *5*, 9–13. https://CRAN.R-project.org/doc/Rnews/
Prăvălie, R., Sîrodoev, I., Nita, I.-A., Patriche, C., Dumitraşcu, M., Roşca, B., Tişcovschi, A., Bandoc, G., Săvulescu, I., Mănoiu, V., & Birsan, M.-V. (2022). NDVI-based ecological dynamics of forest vegetation and its relationship to climate change in Romania during 1987–2018. *Ecological Indicators*, *136*, 108629. https://www.sciencedirect.com/science/article/pii/S1470160X22001005
Pricope, G. N., Mapes, L. K., & Woodward, D. K. (2019). Remote sensing of human–environment interactions in global change research: A review of advances, challenges and future directions. *Remote Sensing*, *11*, 2783. https://www.mdpi.com/2072-4292/11/23/2783
R Core Team. (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/
Radočaj, D., Obhoaš, J., Jurišić, M., & Gašparović, M. (2020). Global open data remote sensing satellite missions for land monitoring and conservation: A review. *Land*, *9*, 402.
Riaño, D., Chuvieco, E., Salas, J., & Aguado, I. (2003). Assessment of different topographic corrections in landsat-tm data for mapping vegetation types (2003). *IEEE Transactions on Geoscience and Remote Sensing*, *41*, 1056–1061.
Richards, J. (2005). Analysis of remotely sensed data: The formative decades and the future. *IEEE Transactions on Geoscience and Remote Sensing*, *43*, 422–432.
Roberts, D., Gardner, M., Church, R., Ustin, S., Scheer, G., & Green, R. (1998). Mapping chaparral in the Santa Monica mountains using multiple endmember spectral mixture models. *Remote Sensing of Environment*, *65*, 267–279. https://www.sciencedirect.com/science/article/pii/S0034425798000376
Roberts, D. A., Roth, K. L., Wetherley, E. B., Meerdink, S. K., & Perroy, R. L. (2018). Hyperspectral vegetation indices. In P. S. Thenkabail, J. G. Lyon, & A. Huete (Eds.), *Hyperspectral indices and image classifications for agriculture and vegetation* (pp. 3–26). CRC Press.
Roussel, J.-R., & Auty, D. (2024). *Airborne LiDAR data manipulation and visualization for forestry applications*. R package Version 4.0.4. https://cran.r-project.org/package=lidR
Roussel, J.-R., Auty, D., Coops, N. C., Tompalski, P., Goodbody, T. R., Meador, A. S., Bourdon, J.-F., de Boissieu, F., & Achim, A. (2020). lidR: An R package for analysis of Airborne Laser Scanning (ALS) data. *Remote Sensing of Environment*, *251*, 112061. https://www.sciencedirect.com/science/article/pii/S0034425720304314
Schoenert, M., Weichelt, H., Zillmann, E., & Jürgens, C. (2014). Derivation of tasseled cap coefficients for RapidEye data. *Proceedings of SPIE—The International Society for Optical Engineering*, *9245*, 153–163.
Thuillier, G., Hersé, M., Labs, D., Foujols, T., Peetermans, W., Gillotay, D., Simon, P., & Mandel, H. (2003). The solar spectral irradiance from 200 to 2400 nm as measured by the SOLSPEC spectrometer from the atlas and Eureca Missions. *Solar Physics*, *214*, 1–22.
USGS. (2024). *Earth Resources Observation and Science (EROS) Center*. https://www.usgs.gov/centers/eros
Wegmann, M., Leutner, B., & Dech, S. (2016). *Remote sensing and GIS for ecologists: Using open source software*. Pelagic. https://www.lehmanns.de/shop/mathematik-informatik/30041547-9781784270230-remote-sensing-and-gis-for-ecologists
Wegmann, M., Schwab-Willmann, J., & Dech, S. (2020). *An introduction to spatial data analysis: Remote sensing and GIS with open source software*. Pelagic Publishing. https://www.lehmanns.de/shop/mathematik-informatik/50737262-9781784272128-an-introduction-to-spatial-data-analysis
Westoby, M. J., Brasington, J., Glasser, N. F., Hambrey, M. J., & Reynolds, J. M. (2012). 'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, *179*, 300–314.
Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag. https://ggplot2.tidyverse.org
Wickham, H. (2019). *Advanced R* (2nd ed.). CRC Press, Taylor and Francis Group.
Yarbrough, L. D., Easson, G., & Kuszmaul, J. (2005). *Quickbird 2 tasseled cap transform coefficients: A comparison of derivation methods*. Global Priorities in Land Remote Sensing. https://api.semanticscholar.org/CorpusID:55245483
Zhiminaicela-Cabrera, J., Quevedo-Guerrero, J., & Morocho-Castillo, A. (2020). *Deforestación y cambios en la cobertura vegetal del archipiélago de jambelí, mediante el uso de imágenes satelitales Landsat-8*. Revista de Investigación Científica.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

Appendix S1. The supportive information provide files to check the benchmark of our spectralIndices() function implemented in C++ and an example for creating custom spectral indices within this implementation.

How to cite this article: Müller, K., Schwalb-Willmann, J., Wegmann, M., & Leutner, B. (2025). RStoolbox: An R package for remote sensing data analysis. *Methods in Ecology and Evolution*, *00*, 1–10. https://doi.org/10.1111/2041-210X.14451