dtwSat: Time-Weighted Dynamic Time Warping for Satellite Image Time Series Analysis in R[1]

Victor Maus
INPE
University of Munster

Gilberto Camara
INPE
University of Munster

Marius Appel
University of Munster

Edzer Pebesma
University of Munster

Abstract

The opening of large archives of satellite data such as LANDSAT, MODIS and the SENTINELs has given researchers unprecedented access to data, allowing them to better quantify and understand local and global land change. The need to analyse such large data sets has lead to the development of automated and semi-automated methods for satellite image time series analysis. However, few of the proposed methods for remote sensing time series analysis are available as open source software. In this paper we present the R package **dtwSat**. This package provides an implementation of the Time-Weighted Dynamic Time Warping method for land use and land cover mapping using sequence of multi-band satellite images. Methods based on dynamic time warping are flexible to handle irregular sampling and out-of-phase time series, and they have achieved significant results in time series analysis. **dtwSat** is available from the Comprehensive R Archive Network and contributes to making methods for satellite time series analysis available to a larger audience. The package supports the full cycle of land cover classification using image time series, ranging from selecting temporal patterns to visualising and evaluating the results.

Keywords: dynamic programming, MODIS time series, land use changes, crop monitoring.

1. Introduction

Remote sensing images are the most widely used data source for measuring land use and land cover change (LUCC). In many areas, remote sensing images are the only data available for this purpose (Lambin and Linderman 2006; Fritz *et al.* 2013). Recently, the opening of large archives of satellite data such as LANDSAT, MODIS and the SENTINELs has given researchers unprecedented access to data, allowing them to better quantify and understand local and global land change. The need to analyse such large data sets has lead to the development of automated and semi-automated methods for satellite image time series analysis. These methods include multi-image compositing (Griffiths *et al.* 2013), detecting forest disturbance and recovery (Kennedy *et al.* 2010; Zhu *et al.* 2012; DeVries *et al.* 2015), crop classification (Xiao *et al.* 2005; Wardlow *et al.* 2007; Petitjean *et al.* 2012; Maus *et al.* 2016), planted forest mapping (le Maire *et al.* 2014), crop expansion and intensification (Galford

[1]This vignette is based on the paper: MAUS, V.; CAMARA, G.; APPEL, M.; PEBESMA, E. dtwSat: Time-Weighted Dynamic Time Warping for satellite image time series analysis in R. Submitted to the Journal of Statistical Software.

et al. 2008; Sakamoto *et al.* 2009), detecting trend and seasonal changes (Lunetta *et al.* 2006; Verbesselt *et al.* 2010a,b, 2012), and extracting seasonality metrics from satellite time series (Jönsson and Eklundh 2002, 2004). Given the open availability of large image data sets, the research community on Earth Observation would get much benefit from methods that are openly available, reproducible and comparable. However, few of the proposed methods for remote sensing time series analysis are available as open source software, the main exception being the BFAST and BFAST-monitor algorithms for change detection (Verbesselt *et al.* 2010a,b). This paper is a contribution to making methods for satellite time series analysis available to a larger audience.

In this paper we describe the **dtwSat** package, written in R (R Core Team 2016) and Fortran programming languages, and available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=dtwSat`. The package provides an implementation of Time-Weighted Dynamic Time Warping (TWDTW) (Maus *et al.* 2016) for satellite image time series analysis.

The TWDTW method is an adaptation of the well-known dynamic time warping (DTW) method for time series analysis (Velichko and Zagoruyko 1970; Sakoe and Chiba 1971, 1978; Rabiner and Juang 1993; Berndt and Clifford 1994; Keogh and Ratanamahatana 2005; Müller 2007) for land use and land cover classification. The standard DTW compares a temporal signature of a known event (*e.g.*, a person's speech) with an unknown time series. It finds all possible alignments between two time series and provides a dissimilarity measure (Rabiner and Juang 1993). In contrast to standard DTW, the TWDTW method is sensitive to seasonal changes of natural and cultivated vegetation types. It also considers inter-annual climatic and seasonal variability. In a tropical forest area, the method has achieved a high accuracy for mapping classes of single cropping, double cropping, forest, and pasture (Maus *et al.* 2016).

We chose R because it is an open source software that offers a large number of reliable packages. The **dtwSat** package builds upon on a number of graphical and statistical tools in R: **dtw** (Giorgino 2009), **proxy** (Meyer and Buchta 2015), **zoo** (Zeileis and Grothendieck 2005), **mgcv** (Wood 2000, 2003, 2004, 2006, 2011), **sp** (Pebesma and Bivand 2005; Bivand *et al.* 2013), **raster** (Hijmans 2015), **caret** (Kuhn *et al.* 2016), and **ggplot2** (Wickham 2009). Other R packages that are related and useful for remote sensing and land use analysis include **landsat** (Goslee 2011), **rgdal** (Bivand and Lewin-Koh 2015), **spacetime** (Pebesma 2012; Bivand *et al.* 2013), **bfast** (Verbesselt *et al.* 2010a,b), **bfastmonitor** (Verbesselt *et al.* 2011), **bfastSpatial** (Dutrieux and DeVries 2014), **MODISTools** (Tuck *et al.* 2014), **maptools** (Bivand and Lewin-Koh 2015), and **lucc** (Moulds *et al.* 2015). Using existing packages as building blocks, software developers in R save a lot of time and can concentrate on their intended goals.

There is already an R package that implements the standard DTW method for time series analysis: the **dtw** package (Giorgino 2009). In the **dtwSat** package, we focus on the specific case of satellite image time series analysis. The analysis method implemented in **dtwSat** package extends that of the **dtw** package; it adjusts the standard DTW method to account for the seasonality of different types of land cover. Our aim is to support the full cycle of land use and land cover classification, from selecting sample patterns to visualising and evaluating the final result.

This paper focuses on the motivation and guidance for using the TWDTW method for remote sensing applications. The full description of the method is available in a paper published by the lead author (Maus *et al.* 2016). In what follows, Section 3 gives an overview of the **dtwSat**

package. The Section 2 describes the application of TWDTW (Maus *et al.* 2016) for satellite time series analysis. Then, Section 4 focuses on the analysis of a single time series and shows some visualisation methods. We then present an example of a complete land use and land cover change analysis for a study area in the Mato Grosso, Brazil in Section 5.

2. The Time-Weighted Dynamic Time Warping method

In this section, we describe the Time-Weighted Dynamic Time Warping (TWDTW) algorithm in general terms. For a detailed technical explanation, refer to Maus *et al.* (2016). TWDTW is time-constrained version of the Dynamic Time Warping (DTW) algorithm. Although the standard DTW method is good for shape matching (Keogh and Ratanamahatana 2005), it is not suited *per se* for satellite image time series analysis, since it disregards the temporal range when finding the best matches between two time series (Maus *et al.* 2016). When using image time series for land cover classification, one needs to balance between shape matching and temporal alignment, since each land cover class has a distinct phenological cycle associated with the vegetation (Reed *et al.* 1994, Zhang *et al.* (2003)). For example, soybeans and maize cycles range from 90 to 120 days, whereas sugar-cane has a 360 to 720 days cycle. A time series with cycle larger than 180 days is unlikely to come from soybeans or maize. For this reason, Maus *et al.* (2016) include a time constraint in DTW to account for seasonality. The resulting method is capable of distinguishing different land use and land cover classes.

The inputs to TWDTW are: (a) a set of time series of known temporal patterns (*e.g.*, phenological cycles of land cover classes); (b) an unclassified long-term satellite image time series. For each temporal pattern, the algorithm finds all matching subintervals in the long-term time series, providing a dissimilarity measure (cf. Figure 1). The result of the algorithm is a set of subintervals, each associated with a pattern and with a dissimilarity measure. We then break the unclassified time series in periods according to our needs (*e.g.*, yearly, seasonality, monthly). For each period, we consider all matching subintervals that intersect with it, and classify them based on the land cover class of the best matching subinterval. In this way, the long-term satellite time series is divided in periods, and each period is assigned a land cover class.

To use TWDTW for land use and land cover classification, we need the following data sets:

- A set of remote sensing time series for the study area. For example, a tile of a MODIS MOD13Q1 image consists of 4800 x 4800 pixels, covering an area of 10 degrees x 10 degrees at the Equator (Friedl *et al.* 2010). A 15-year (2000-2015) MODIS MOD13Q1 set time series has 23 images per year, with a total of 23 million time series, each with 346 samples.

- A set of time series with land cover information, called *temporal patterns*. Typically, each time series is short and covers one phenological cycle of one land cover type. Examples would be a time series of a soybean crop, or one that describes a mature tropical forest. These temporal patterns can be extracted from the remote sensing image data, if the user knows their spatial and temporal location.

- A set of ground truth points, with spatial and temporal information and land cover classification. These *ground truth* points are used for validation and accuracy assessment.

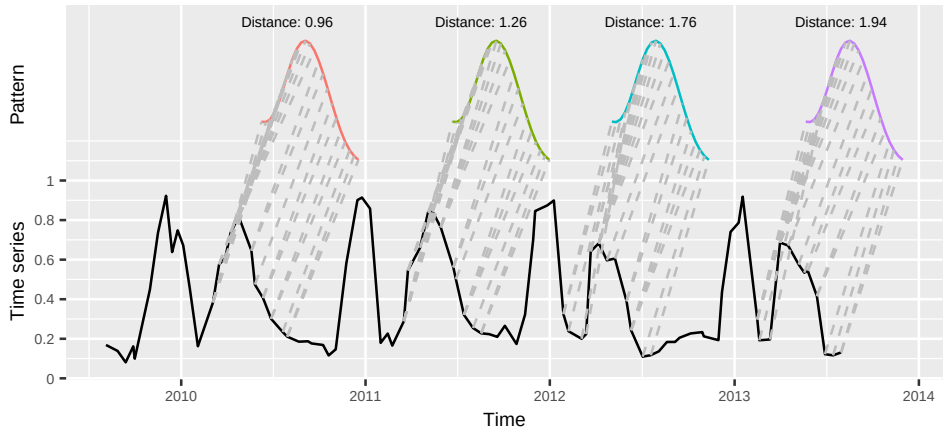

Figure 1: Matches of the known temporal pattern to subintervals of the long-term time series. The solid black line is the long-term time series, the colored lines are the different matches of the same pattern ordered by TWDTW dissimilarity measure, and the gray dashed lines are the matching points.

Based on the information provided by the user about the images to be analysed, our method maps them to a three-dimensional (3-D) array in space-time (Figure 2). This array can have multiple attributes, such as the satellite bands (*e.g.*, "red", "nir", and "blue"), and derived indices (*e.g.*, "NDVI", "EVI", and "EVI2"). This way, each pixel location is associated to a sequence of measurements, building a satellite image time series. Figure 2 shows an example of "evi" time series for a location in the Brazilian Amazon from 2000 to 2008. In the first two years, the area was covered by forest that was cut in 2002. The area was then used for cattle raising (pasture) for three years, and then for crop production from 2006 to 2008. Satellite image time series are thus useful to describe the dynamics of the landscape and the land use trajectories.

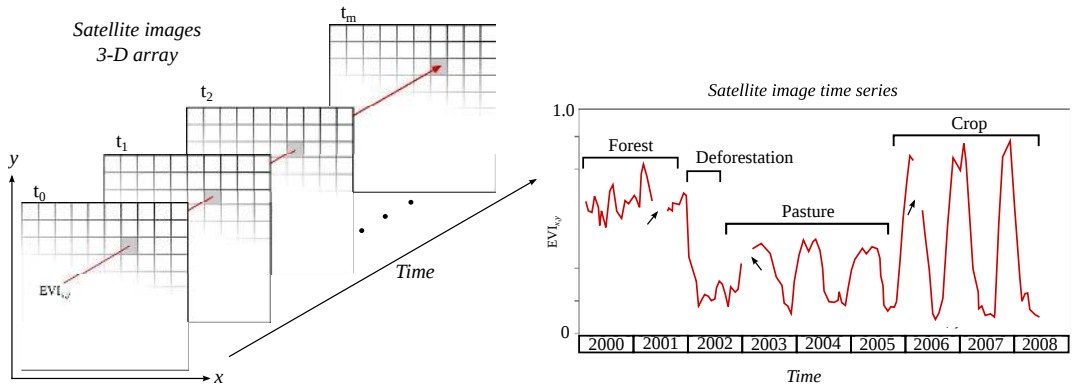

Figure 2: A 3-dimensional array of satellite images (left), an enhanced vegetation index (EVI) time series at the pixel location (x, y) (right). The arrows indicate gaps in the time series. Adapted from Maus *et al.* (2016).

3. dtwSat package overview

dtwSat provides a set of functions for land cover change analysis using satellite image time series. This includes functions to build temporal patterns for land cover types, apply the TWDTW analysis using different weighting functions, visualise the results in a graphical interface, produce land cover maps, and create spatiotemporal plots for land changes. Therefore, **dtwSat** gives an end-to-end solution for satellite time series analysis, which users can make a complete land change analysis.

For the **dtwSat** package, the user should provide the following inputs:

- A set of time ordered satellite images, all with the same spatial extent. The user should also inform the date of each image. In R the images should use the `RasterBrick` or `RasterStack` class of the **raster** package.
- A list of temporal patterns, each associated to a time series in **zoo** format.
- A list of known ground truth points, each with spatial and temporal information, in a format readable in R, such as CSV or shapefile.

The **dtwSat** package organizes the data in three `S4` classes of objects: `twdtwTimeSeries`, `twdtwMatches`, and `twdtwRaster`. To store time series we use the class `twdtwTimeSeries`. The objects of class `twdtwTimeSeries` have two slots; the slot called `timeseries` has a list of `zoo` objects; and the slot called `labels` stores the labels of the time series. The class `twdtwMatches` has 3 slots to store inputs and results of the TWDTW analysis. The slots called `timeseries` and `patterns` are objects of the class `twdtwTimeSeries` with the unclassified long-term time series and the temporal patterns, respectively. A third slot called `alignments` has a `list` with detailed information about the matches between the patterns and the unclassified long-term time series. The classes `twdtwTimeSeries` and `twdtwMatches` are used to analyse lists of time series.

The class `twdtwRaster` is used for satellite image time series. This class can store either unclassified raster time series with the satellite raw data, the results of the TWDTW analyis, or a classified raster time series. In both cases, the objects of class `twdtwRaster` have five slots. The slot called `timeseries` is a list of `RasterBrick` or `RasterStack` objects with time ordered satellite images (all with the same temporal and spatial extents); the slot called `timeline` is a vector of class `Date` with dates of the satellite images; the slot called `layers` has the names of satellite bands; the slot called `levels` has levels for the raster values; and the slot called `labels` has labels for the raster values. This class builds upon the R package **raster** to build a multi-attribute 3-D raster in space-time, allowing for multi-band satellite image time series analysis.

4. Classifying a time series

This section describes how to classify one time series, using examples that come with the **dtwSat** package. We will show how to match three temporal patterns ("soybean", "cotton", and "maize") to subintervals of a long-term satellite image time series. These time series have been extracted from a set of MODIS MOD13Q1 (Friedl *et al.* 2010) images and include the

vegetation indices "ndvi", "evi", and the original bands "nir", "red", "blue", and "mir". In this example, the classification of crop types for the long-term time series is known.

4.1. Input data

The inputs for the next examples are time series in **zoo** format. The first is an object of class `zoo` with a long-term time series, referred to as `example_ts`, and the second is a `list` of time series of class `zoo` with the temporal patterns of "soybean", "cotton", and "maize", referred to as `patterns.list`.

From `zoo` objects we construct time series of class `twdtwTimeSeries`, for which we have a set of visualization and analysis methods implemented in the **dtwSat** package. The code below builds two objects of class `twdtwTimeSeries`. The first has the long-term time series and second has the temporal patterns. We use the plot method types `timeseries` and `patterns` to shown the objects `ts` in Figure 3 and `patterns_ts` in Figure 4, respectively. This plot method uses `ggplot` syntax.

```
ts = twdtwTimeSeries(example_ts, labels="Time series")
patterns_ts = twdtwTimeSeries(patterns.list)
example_ts_labels
```

```
    label       from         to
1 Soybean 2009-09-01 2010-03-01
2  Cotton 2010-03-01 2010-09-01
3 Soybean 2010-09-01 2011-03-01
4  Cotton 2011-03-01 2011-09-01
5 Soybean 2011-09-01 2012-03-01
6   Maize 2012-03-01 2012-09-01
7 Soybean 2012-09-01 2013-03-01
8   Maize 2013-03-01 2013-09-01
```

```
library(dtwSat)
names(patterns.list)
```

```
[1] "Soybean" "Cotton"  "Maize"
```

```
head(example_ts, n = 2)
```

```
             ndvi    evi    red    nir   blue    mir
2009-08-05 0.3169 0.1687 0.1167 0.2250 0.0427 0.2193
2009-08-28 0.2609 0.1385 0.1168 0.1993 0.0548 0.2657
```

```
plot(ts, type = "timeseries") +
  annotate(geom = "text", x = example_ts_labels$from+90, y = 0.98,
  label = example_ts_labels$label, size = 2)


plot(patterns_ts, type = "patterns")
```

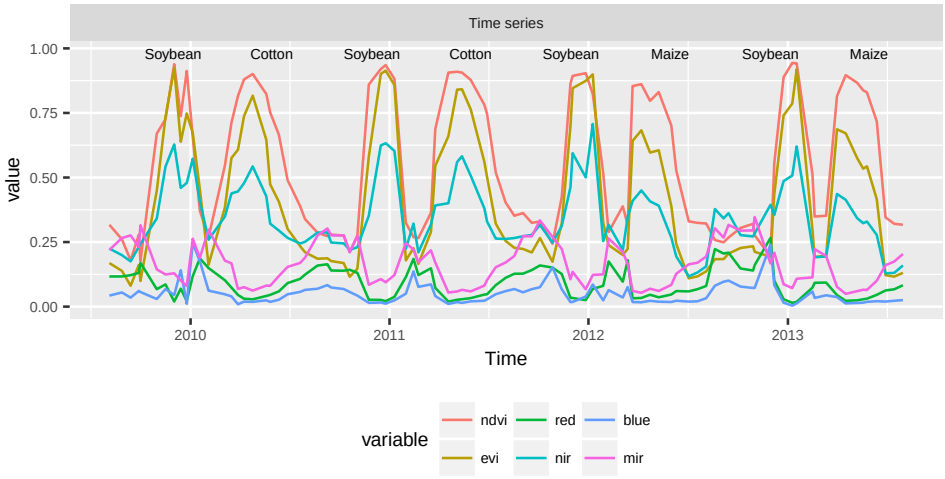

Figure 3: Example of time series based on MODIS product MOD13Q1 (Friedl *et al.* 2010). The labels of the phenological cycle are shown in the plot.

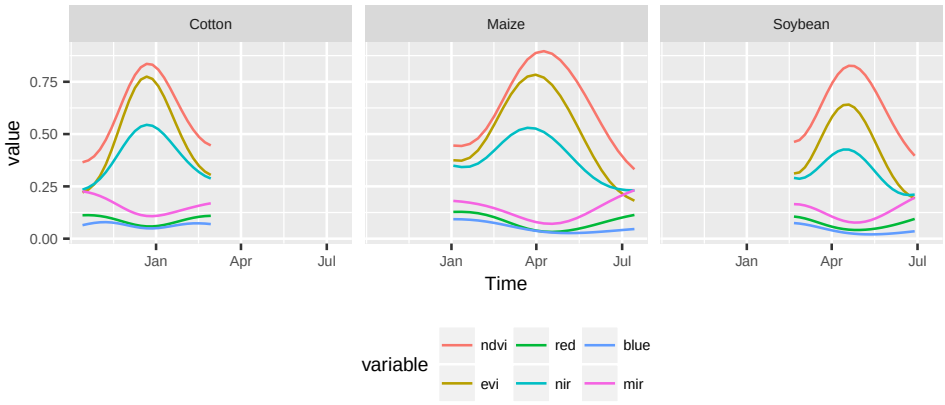

Figure 4: Temporal patterns of soybean, cotton, and maize based on MODIS product MOD13Q1 (Friedl *et al.* 2010).

TWDTW uses both amplitude and phase information to classify the phenological cycles in the long-term time series. The EVI peak of the "soybean" time series has a similar amplitude as that of "cotton". However, the "soybean" series peaks in late December while the "cotton" series peaks in early April. The EVI peak of the "maize" time series is at the same period as the peak of "cotton". However, the "maize" time series has smaller amplitude than the "cotton" one. Therefore, we can improve the time series classification by combining shape and time information.

4.2. Detection of time series patterns with TWDTW

Each subinterval of the long-term time series in `ts` has a known phenological cycle. We will now compare the known information with the result of the TWDTW algorithm. We use the function `twdtwApply` that returns an R object of class `twdtwMatches` with all matches of each temporal pattern in the time series.

```
log_weight = logisticWeight(alpha = -0.1, beta = 100)
matches =
  twdtwApply(x = ts, y = patterns_ts, weight.fun = log_weight, keep=TRUE)
slotNames(matches)
```

```
[1] "timeseries" "patterns"   "alignments"
```

```
show(matches)
```

```
An object of class "twdtwMatches"
Number of time series: 1
Number of Alignments: 16
Patterns labels: Soybean Cotton Maize
```

To retrieve the complete information of the matches we set `keep=TRUE`. We need this information for the plot methods of the class `twdtwMatches`. The argument `weight.fun` defines the time-weight to the dynamic time warping analysis (Maus *et al.* 2016). By default the time-weight is zero, meaning that the function will run a standard dynamic time warping analysis. The arguments `x` and `y` are objects of class `twdtwTimeSeries` with the unclassified long-term time series and the temporal patterns, respectively. For details and other arguments see `?twdtwApply`.

In our example we use a logistic weight function for the temporal constraint of the TWDTW algorithm. This function is defined by `logisticWeight`. The **dtwSat** package provides two in-built functions: `linearWeight` and `logisticWeight`. The `linearWeight` function with slope `a` and intercept `b` is given by

$$\omega = a \cdot g(t_1, t_2) + b,$$

and the `logisticWeight` with midpoint `beta`, and steepness `alpha`, given by

$$\omega = \frac{1}{1 + e^{-\alpha(g(t_1,t_2)-\beta)}}.$$

The function g is the absolute difference in days between two dates, t_1 and t_2. The linear function creates a strong time constraint even for small time differences. The logistic function has a low weight for small time warps and significant costs for bigger time warps, cf. Figure 5. In our previous studies (Maus *et al.* 2016) the logistic-weight had better results than the linear-weight for land cover classification. Users can define different weight functions as temporal constraints in the argument `weight.fun` of the `twdtwApply` method.

4.3. Visualising the result of the TWDTW algorithm

dtwSat provides five ways to visualise objects of class `twdtwMatches` through the plot types: `matches`, `alignments`, `classification`, `path`, and `cost`. The plot type `matches` shows the matching points of the patterns in the long-term time series; the plot type `alignments` shows the alignments and dissimilarity measures; the plot type `path` shows the low cost paths in the TWDTW cost matrix; and the plot type `cost` allows the visualisation of the cost matrices (local cost, accumulated cost, and time cost); and the plot type `classification` shows the

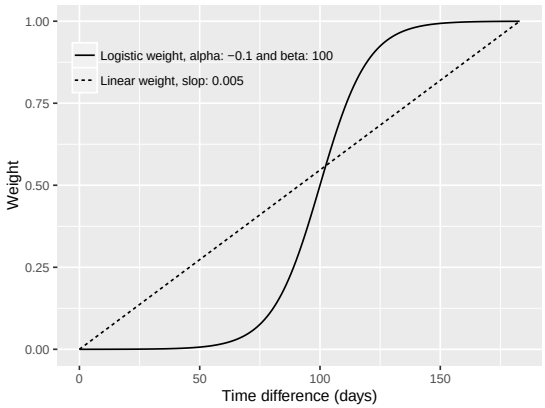

Figure 5: Logistic time-weight function `logisticWeight` with steepness `alpha=-0.1` and midpoint `beta=100`. The x axis shows the absolute difference between two dates in days and the y axis shows the time-weight (Maus *et al.* 2016).

classification of the long-term time series based on the TWDTW analysis. The plot methods for class `twdtwMatches` return a `ggplot` object, so that users can further manipulate the result using the **ggplot2** package. For more details on visualisation functions, please refer to the **dtwSat** documentation in the CRAN (Maus 2015).

We now describe the plot types `matches` and `alignments`. The code bellow shows how to visualise the matching points of the four best matches of "soybean" pattern in the long-term time series, cf. Figure 6.

```
plot(matches, type="matches", patterns.labels="Soybean", k=4)
```

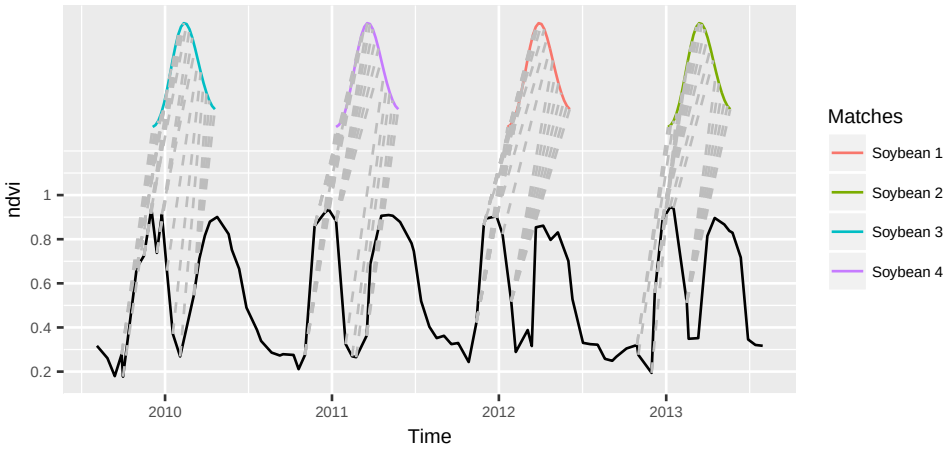

Figure 6: The four best matches of the "soybean" pattern in the time series using a logistic time-weight. The solid black line is the long-term time series; the coloured lines are the temporal patterns; and the grey dashed lines are the respective matching points.

The next example (Figure 7) uses the plot type `alignments` to show the alignments of the temporal patterns. We set the threshold for the dissimilarity measure to be lower than 3.0.

This is useful to display the different subintervals of the long-term time series that have at least one alignment whose dissimilarity is less than the specified threshold.

```
plot(matches, type="alignments", attr = "evi", threshold = 3.0)
```

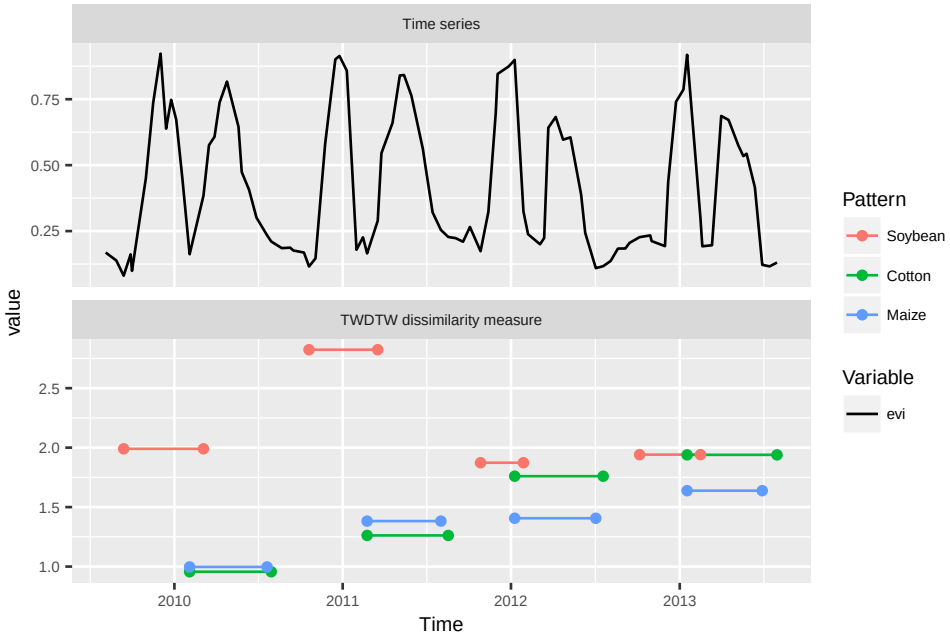

Figure 7: Alignments and dissimilarity measures of the patterns ”soybean”, ”cotton”, and ”maize” to the subintervals of the long-term time series using a logistic time-weight. The solid black line is the EVI time series, and the coloured lines are the alignments of the patterns that have dissimilarity measure lower than three.

4.4. Classifying the long-term time series

Using the matches and their associated dissimilarity measures, we can classify the subintervals of the long-term time series using `twdtwClassify`. To do this, we need to define a period for classification and the minimum overlap between the period and the alignments that intersect with it. We use the plot type `classification` to show the classification of the subintervals of the long-term time series based on the TWDTW analysis. For this example, we set classification periods of 6 months from September 2009 to September 2013, and a minimum overlap of 50% between the alignment and the classification period. This means that at least 50% of the alignment has to be contained inside of the classification period.

```
ts_classification = twdtwClassify(x = matches,
  from = as.Date("2009-09-01"), to = as.Date("2013-09-01"),
  by = "6 month", overlap = 0.5)
plot(ts_classification, type="classification")
```

Comparing the results of the classified time series in Figure 8 with the crop cycles in Figure 3 we see that the algorithm has classified correctly all the eight subintervals from 2009 to

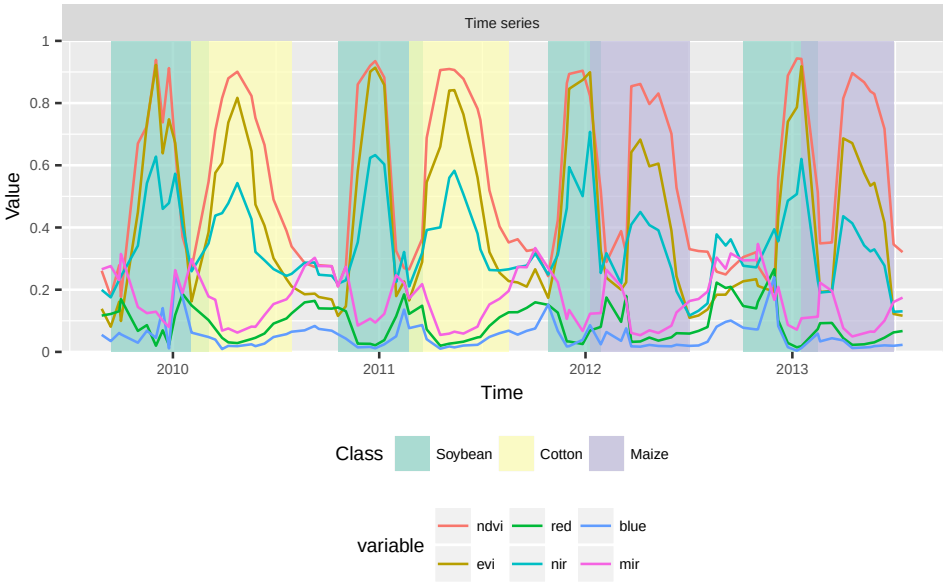

Figure 8: Classification of each 6 months periods of the time series using results of the TWDTW analysis with logistic time-weight. The solid lines are the attributes of the time series, the background colours indicate the classification of the periods.

2013, which are, respectively: “Soybean”, “Cotton”, “Soybean”, “Cotton”, “Soybean”, “Maize”, “Soybean”, “Maize”.

5. Producing a land cover map

In this section we present an application of TWDTW for land use and land cover change analysis using satellite image time series. Our input is a set of images, each covering the same geographical area at different times. Each pixel location is then associated to an unclassified satellite image time series. We assume to have done field work in the area; for some pixel locations and time periods, we know what is the land cover. We then will show how to obtain a set of template patterns, based on the field samples and how to apply these patterns to land cover classification of the set of images. In the end of this section we show how to perform land cover change analysis and how to do accuracy assessment. The satellite images and the field samples used in the examples come with **dtwSat** package.

Our method is not restricted to cases where the temporal patterns are obtained from the set of images. The patterns for the TWDTW analysis can be any time series with same bands or indices as the unclassified images, such as in the examples of Section 4 above.

5.1. Input data

The inputs are: *a)* the satellite images for a given geographical area, organised as a set of georeferenced raster files in GeoTIFF format, each containing all time steps of a spectral band or index; and *b)* a set of ground truth samples. The satellite images are extracted from the MODIS product MOD13Q1 collection 5 (Friedl *et al.* 2010) and include vegetation indexes “ndvi”, “evi”, and original bands “nir”, “red”, “blue”, and “mir”. This product has 250 x 250

m spatial and 16 day temporal resolution.

The region is a tropical forest area in Mato Grosso, Brazil of approximately 5300 km^2 with images from 2007 to 2013 (Figure 9). This is a sequence of 160 images with 999 pixels each for 6 years. We also have a set of 603 ground truth samples of the following classes: "forest", "cotton-fallow", "soybean-cotton", "soybean-maize", and "soybean-millet".

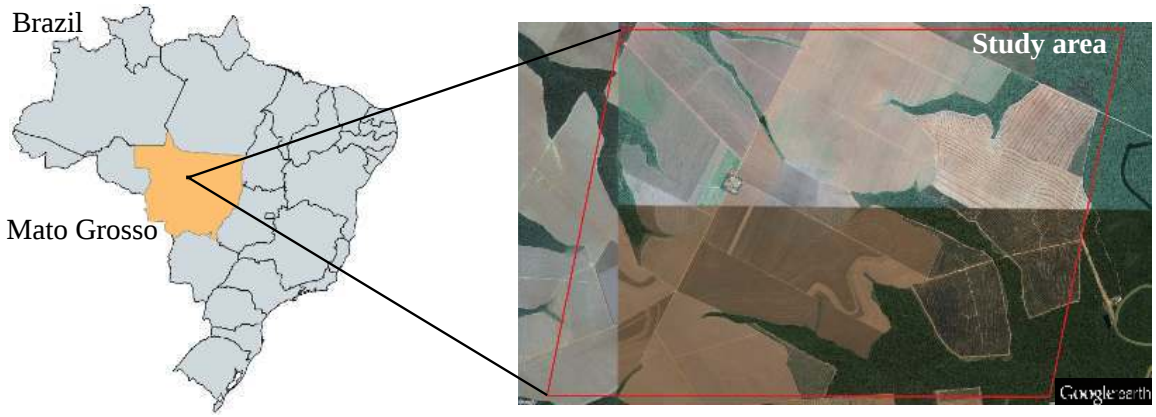

Figure 9: Study area in Mato Grosso, Brazil, shown in a © Google Earth image. The area was originally covered by tropical forest that has been removed for agricultural use.

The data files for the examples that follow are in the **dtwSat** installation folder *lucc_MT/data/*. The *tif* files include the time series of "blue", "red", "nir", "mir", "evi", "ndvi", and "doy" (day of the year); the text file *timeline* has the dates of the satellite images; the CSV file *samples.csv* has the `longitude`, `latitude`, `from`, `to`, and `label` for each field sample; and the text file *samples_projection* contains information about the cartographic projection of the samples, in the format of coordinate reference system used by `sp::CRS`.

```
data_folder = system.file("lucc_MT/data", package = "dtwSat")
dir(data_folder)
```

```
 [1] "blue.tif"           "doy.tif"            "evi.tif"
 [4] "mir.tif"            "ndvi.tif"           "nir.tif"
 [7] "red.tif"            "samples_projection" "samples.csv"
[10] "timeline"
```

In this example, we have stored all the time series for each band in one single file. In this way, we can use the function `raster::brick` to read the satellite images. The algorithm also works when the time steps for each band are split in many files. In this case, the user should call the function `raster::stack` with the appropriate parameters. Because of processing performance, we suggest that interested users group their images in bricks and follow the procedures given below.

```
blue = brick(paste(data_folder,"blue.tif", sep = "/"))
red  = brick(paste(data_folder,"red.tif", sep = "/"))
```

```
nir  = brick(paste(data_folder,"nir.tif", sep = "/"))
mir  = brick(paste(data_folder,"mir.tif", sep = "/"))
evi  = brick(paste(data_folder,"evi.tif", sep = "/"))
ndvi = brick(paste(data_folder,"ndvi.tif", sep = "/"))
day_of_year  = brick(paste(data_folder,"doy.tif", sep = "/"))
dates = scan(paste(data_folder,"timeline", sep = "/"), what = "dates")
```

The set of ground truth samples in the CSV file *samples.csv* has a total of 603 samples divided in five classes: 68 "cotton-fallow", 138 "forest", 79 "soybean-cotton", 134 "soybean-maize", and 184 "soybean-millet". Reading this CSV file, we get a `data.frame` object, with the spatial location (`latitude` and `longitude`), starting and ending dates (`from` and `to`), and the `label` for each sample.

```
field_samples = read.csv(paste(data_folder,"samples.csv", sep = "/"))
head(field_samples, 2)
```

```
  longitude  latitude       from         to         label
1 -55.98819 -12.03646 2011-09-01 2012-09-01 Cotton-fallow
2 -55.99118 -12.04062 2011-09-01 2012-09-01 Cotton-fallow
```

```
table(field_samples[["label"]])
```

```
 Cotton-fallow         Forest Soybean-cotton  Soybean-maize Soybean-millet
            68            138             79            134            184
```

```
proj_str = scan(paste(data_folder,"samples_projection", sep = "/"),
  what = "character")
proj_str
```

```
[1] "+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"
```

5.2. Creating the time series and the temporal patterns

After reading our data, we need to create the time series for analysis. For this purpose, **dtwSat** provides the constructor `twdtwRaster` that builds a multi-band satellite image time series. The inputs of this function are `RasterBrick` objects with the same temporal and spatial extents, and a vector (`timeline`) with the acquisition dates of the images in the format `"YYYY-MM-DD"`. The argument `doy` is optional. If `doy` is not declared, the function builds a `RasterBrick` object using the dates given by `timeline`. This function produces an object of class `twdtwRaster` with the time series of multiple satellite bands.

```
raster_timeseries = twdtwRaster(blue, red, nir, mir, evi, ndvi,
  timeline = dates, doy = day_of_year)
```

We now need to identify the temporal patterns. Usually, this can be done using the collected field samples. In the next example we use the function `getTimeSeries` to get the time series of each field sample from our raster time series. The arguments of the function are a set of raster time series, a `data.frame` with spatial and temporal information about the fields samples (as in the object `field_samples` given above), and a `proj4string` with the projection information. The projection should follow the `sp::CRS` format. The result is an object of class `twdtwTimeSeries` with one time series for each field sample.

```
field_samples_ts = getTimeSeries(raster_timeseries,
  y = field_samples, proj4string = proj_str)
field_samples_ts
```

```
An object of class "twdtwTimeSeries"
Slot "timeseries" length: 603
Slot "labels": [1] Cotton-fallow Cotton-fallow Cotton-fallow
5 Levels: Cotton-fallow Forest Soybean-cotton ... Soybean-millet
```

After obtaining the time series associated to the field samples, we need to create the template patterns for each class. For this purpose, **dtwSat** provides the function `createPatterns`. This function fits a Generalized Additive Model (GAM) [Hastie:1986,Wood:2011] to the field samples and retrieves a smoothed temporal pattern for each band (*e.g.*, "blue", "red", "nir", "mir", "evi", and "ndvi"). We use the GAM because of its flexibility for non-parametric fits, with less rigorous assumptions on the relationship between response and predictor. This potentially provides better fit to satellite data than purely parametric models, due to the data's inter- and intra-annual variability.

To produce the set of template patterns using the function `createPatterns`, we need to set the temporal frequency of the resulting patterns and the smoothing function for the GAM model. In the example below, we set `freq=8` to get temporal patterns with a frequency of 8 days. We also set the GAM smoothing formula to be `formula = y ~ s(x)`, where function `s` sets up a spline model, with `x` the time and `y` a satellite band (for details see `?mgcv::gam` and `?mgcv::s`).

```
temporal_patterns =
  createPatterns(field_samples_ts, freq = 8, formula = y ~ s(x))
```

We use the plot method `type="patterns"` to show the results of the `createPatterns` in Figure 10.

```
plot(temporal_patterns, type = "patterns") +
  theme(legend.position = c(.8,.25))
```

After obtaining the template patterns for each land cover class, it is useful to perform a pre-classification analysis to assess their quality and their informational content. Ideally, one would need template patterns that, when applied to the set of unknown time series, produce consistent results. For this reason, it is advisable that the user performs a pre-classification step, along the lines of the individual analysis described in Section 4. In this way, the users would assess how good their patterns are before classifying a large data set.

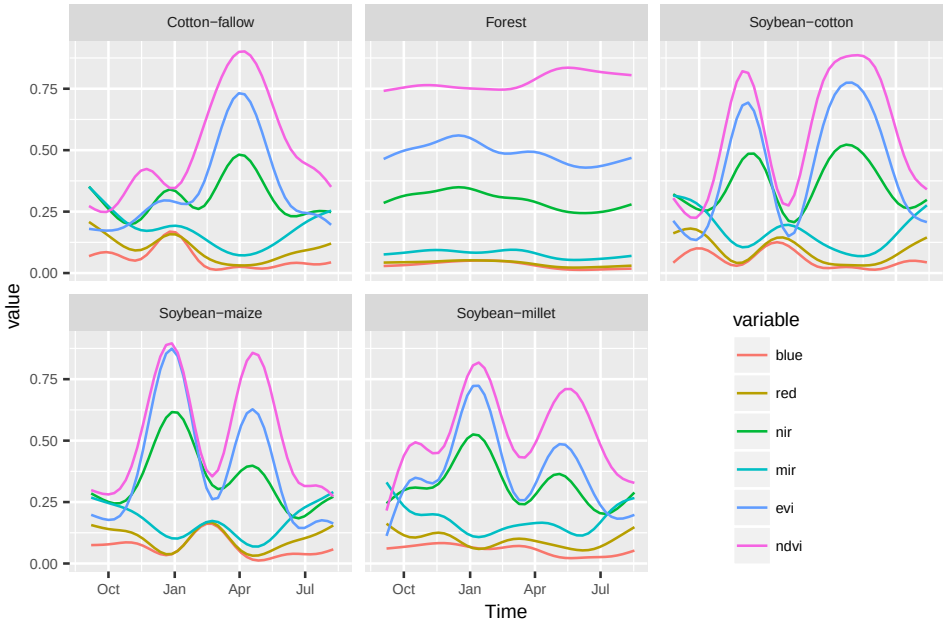

Figure 10: Temporal patterns of forest, cotton-fallow, soybean-cotton, soybean-maize, and soybean-millet based on the ground truth samples.

5.3. Classifying the image time series

After obtaining a consistent set of temporal patterns, we use the function `twdtwApply` to run the TWDTW analysis for each pixel location in the raster time series. The input raster time series in the object `twdtwRaster` should be longer or have approximatly the same length as the temporal patterns. This function retrieves an object of class `twdtwRaster` with the TWDTW dissimilarity measure of the temporal patterns for each time period. The arguments `overwrite` and `format` are passed to `raster::writeRaster`. The arguments `weight.fun` and `overlap` are described in Section 4. Here we set the parameters of the time weight (logistic function) base on our the experience about the phenological cycle of the vegetation in the study area. In the next example, we classify the raster time series using the temporal patterns in `temporal_patterns` obtained as described above. The result is a `twdtwRaster` with five layers; each layer contains the TWDTW dissimilarity measure for one temporal pattern over time. We use the plot type `distance` to illustrate the TWDTW dissimilarity for each temporal pattern in 2013, cf. Figure 11.

```
log_fun = logisticWeight(alpha=-0.1, beta=50)
twdtw_dist = twdtwApply(x = raster_timeseries, y = temporal_patterns,
  overlap = 0.5, weight.fun = log_fun, overwrite=TRUE, format="GTiff")
```

```
[1] "Procesing chunk 1/1"
```

```
plot(x = twdtw_dist, type="distance", time.levels = 6)
```

The results of the example above can be used to create categorical land cover maps. The function `twdtwClassify` selects the most similar pattern for each time period and retrieves

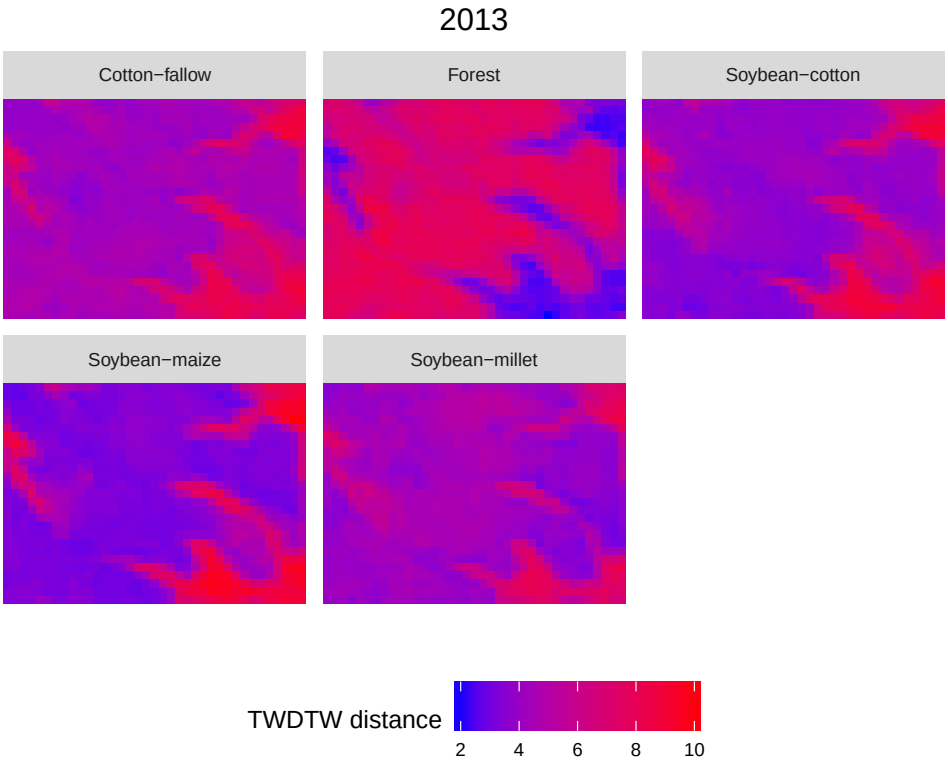

Figure 11: Illustration of the TWDTW dissimilarity from each temporal pattern in 2013. The blue areas are more similar to the pattern and the red areas are less similar to the pattern.

a `twdtwRaster` object with the time series of land use maps. The resulting object includes two layers, the first has the classified categorical maps and the second has the TWDTW dissimilarity measure.

```
land_use_maps = twdtwClassify(twdtw_dist, format="GTiff", overwrite=TRUE)
```

5.4. Looking at the classification results

The classification results can be visualised using the `plot` methods of the class `twdtwRaster`, which supports four plot types: “maps”, “area”, “changes”, and “distance”. The `type="maps"` shows the land cover classification maps for each period, cf. Figure 12.

```
plot(x = land_use_maps, type = "maps")
```

The next example shows the accumulated area for each class over time, using `type="area"`, cf. Figure 13.

```
plot(x = land_use_maps, type = "area")
```

Users can also view the land cover transition for each time period, by setting `type="changes"`. For each land cover class and each period, the plot shows gains and losses in area from the other classes. This is the visual equivalent of a land transition matrix, cf. Figure 14.

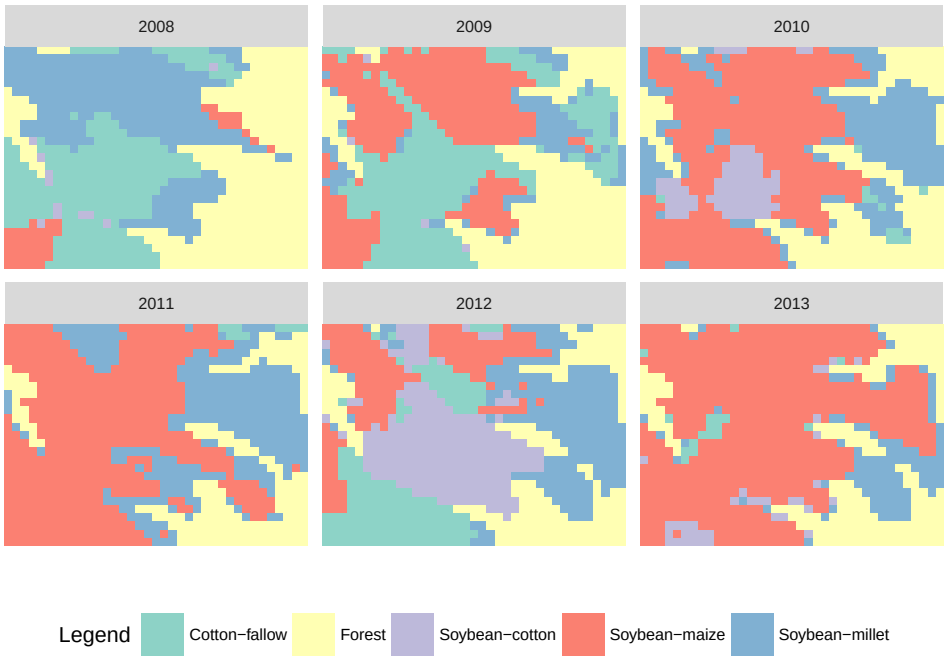

Figure 12: Land use maps for each year from 2008 to 2013.

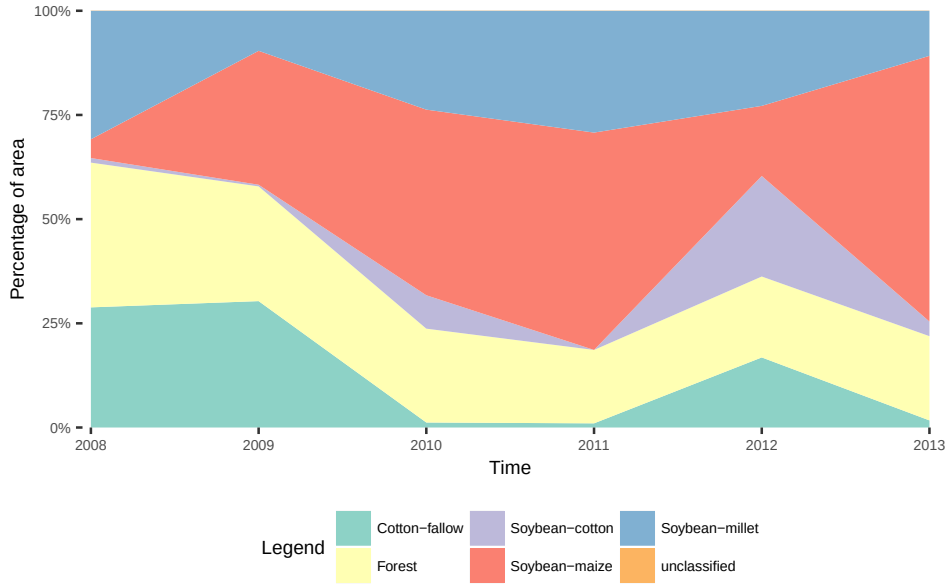

Figure 13: Percentage of area for each land use class from 2008 to 2013.

```
plot(x = land_use_maps, type = "changes")
```

We can also look at the dissimilarity of each classified pixel setting `type="distance"`. This plot can give a measure of the uncertainty of the classification of each pixel for each time period, cf. Figure 15.

```
plot(x = land_use_maps, type="distance")
```

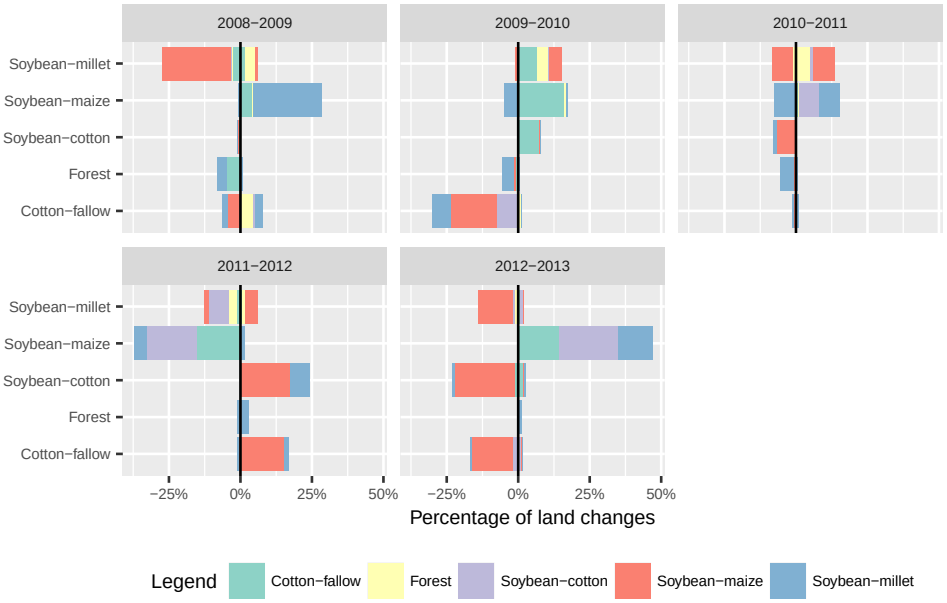

Figure 14: Gains and losses in area from the other classes. The y axis shows the actual class; the positive direction of x axis shows the gains and the negative direction of x axis shows the losses of the classes indicated in y. The colors indicate from/to which classes the gains/losses belong.

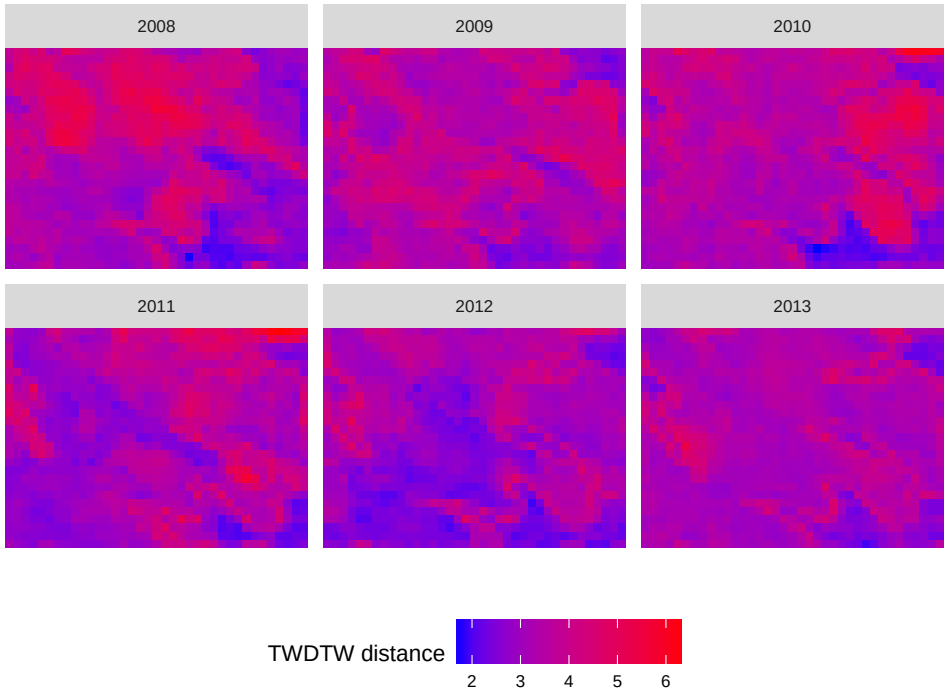

Figure 15: TWDTW dissimilarity measure for each pixel over each classified period. The blue areas have high confidence and the red areas have low confidence in the classification.

5.5. Assessing the classification accuracy

In this section we show how to assess the accuracy of the TWDTW method for land cover classification. To do this, we split the ground truth samples into training and validation sets, using the function `splitDataset` from the package **dtwSat**. This function splits set of time series in the object `twdtwTimeSeries` for training and validation. The argument `p` defines the percentage used for training and the argument `times` gives the number of different partitions to create. This is a a stratified sampling with a simple random sampling within each stratum, see `?createDataPartition` for details. In the next example we create 100 different partitions of the data. Each partition uses 10% of the data for training and 90% for validation. The output is a list with 100 different data partitions; each partition has the temporal patterns based on the training samples and a set of time series for validation.

```
set.seed(1)
partitions = splitDataset(field_samples_ts, p=0.1, times=100,
  freq = 8, formula = y ~ s(x, bs="cc"))
```

For each data partition we run the TWDTW analysis to classify the set of validation time series using the trained temporal patterns. The result is a list of `twdtwMatches` objects with the classified set of time series for each data partition. To compute the *User's Accuracy* (UA) and *Producer's Accuracy* (PA) of the classified time series we use the function `dtwSat::twdtwAssess` that retrieves a `data.frame` with the accuracy assessment for all data partitions.

```
log_fun = logisticWeight(alpha=-0.1, beta=50)
twdtw_res = lapply(partitions, function(x){
  res = twdtwApply(x = x$ts, y = x$patterns, weight.fun = log_fun, n=1)
  twdtwClassify(x = res)
})
assessment = twdtwAssess(twdtw_res)
head(assessment, 5)
```

Figure 16 shows the average μ and standard deviation σ of *user's* and *producer's accuracy* based on a bootstrap simulation of 100 different data partitions using resampling-with-replacement. The *user's accuracy* gives the confidence and the *producer's accuracy* gives the sensitivity of the method for each class. In our analysis all classes had high *user's* and *producer's accuracy*, meaning that TWDTW has high confidence and sensitivity for the classes included in the analysis. The average, standard deviation, and the 99% confidence interval is also shown in Table 1.

6. Conclusions and Discussion

Nowadays, there are large open archives of Earth Observation data, but few open source methods for analysing them. With this motivation, this paper provides guidance on how to use the Time-Weighed Dynamic Time Warping (TWDTW) method for remote sensing

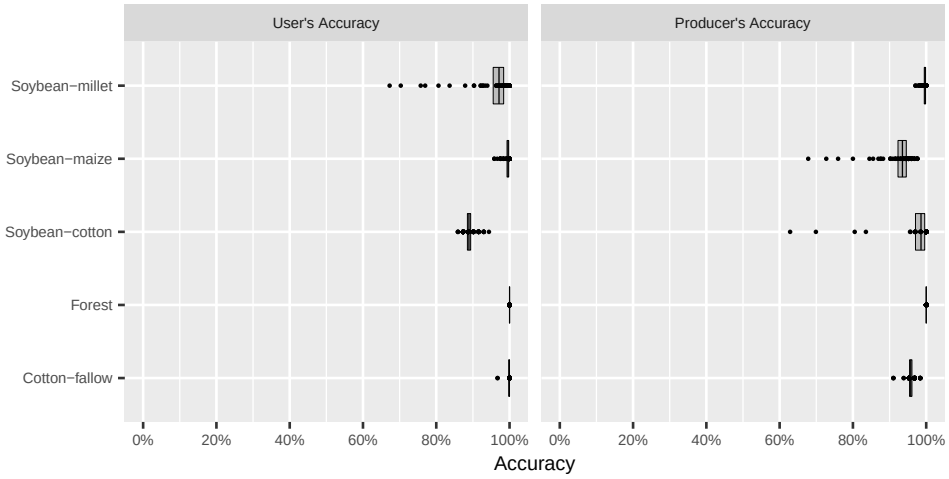

Figure 16: User's Accuracy (UA) and Producer's Accuracy (PA) of the TWDTW method for land cover classification. The plot shows the averages and their confidence interval for 99%.

Class	User's Accuracy (UA) %			Producer's Accuracy (PA)%		
	μ	σ	CI	μ	σ	CI
Cotton-fallow	99.93	(0.46)	[99.80-100.00]	95.78	(1.29)	[95.42-96.08]
Forest	100.00	(0.00)	[100.00-100.00]	100.00	(0.00)	[100.00-100.00]
Soybean-cotton	88.92	(1.81)	[88.46-89.35]	98.58	(5.37)	[97.21-99.60]
Soybean-maize	99.58	(0.95)	[99.35-99.79]	93.51	(4.79)	[92.13-94.56]
Soybean-millet	97.12	(6.09)	[95.50-98.44]	99.67	(0.70)	[99.48-99.84]

Table 1: User's and Producer's Accuracy of the land use classification based on TWDTW analysis. μ is the average accuracy, σ the standard deviation, and CI is the confidence interval of 99% using 100 resampling-with-replacement.

applications. As we have discussed in a companion paper (Maus *et al.* 2016), the TWDTW method is well suited for land cover change analysis of satellite image time series.

The main goal of **dtwSat** package is to make TWDTW accessible for researchers. The package supports the full cycle of land cover classification using image time series, ranging from selecting temporal patterns to visualising and evaluating the results. The current version of the **dtwSat** package provides a pixel-based time series classification method. We envisage that future versions of the package could include local neighborhoods to reduce border effects and improve classification homogeneity.

The **dtwSat** package provides two in-built functions for linear and logistic time weight. In the current version of the package the parameters of the weight functions are set manually to the same value for all land use/cover classes. Future versions of the package could include methods to search for the best parameters to be set class-by-class using field data.

To aim for maximum usage by the scientific community, the **dtwSat** package described in this paper works with well-known R data classes such as provided by packages **zoo** and **raster**. We are planning improvements, so that **dtwSat** can be combined with array databases, such as SciDB (Stonebraker *et al.* 2013). We believe that combining array databases with image time series analysis software such as presented here is one way forward to scaling the process

of information extracting to very large Earth Observation data.

Acknowledgments

Victor Maus has been supported by the Institute for Geoinformatics, University of Münster (Germany), and by the Earth System Science Center, National Institute for Space Research (Brazil). Part of the research was developed in the Young Scientists Summer Program at the International Institute for Applied Systems Analysis, Laxenburg (Austria). Gilberto Camara's term as Brazil Chair at IFGI has been supported by CAPES (grant 23038.007569/2012-16). Gilberto's work is also supported by FAPESP e-science program (grant 2014-08398-6) and CNPq (grant 312151/2014-4).

References

Berndt DJ, Clifford J (1994). "Using Dynamic Time Warping to Find Patterns in Time Series." In UM Fayyad, R Uthurusamy (eds.), *KDD Workshop*, pp. 359–370. AAAI Press. ISBN 0-929280-73-3.

Bivand R, Lewin-Koh N (2015). ***maptools**: Tools for Reading and Handling Spatial Objects*. R package version 0.8-37, URL http://CRAN.R-project.org/package=maptools.

Bivand RS, Pebesma E, Gomez-Rubio V (2013). *Applied Spatial Data Analysis With R, Second edition*. Springer-Verlag, New York. URL http://www.asdar-book.org/.

DeVries B, Verbesselt J, Kooistra L, Herold M (2015). "Robust Monitoring of Small-Scale Forest Disturbances in a Tropical Montane Forest Using Landsat Time Series." *Remote Sensing of Environment*, **161**(0), 107 – 121. doi:10.1016/j.rse.2015.02.012.

Dutrieux L, DeVries B (2014). "**bfastSpatial**: Set of Utilities and Wrappers to Perform Change Detection on Satellite Image Time-Series." doi:10.5281/zenodo.49693. URL https://github.com/dutri001/bfastSpatial.

Friedl MA, Sulla-Menashe D, Tan B, Schneider A, Ramankutty N, Sibley A, Huang X (2010). "MODIS Collection 5 Global Land Cover: Algorithm Refinements and Characterization of New Datasets." *Remote Sensing of Environment*, **114**(1), 168 – 182. ISSN 0034-4257. doi:10.1016/j.rse.2009.08.016.

Fritz S, See L, You L, Justice C, Becker-Reshef I, Bydekerke L, Cumani R, Defourny P, Erb K, Foley J, Gilliams S, Gong P, Hansen M, Hertel T, Herold M, Herrero M, Kayitakire F, Latham J, Leo O, McCallum I, Obersteiner M, Ramankutty N, Rocha J, Tang H, Thornton P, Vancutsem C, van der Velde M, Wood S, Woodcock C (2013). "The Need for Improved Maps of Global Cropland." *Eos, Transactions American Geophysical Union*, **94**(3), 31–32. ISSN 2324-9250. doi:10.1002/2013EO030006.

Galford GL, Mustard JF, Melillo J, Gendrin A, Cerri CC, Cerri CE (2008). "Wavelet Analysis of MODIS Time Series to Detect Expansion and Intensification of Row-Crop Agriculture in Brazil." *Remote Sensing of Environment*, **112**(2), 576–587.

Giorgino T (2009). "Computing and Visualizing Dynamic Time Warping Alignments in R: The **dtw** Package." *Journal of Statistical Software*, **31**(7), 1–24. doi:10.18637/jss.v031.i07.

Goslee S (2011). "Analyzing Remote Sensing Data in R: The landsat Package." *Journal of Statistical Software*, **43**(1), 1–25. ISSN 1548-7660. doi:10.18637/jss.v043.i04.

Griffiths P, van der Linden S, Kuemmerle T, Hostert P (2013). "A Pixel-Based Landsat Compositing Algorithm for Large Area Land Cover Mapping." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **6**(5), 2088–2101. ISSN 1939-1404. doi:10.1109/JSTARS.2012.2228167.

Hijmans RJ (2015). ***raster**: Geographic Data Analysis and Modeling*. R package version 2.5-2, URL http://CRAN.R-project.org/package=raster.

Jönsson P, Eklundh L (2002). "Seasonality Extraction by Function Fitting to Time-Series of Satellite Sensor Data." *IEEE Transactions on Geoscience and Remote Sensing*, **40**(8), 1824–1832. doi:10.1109/TGRS.2002.802519.

Jönsson P, Eklundh L (2004). "TIMESAT – A Program for Analyzing Time-Series of Satellite Sensor Data." *Computers & Geosciences*, **30**(8), 833 – 845. ISSN 0098-3004. doi:10.1016/j.cageo.2004.05.006.

Kennedy RE, Yang Z, Cohen WB (2010). "Detecting Trends in Forest Disturbance and Recovery Using Yearly Landsat Time Series: 1. LandTrendr – Temporal Segmentation Algorithms." *Remote Sensing of Environment*, **114**(12), 2897–2910. ISSN 0034-4257. doi:10.1016/j.rse.2010.07.008.

Keogh E, Ratanamahatana CA (2005). "Exact Indexing of Dynamic Time Warping." *Knowledge Information Systems*, **7**(3), 358–386.

Kuhn M, with contributions from Jed Wing, Weston S, Williams A, Keefer C, Engelhardt A, Cooper T, Mayer Z, Kenkel B, the R Core Team, Benesty M, Lescarbeau R, Ziem A, Scrucca L, Tang Y, Candan C (2016). ***caret**: Classification and Regression Training*. R package version 6.0-64, URL http://CRAN.R-project.org/package=caret.

Lambin E, Linderman M (2006). "Time Series of Remote Sensing Data for Land Change Science." *IEEE Transactions on Geoscience and Remote Sensing*, **44**(7), 1926–1928. ISSN 0196-2892. doi:10.1109/TGRS.2006.872932.

le Maire G, Dupuy S, Nouvellon Y, Loos RA, Hakamada R (2014). "Mapping Short-Rotation Plantations at Regional Scale Using MODIS Time Series: Case of Eucalypt Plantations in Brazil." *Remote Sensing of Environment*, **152**(0), 136 – 149. doi:10.1016/j.rse.2014.05.015.

Lunetta RS, Knight JF, Ediriwickrema J, Lyon JG, Worthy LD (2006). "Land-cover Change Detection Using Multi-Temporal MODIS NDVI Data." *Remote Sensing of Environment*, **105**(2), 142 – 154. doi:10.1016/j.rse.2006.06.018.

Maus V (2015). ***dtwSat**: Time-Weighted Dynamic Time Warping for Satellite Image Time Series Analysis*. R package version 0.1.0, URL http://CRAN.R-project.org/package=dtwSat.

Maus V, Camara G, Cartaxo R, Sanchez A, Ramos FM, de Queiroz GR (2016). "A Time-Weighted Dynamic Time Warping Method for Land-Use and Land-Cover Mapping." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **PP**(99), 1–11. ISSN 1939-1404. doi:10.1109/JSTARS.2016.2517118.

Meyer D, Buchta C (2015). ***proxy**: Distance and Similarity Measures*. R package version 0.4-15, URL http://CRAN.R-project.org/package=proxy.

Moulds S, Buytaert W, Mijic A (2015). "An Open and Extensible Framework for Spatially Explicit Land Use Change Modelling: The LULCC R Package." *Geoscientific Model Development*, **8**(10), 3215–3229. doi:10.5194/gmd-8-3215-2015.

Müller M (2007). *Information Retrieval for Music and Motion*. Springer-Verlag, London.

Pebesma E (2012). "**spacetime**: Spatio-Temporal Data in R." *Journal of Statistical Software*, **51**(1), 1–30. ISSN 1548-7660. doi:10.18637/jss.v051.i07.

Pebesma EJ, Bivand RS (2005). "Classes and methods for spatial data in **R**." *R News*, **5**(2), 9–13. URL http://CRAN.R-project.org/doc/Rnews/.

Petitjean F, Inglada J, Gancarski P (2012). "Satellite Image Time Series Analysis Under Time Warping." *IEEE Transactions on Geoscience and Remote Sensing*, **50**(8), 3081–3095. ISSN 0196-2892. doi:10.1109/TGRS.2011.2179050.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rabiner L, Juang BH (1993). *Fundamentals of speech recognition*. Prentice-Hall International, Inc., New Jersey.

Reed BC, Brown JF, VanderZee D, Loveland TR, Merchant JW, Ohlen DO (1994). "Measuring Phenological Variability from Satellite Imagery." *Journal of Vegetation Science*, **5**(5), 703–714. doi:10.2307/3235884.

Sakamoto T, Van PC, Kotera, Nguyen KD, Yokozawa M (2009). "Analysis of Rapid Expansion of Inland Aquaculture and Triple Rice-Cropping Areas in a Coastal Area of the Vietnamese Mekong Delta Using MODIS Time-Series Imagery." *Landscape and Urban Planning*, **92**(1), 34–46. doi:10.1016/j.landurbplan.2009.02.002.

Sakoe H, Chiba S (1971). "A Dynamic Programming Approach to Continuous Speech Recognition." In *Proceedings of the Seventh International Congress on Acoustics, Budapest*, volume 3, pp. 65–69. Akadémiai Kiadó, Budapest.

Sakoe H, Chiba S (1978). "Dynamic Programming Algorithm Optimization for Spoken Word Recognition." *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **26**(1), 43–49. doi:10.1109/TASSP.1978.1163055.

Stonebraker M, Brown P, Zhang D, Becla J (2013). "SciDB: A Database Management System for Applications with Complex Analytics." *Computing in Science & Engineering*, **15**(3), 54–62.

Tuck SL, Phillips HR, Hintzen RE, Scharlemann JP, Purvis A, Hudson LN (2014). "**MODISTools** – Downloading and Processing MODIS Remotely Sensed Data in R." *Ecology and Evolution*, **4**(24), 4658–4668. ISSN 2045-7758. doi:10.1002/ece3.1273.

Velichko V, Zagoruyko N (1970). "Automatic Recognition of 200 Words." *International Journal of Man-Machine Studies*, **2**(3), 223–234. ISSN 0020-7373. doi:10.1016/S0020-7373(70)80008-6.

Verbesselt J, Hyndman R, Newnham G, Culvenor D (2010a). "Detecting Trend and Seasonal Changes in Satellite Image Time Series." *Remote Sensing of Environment*, **114**(1), 106–115. ISSN 0034-4257. doi:10.1016/j.rse.2009.08.014.

Verbesselt J, Hyndman R, Zeileis A, Culvenor D (2010b). "Phenological Change Detection While Accounting for Abrupt and Gradual Trends in Satellite Image Time Series." *Remote Sensing of Environment*, **114**(12), 2970 – 2980. ISSN 0034-4257. doi:10.1016/j.rse.2010.08.003.

Verbesselt J, Zeileis A, Herold M (2011). "Near Real-Time Disturbance Detection in Terrestrial Ecosystems Using Satellite Image Time Series: Drought Detection in Somalia." *Working papers*, Faculty of Economics and Statistics, University of Innsbruck.

Verbesselt J, Zeileis A, Herold M (2012). "Near Real-Time Disturbance Detection Using Satellite Image Time Series." *Remote Sensing of Environment*, **123**(0), 98 – 108. doi:10.1016/j.rse.2012.02.022.

Wardlow BD, Egbert SL, Kastens JH (2007). "Analysis of Time-Series MODIS 250 m Vegetation Index Data for Crop Classification in the U.S. Central Great Plains." *Remote Sensing of Environment*, **108**(3), 290 – 310. doi:10.1016/j.rse.2006.11.021.

Wickham H (2009). ***ggplot2**: Elegant Graphics for Data Analysis*. Springer-Verlag, New York. ISBN 978-0-387-98140-6. URL http://had.co.nz/ggplot2/book.

Wood S (2006). *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC.

Wood SN (2000). "Modelling and Smoothing Parameter Estimation with Multiple Quadratic Penalties." *Journal of the Royal Statistical Society (B)*, **62**(2), 413–428.

Wood SN (2003). "Thin-Plate Regression Splines." *Journal of the Royal Statistical Society (B)*, **65**(1), 95–114.

Wood SN (2004). "Stable and Efficient Multiple Smoothing Parameter Estimation for Generalized Additive Models." *Journal of the American Statistical Association*, **99**(467), 673–686.

Wood SN (2011). "Fast Stable Restricted Maximum Likelihood and Marginal Likelihood Estimation of Semiparametric Generalized Linear Models." *Journal of the Royal Statistical Society (B)*, **73**(1), 3–36.

Xiao X, Boles S, Liu J, Zhuang D, Frolking S, Li C, Salas W, III BM (2005). "Mapping Paddy Rice Agriculture in Southern China Using Multi-Temporal MODIS Images." *Remote Sensing of Environment*, **95**(4), 480 – 492. doi:10.1016/j.rse.2004.12.009.

Zeileis A, Grothendieck G (2005). "**zoo**: S3 Infrastructure for Regular and Irregular Time Series." *Journal of Statistical Software*, **14**(6), 1–27. URL http://www.jstatsoft.org/v14/i06/.

Zhang X, Friedl MA, Schaaf CB, Strahler AH, Hodges JC, Gao F, Reed BC, Huete A (2003). "Monitoring Vegetation Phenology Using MODIS." *Remote Sensing of Environment*, **84**(3), 471 – 475. doi:10.1016/S0034-4257(02)00135-9.

Zhu Z, Woodcock CE, Olofsson P (2012). "Continuous Monitoring of Forest Disturbance Using all Available Landsat Imagery." *Remote Sensing of Environment*, **122**(0), 75–91. ISSN 0034-4257. doi:10.1016/j.rse.2011.10.030. Landsat Legacy Special Issue.

Affiliation:

Victor Maus
INPE
Image Processing Division
National Institute for Space Research
Av. dos Astronautas, 1758
12227-010 Sao Jose dos Campos, Brazil
E-mail: vwmaus1@gmail.com
URL: www.dpi.inpe.br/maus