Chloris Geospatial:

An interactive application for land surface reflectance

PREPARED BY:

BOSTON UNIVERSITY MSSP PROGRAM

Nuo Chen, Cathy Feng, Bill Gao, Brenda Han, Ivan Jin, Ziyang Lin, Christine Liu, Christian Luo, Haochen Pan, Qihan Su, Tong Sun, Priam Vyas, Lucy Wang, Shengbo Wang, Yiluo Wang, Kai Wu, Kelly Xu, Danya Zhang, Lanlin Zhao

PREPARED FOR:

CHLORIS GEOSPATIAL

Alessandro Baccini, CO-FOUNDER & CSO

DECEMBER 20, 2022

Table of Contents

Introduction

With the goal of reducing greenhouse gas emissions, as stated by the Paris Agreement, we provide a simple intuitive R Shiny application to help Chloris Geospatial visualize the changes in forests around the world.

We will discuss the background and the mechanics behind the end product. First, we begin by using data from the U.S. Geological Survey (USGS), a science bureau that provides open source data from satellite imagery. The specific satellite that we will focus on is the Landsat 8.

Data

The Landsat 8 contains two sensors, OLI (operational land imager) and TIRS (thermal infrared sensor). We have made a temporary cloud server to help us store the data for our application. An important note, in order to use our application, it will be necessary to clone or create your own database with the necessary GeoTIFF files.

	path	row	date	band	filename	count
1	227	62	2013-03-27	2	LC08_L1TP_227062_20130327_20170505_01_T2_B2...	841
2	227	62	2013-03-27	3	LC08_L1TP_227062_20130327_20170505_01_T2_B3...	841
3	227	62	2013-03-27	4	LC08_L1TP_227062_20130327_20170505_01_T2_B4...	841
4	227	62	2013-03-27	5	LC08_L1TP_227062_20130327_20170505_01_T2_B5...	841
5	227	62	2013-03-27	QA	LC08_L1TP_227062_20130327_20170505_01_T2_BQ...	841
6	227	62	2013-04-18	2	LC08_L1GT_227062_20130418_20170505_01_T2_B2...	870
7	227	62	2013-04-18	3	LC08_L1GT_227062_20130418_20170505_01_T2_B3...	870
8	227	62	2013-04-18	4	LC08_L1GT_227062_20130418_20170505_01_T2_B4...	870
9	227	62	2013-04-18	5	LC08_L1GT_227062_20130418_20170505_01_T2_B5...	870
10	227	62	2013-04-18	QA	LC08_L1GT_227062_20130418_20170505_01_T2_B...	870

As you can see from the example query output above, there are variables that you may extract information from, including path, row, date, band, and filename. Our provided R code for the function simply produces a graph, so you will not be able to view the data directly using our application. This is simply some background information on the data.

Functions & Queries

The first step in our development process was to create a series of functions with suitable queries to implement in our Shiny App. The general purpose of our functions is to classify each pixel in an image based on integer values, which represent the natural conditions.

Given below are our original functions implemented in our application:

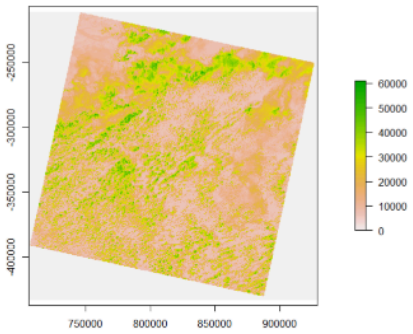
1. **number2binary()**: the purpose of this function is to change the data number to a binary format. In a binary format, pixels that are affected by certain natural conditions will be distinguished by either a '0' or a '1'. This is then converted into a paired value through binary calculation. The table below is an example of a binary array.

	Cloud	Cloud shadow	Snow/Ice	Cirrus
User select	1	0	1	0
Value in QA bits	1	1	0	0
Paired Value	1	0	0	0

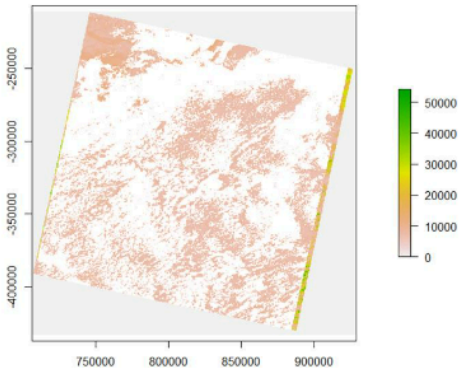
2. **getrcl()**: the purpose is to get the reclassify values of a Raster object. In short, this code is to get a numeric matrix by selected conditions to reclassify the QA band, then used the reclassified QA band to mask value from bands, otherwise the scatter plot will not have color distinctions. The specific content of the code is as follows:

$$rcl = getrcl(condition\_array)$$

3. **getMask()**: the purpose of this function is to get an image mask. In our part, the four conditions we choose will be "masked". After entering *maskValue = c(,)* value in the console, we are able to get a masked graph. The graph below shows an example of the result. We can see that if we choose band 4 with no clouds, but with cloud shadows, snow, and cirrus points ('4', 'c(1,0,0,0)'), the "mask" will help to see which areas of the map fit within the selected choice.



Raw graph

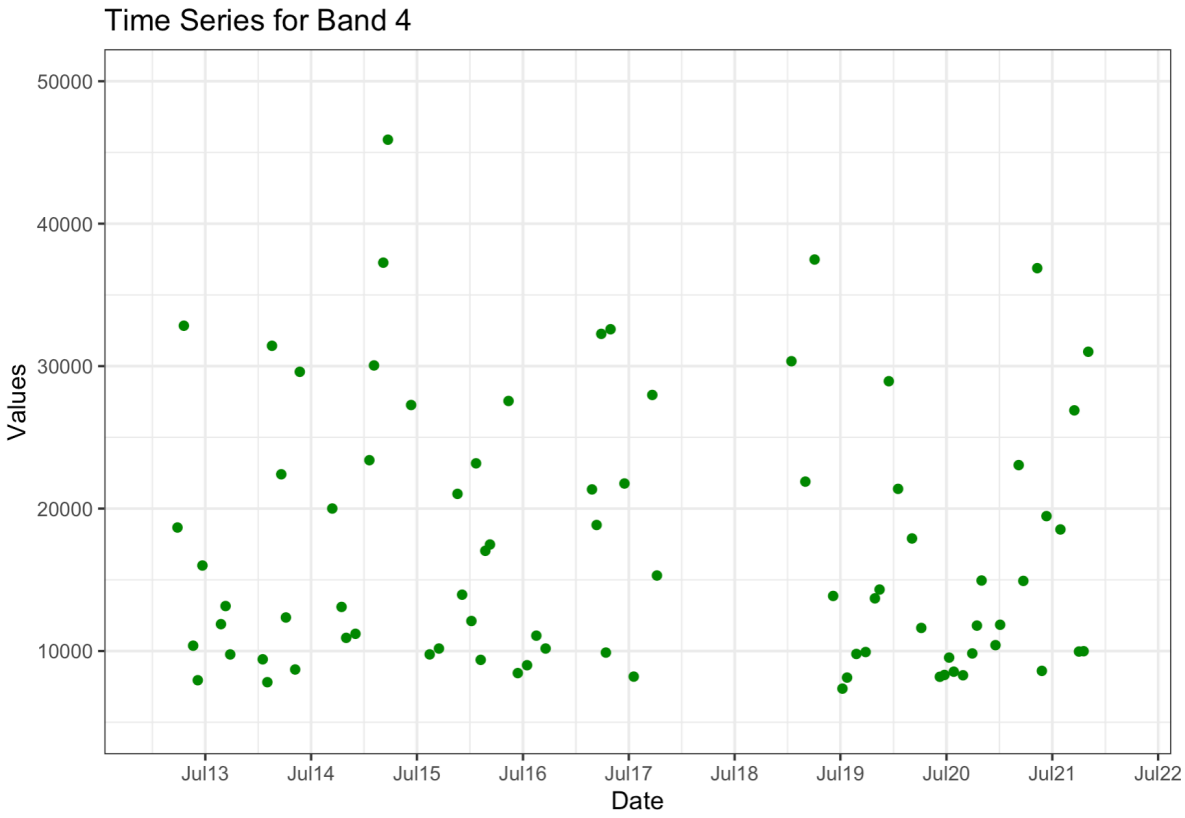


getMask('2013-03-27', '4', c(1,0,0,0))

4. **LS_band_graph**: the final function that we created incorporates the functions described above. It summarizes and combines the above functions and provides a graph for time series which is visible on the website. The function requires the following arguments:

x	easting of selected pixel (UTM format, 6 digits) UTM Stands for Universal Transverse Mercator
y	northing of selected pixel (UTM format, 6 digits)
band	landsat band of interest
condition_array	Binary array of 1's and 0's with conditions to be plotted. c(cloud_value, cloudshadow_value, snowice_value, cirrus_value) -> c(1,1,1,1). 1 means values will **not** be included.

For example, $LS\_band\_graph(760000, \ -270000, 4, \ c(1, 0, 0, 0))$ provides the following scatter plot:

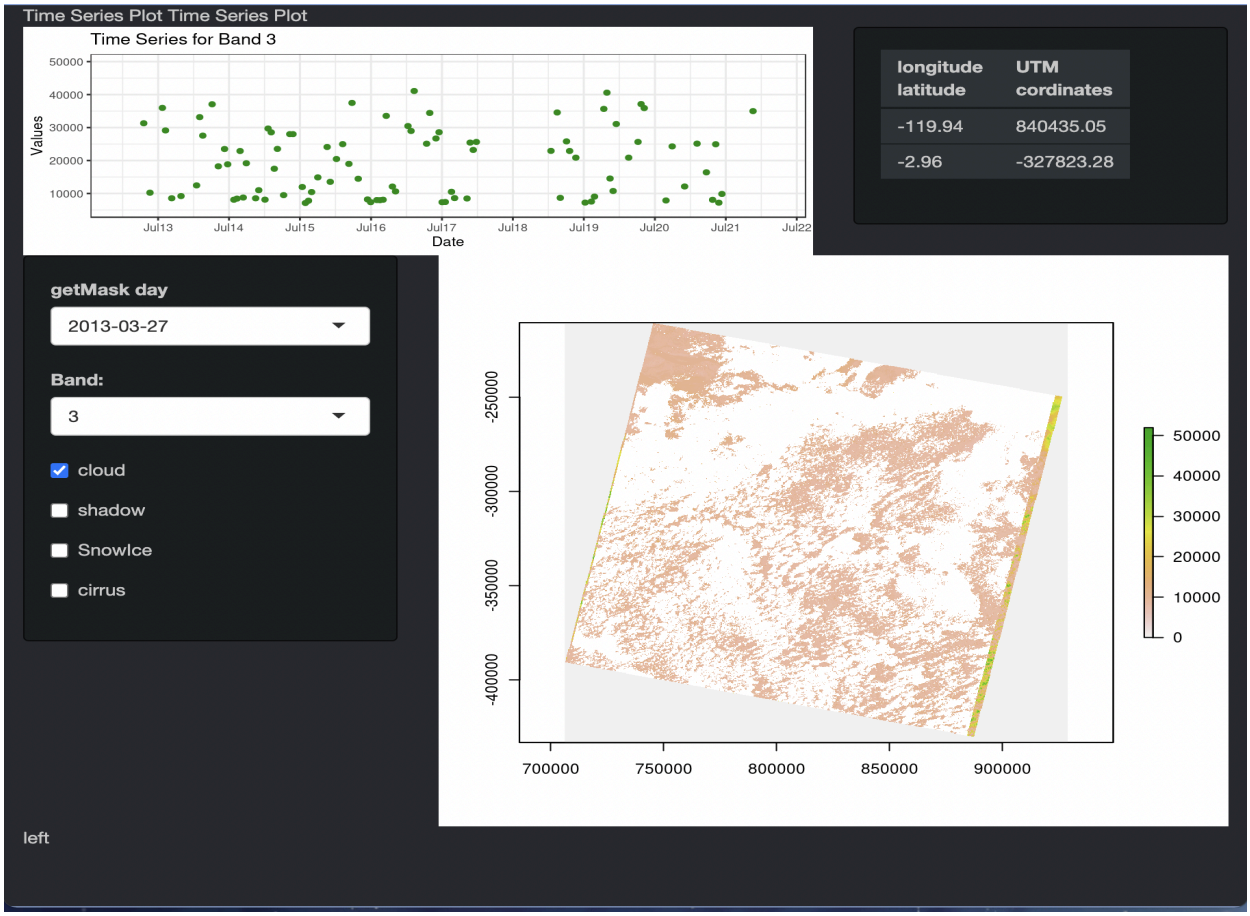


This plot represents the intensity value (surface light reflectance) for the location indicated by 760000, -270000, band 4, not including clouds, but including cloud shadows, snow, and cirrus points.

Shiny Application

Using the functions above, we have developed a R Shiny application. The interface is as follows:

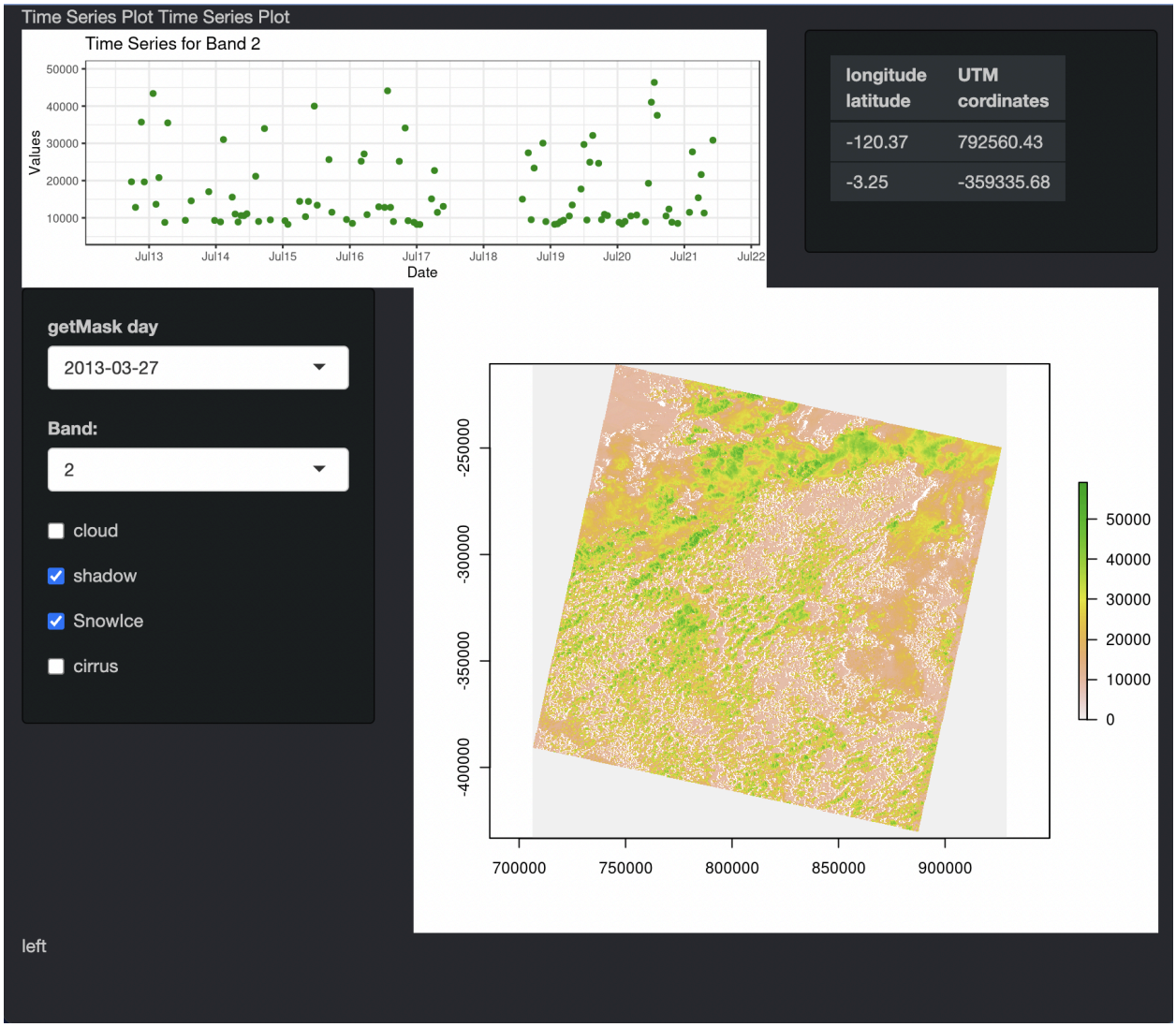
Figure 1



The image in the bottom right displays an area of land of the user's choosing. Upon clicking on a specific pixel on the image, the respective time series band plot will display in the upper left; the graph will change with respect to which pixel the user selects. As you can see, the plot above shows band three with cloud part masked condition. The user is free to select any combination of start date, band number, and mask conditions.

For convenience, we have also included the UTM coordinates and their respective longitude and latitude translations.

Figure 2



Looking at a different combination of selections, we can see the plot for band 2 from the date 2013–3-27, with shadow and SnowIce chosen masked are shown above.

Although our current shiny does not cover a large part of the area, our method is applicable to different locations on the planet.

This concludes our report on our newly developed interaction R Shiny application for visualizing band values worldwide. For more information on Landsat and the data, please visit https://www.usgs.gov/landsat-missions/landsat-8.