Processing MODIS data

Benson Kenduiywo, Aniruddha Ghosh and Robert J. Hijmans

Nov 30, 2023

CONTENTS

Benson Kenduiywo, Aniruddha Ghosh and Robert Hijmans

CHAPTER

ONE

THE MODERATE RESOLUTION IMAGING SPECTRORADIOMETER (MODIS)

This document provides a basic overview of MODIS satellite data. Additional resources on terminology and fundamentals of remote sensing can be found here and here. and this "review of Selected MODIS Algorithms, Data Products, and Applications".

1.1 Introduction

The Moderate Resolution Imaging Spectroradiometer (MODIS) is an optical sensor onboard two satellites named Terra (originally known as EOS AM-1) and Aqua (originally known as EOS PM-1) operated by NASA. Terra was launched on 18 December 1999 and Aqua was launched on 24 May 2002. Terra's orbit around the Earth is such that it passes from North to South across the equator in the morning, while Aqua passes South to North over the equator in the afternoon. Terra MODIS and Aqua MODIS take images for the entire Earth's surface every 1 to 2 days. The data collected is the intensity of the light reflected by earth in 36 different wavelengths ("colors"). Both Terra and Aqua also have other sensors besides MODIS. These two sensors have publicly available daily archive of conditions on the earth surface for the past 20 years!

1.2 Spatial and spectral properties

The table below summarized the intended use for cation and corresponding spatial resolution in which they are acquired.

Table 1: MODIS spectral bands and their spatial resolution

Spectral band number	Intended use	Spatial resolution
1-2	Land/Cloud boundaries	250
3-7	Land/Cloud/Aerosol properties	500
8-16	Ocean Color/Phytoplankton/B iogeochemistry	1000
17-19	Atmospheric water vapor	1000
20-23	Surface/Cloud temperature	1000
24-25	Atmospheric temperature	1000
26	Cirrus clouds	1000
27-29	Water Vapor	1000
30	Ozone	1000
31-36	Surface/Cloud temperature cloud top altitude	1000

1.3 Processing levels

MODIS data is available at different levels of processing.

- Level 0: Unprocessed raw satellite feeds ("ungridded orbital swath data").
- Level 1A: Raw un-calibrated swath data with georeference information.
- Level 1B: radiometrically calibrated with georeference information.
- Level 2: level 1 data atmospherically corrected to yield surface reflectance.
- Level 2G: Levcel 2 data calibrated and stored as integers on the sinusoidal tiled gridded system.
- Level 3: level 2 data, properly georeferenced, and often temporally composited or averaged. "Best" pixel selection (e.g. one value per pixel over a period of 7 days)
- Level 4: data products that have been put through additional processing. Higher processing level than level 3.

More information of these products can be obtain from MODIS Nomenclature.

1.4 Products

The higher level data products can be grouped based on their application areas (see: https://modis.gsfc.nasa.gov/data/dataprod/). These products are usually distributed in different spatial resolution (250/500/1000 m) and temporal composite (daily/8-day/16-day/monthly/annual) combinations for both sensors.

Each of those products has a prefix for instance, "MOD" are products from the Terra satellite while "MYD" are products from the Aqua satellite and prefix "MCD" are products generated from a combination of the two satellites.

MODIS Land Products
- MODIS Surface Reflectance
- MODIS Land Surface Temperature and Emissivity (MOD11)
- MODIS Land Surface Temperature and Emissivity (MOD21)
- MODIS Land Cover Products
- MODIS Vegetation Index Products (NDVI and EVI)
- MODIS Thermal Anomalies - Active Fires
- MODIS Fraction of Photosynthetically Active Radiation (FPAR) / Leaf Area Index (LAI)
- MODIS Evapotranspiration
- MODIS Gross Primary Productivity (GPP) / Net Primary Productivity (NPP)
- MODIS Bidirectional Reflectance Distribution Function (BRDF) / Albedo Parameter
- MODIS Vegetation Continuous Fields
- MODIS Water Mask
- MODIS Burned Area Product

MODIS Atmosphere Products
- MODIS Aerosol Product
- MODIS Total Precipitable Water
- MODIS Cloud Product
- MODIS Atmospheric Profiles
- MODIS Atmosphere Joint Product

- MODIS Atmosphere Gridded Product
- MODIS Cloud Mask

MODIS Cryosphere Products
- MODIS Snow Cover
- MODIS Sea Ice and Ice Surface Temperature

MODIS Ocean Products
- MODIS Sea Surface Temperature
- MODIS Remote Sensing Reflectance
- MODIS Chlorophyll-a Concentration
- MODIS Diffuse Attenuation at 490 nm
- MODIS Particulate Organic Carbon
- MODIS Particulate Inorganic Carbon
- MODIS Normalized Fluorescence Line Height (FLH)
- MODIS Instantaneous Photosynthetically Available Radiation
- MODIS Daily Mean Photosynthetically Available Radiation

These products are distributed by different Distributed Active Archive Center(DAAC). For example Land Processes Distributed Active Archive Center, LP DAAC distributes the following products.

MODIS products; # is the product number variant e.g., 'MOD', 'MYD' or 'MCD'.

ID	Name	ID	Name
1	Level-1A Radiance Counts	20	Chlorophyll II Fluorescence
2	Level-1B Calibrated, Geolocated Radiances	21	Chlorophyll a Pigment Concentration
3	Geolocation Data Set	22	Photosynthetically Active Radiation (PAR)
4	Aerosol Product	23	Suspended Solids Concentration in Ocean Water
5	Total Precipitable Water	24	Organic Matter Concentration
6	Cloud Product	25	Coccolith Concentration
7	Atmospheric Profiles	26	Ocean Water Attenuation Coefficient
8	Gridded Atmosphere Products (Level-3)	27	Ocean Primary Productivity
9	Atmospherically Corrected Surface Reflectance	28	Sea Surface Temperature
10	Snow Cover	29	Sea Ice Cover
11	Land Surface Temperature & Emissivity	31	Phycoerythrin Concentration
12	Land Cover/Land Cover Change	35	Cloud Mask
13	Vegetation Indices	36	Total Absorption Coefficient
14	Thermal Anomalies, Fires and Biomass Burning	37	Ocean Aerosol Properties
15	Leaf Area Index and FPAR	39	Clear Water Epsilon
16	Surface Resistance & Evapotranspiration	43	Albedo 16-Day Level-3
17	Vegetation Production, Net Primary Productivity	44	Vegetation Cover Conversion & Continuous Fields
18	Normalized Water Leaving Radiance	ALB	Snow and Sea Ice Albedo
19	Pigment Concentration		

1.5 File format

MODIS data is stored in "Hierarchical Data Format - Earth Observation System" (HDF-EOS) files. HDF format was developed to support storing multiobject data sets (subdatasets) within one file (container). These subdatasets can include raster images, tables, scientific data sets (multidimensional arrays), annotations, vector files, palette information and many more formats.

Read more about use of HDF in NASA Earth Observation System data in HDF-EOS Information Center.

Converting HDF files to a common format (e.g. GeoTiff) that can be read across different software libraries has been one of the major challenges with MODIS data processing. Until recently, commercial programs like ArcGIS, ENVI were not able to read and explore HDF files. In the next section we will show how you can use R to search, download and process MODIS data..

1.6 Geographic information

Each file stores data for an area of 10 by 10 degrees, using the sinusoidal coordinate reference system.

1.7 Tiling Scheme

The tiles are identified using rows "h" (for horizontal) and columns "v" (for vertical). MODIS uses a vertical horizontal tiling systems (e.g.tile h09v04 as shown in the figure below. A kmz file with MODIS Tile boundaries can be found here

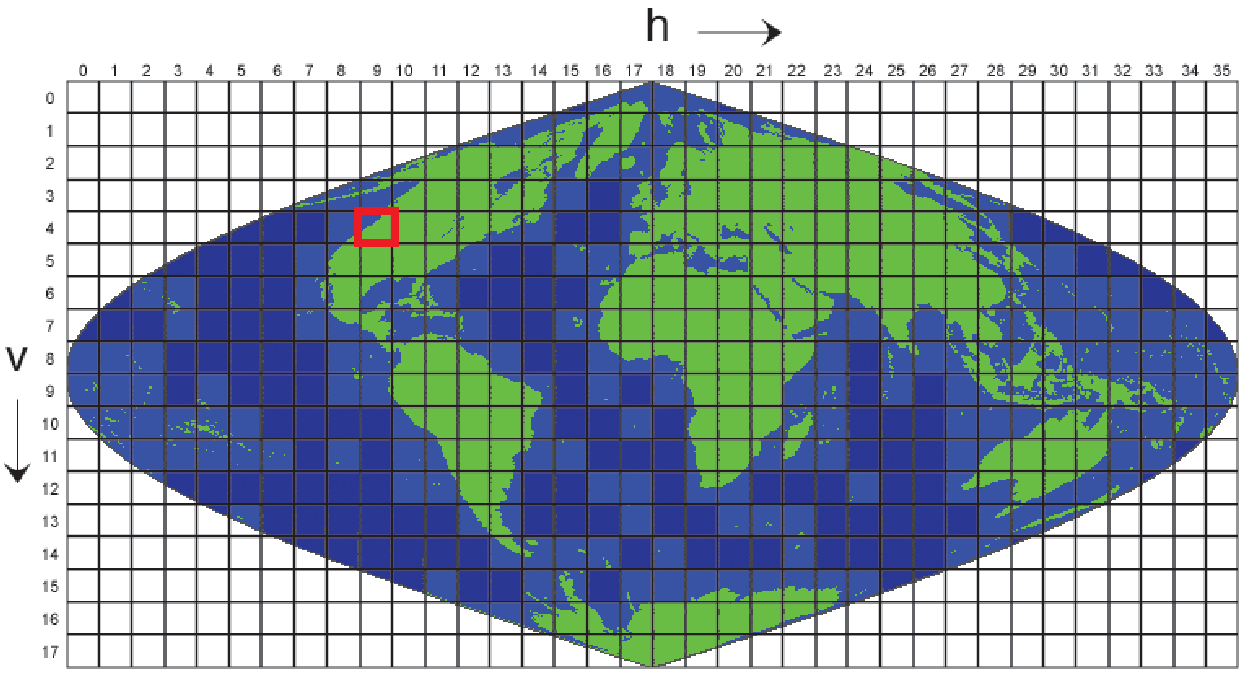


Fig. 1: MODIS tiles indexed horizontally and vertically

1.8 MOD09A1 product

Now let’s look at an example MODIS surface reflectance product. “MOD09A1” is a surface reflectance product from data acquired by the MODIS sensor onboard the Terra satellite — hence the prefix “MOD”. It provides estimated “target at surface” spectral reflectance values that have been calibrated for atmospheric conditions such as gasses, aerosols, and Rayleigh scattering. Global surface reflectance products can be obtained at either 250 m or 500 m spatial resolution, and as as daily or 8-day composite data. Below we show a table of the characteristics of the 13 layers that come with “MOD09A1”. A more in-depth description is available here.

MODIS products

Band	Name	Type/ units	Data Type	Fill value	Valid Range	Scale fact
1	Red (620-670 nm)	Reflectance	16-bit signed integer	28672	-100 - 16000	0.0001
2	NIR (841-876 nm)	Reflectance	16-bit signed integer	28672	-100 - 16000	0.0001
3	Blue (459-479 nm)	Reflectance	16-bit signed integer	28672	-100 - 16000	0.0001
4	Green (545-565 nm)	Reflectance	16-bit signed integer	28672	-100 - 16000	0.0001
5	NIR (1230-1250 nm)	Reflectance	16-bit signed integer	28672	-100 - 16000	0.0001
6	SWIR (1628-1652)	Reflectance	16-bit signed integer	28672	-100 - 16000	0.0001
7	SWIR (2105-2155 nm)	Reflectance	16-bit signed integer	28672	-100 - 16000	0.0001
8	Reflectance Band Quality	Bit Field	32-bit unsigned integer	4294967295	NA	NA
9	Solar Zenith Angle	Degree	16-bit signed integer	0	0 - 18000	0.01
10	View Zenith Angle	Degree	16-bit signed integer	0	0 - 18000	0.01
11	Relative Azimuth Angle	Degree	16-bit signed integer	0	-18000 - 18000	0.01
12	500 m State Flags	Bit field	16-bit unsigned integer	65535	NA	NA
13	Day of Year	Julian day	16-bit unsigned integer	65535	1 - 366	NA

CHAPTER

TWO

DOWNLOADING MODIS DATA

MODIS data can be acquired through multiple services, and it can be confusing to find out what to get where. This website has a lot of information on MODIS resources.

Websites are great for preliminary data exploration, they are not ideal for accessing large number of files or when automatic updates are required. Manually downloading also makes workflows less reproducible. So once you know what data you need, it is better to use automatic downloading methods. Here we show how to use *R* package `luna` to download MODIS data for a geographic area and time period.

A large number of MODIS products is available. We first need to find data for the product that we deem best suited for our study.

```
library(terra)
library(luna)

# lists all products that are currently searchable
prod <- getProducts()
head(prod)
##      provider             concept_id                          short_name
## 1        GHRC             C1000-GHRC                            dc8capac
## 5       CDDIS      C1000000000-CDDIS              CDDIS_DORIS_data_cycle
## 18 LANCEAMSR2 C1000000000-LANCEAMSR2                      A2_RainOcn_NRT
## 19  NSIDC_ECS  C1000000000-NSIDC_ECS                            NmHRIR3H
## 23  ORNL_DAAC  C1000000000-ORNL_DAAC GLOBAL_MICROBIAL_BIOMASS_C_N_P_1264
## 25      SEDAC      C1000000000-SEDAC               CIESIN_SEDAC_EPI_2012
##    version
## 1        1
## 5        1
## 18       0
## 19       1
## 23       1
## 25 2012.00

# to find the MODIS products
modis <- getProducts("^MOD|^MYD|^MCD")
head(modis)
##        provider             concept_id short_name version
## 806  LPDAAC_ECS C1000000120-LPDAAC_ECS     MOD44B     051
## 1433 LPDAAC_ECS C1000000400-LPDAAC_ECS   MCD43D10     006
## 1447 LPDAAC_ECS C1000000401-LPDAAC_ECS   MCD43D33     006
## 1452 LPDAAC_ECS C1000000402-LPDAAC_ECS   MCD43D45     006
## 1454 LPDAAC_ECS C1000000403-LPDAAC_ECS   MCD43D26     006
## 1456 LPDAAC_ECS C1000000404-LPDAAC_ECS   MCD43D49     006
```

We will use “MOD09A1” for this turorial.

```
product <- "MOD09A1"
```

To learn more about a specific product you can launch a webpage

```
productInfo(product)
```

Note that the entire MODIS archive is regularly re-processed for overall imporvement and revisions. We use version 6 or later for our analysis.

Once we finalize the product we want to use, we define some parameters for the data we want: product name, start and end date, and area of interest.

```
start <- "2010-01-01"
end <- "2010-01-07"
```

We will download an example MODIS 8-day composite tile. Our area of interest is Marsabit county, Kenya. To define the area of interest, we can define a spatial extent, or use an object that has an extent. Here we use a polygon for Marsabit.

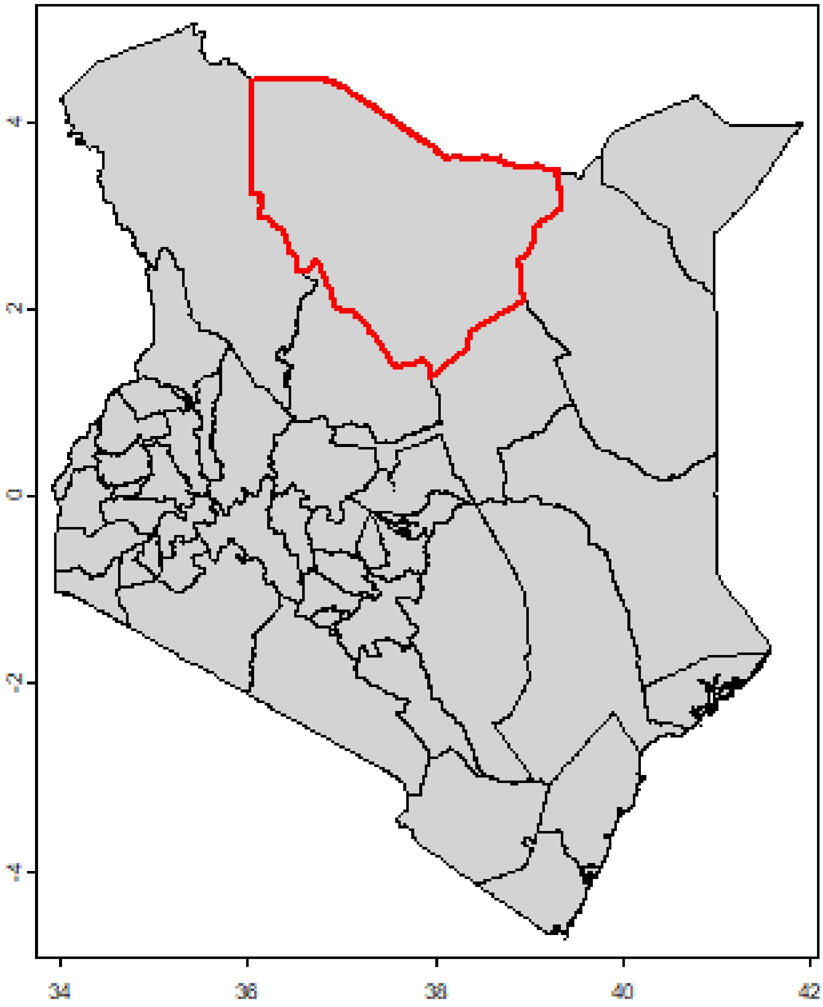
```
ken <- geodata::gadm("Kenya", level=1, path=".")
ken
##  class       : SpatVector
##  geometry    : polygons
##  dimensions  : 47, 11  (geometries, attributes)
##  extent      : 33.90959, 41.92622, -4.720417, 5.061166  (xmin, xmax, ymin, ymax)
##  coord. ref. : lon/lat WGS 84 (EPSG:4326)
##  names       :   GID_1 GID_0 COUNTRY  NAME_1 VARNAME_1 NL_NAME_1 TYPE_1
##  type        :   <chr> <chr>   <chr>   <chr>     <chr>     <chr>  <chr>
##  values      : KEN.1_1   KEN   Kenya Baringo        NA        NA County
##                KEN.2_1   KEN   Kenya   Bomet        NA        NA County
##                KEN.3_1   KEN   Kenya Bungoma        NA        NA County
##  ENGTYPE_1  CC_1 HASC_1 ISO_1
##      <chr> <chr>  <chr> <chr>
##     County    30  KE.BA KE-01
##     County    36  KE.BO KE-02
##     County    39  KE.BN KE-03
```

ken is a SpatVector of polygons. We can subset it get the polygon for Marsabit:

```
i <- ken$NAME_1 == "Marsabit"
aoi <- ken[i,]
```

And the check our results we make a map

```
plot(ken, col="light gray")
lines(aoi, col="red", lwd=2)
```



Let’s now find out what MODIS data is available for this area. We can search the data available from a NASA server

```
mf <- luna::getModis(product, start, end, aoi=aoi, download = FALSE)
## Warning in luna::getModis(product, start, end, aoi = aoi, download = FALSE):
## this method has been replaced by getNASA. It will be removed in future versions
mf
## [1] "MOD09A1.A2009361.h21v08.061.2021149144347.hdf"
## [2] "MOD09A1.A2010001.h21v08.061.2021150093948.hdf"
```

To download the tiles, usually you would download them to a folder where you save the data for your project. Here we use the temporary directory. You should use a specific direcory of your choice instead.

```
datadir <- file.path(dirname(tempdir()), "_modis")
dir.create(datadir, showWarnings=FALSE)
```

You also need to provide the username and password for your (free) EOSDIS account. If you do not have an account, you can sign up here. My passwords are stored in a file that I read below (sorry, I cannot show you the values).

```
up <- readRDS("../../../../pwds.rds")
up <- up[up$service == "EOSDIS", ]
```

Now we are ready to download the data

```
mf <- luna::getNASA(product, start, end, aoi=aoi, download=TRUE,
                     path=datadir, username=up$user, password=up$pwd)
mf
## [1] "c:/temp/_modis/MOD09A1.A2009361.h21v08.061.2021149144347.hdf"
## [2] "c:/temp/_modis/MOD09A1.A2010001.h21v08.061.2021150093948.hdf"
```

In the next chapters we will use the downloaded files.

CHAPTER

THREE

IMAGE EXPLORATION

3.1 Introduction

Now that we have successfully downloaded one MODIS tile, we can use `terra` package to explore and visualize it. Please note that MODIS tiles are distributed in HDF format that may include sub-datasets. The sub-dataset and processing steps might be different for various MODIS collections (e.g. daily scenes, vegetation indices).

Now that we have download some MODIS data, we can explore and visualize it.

First create a SpatRaster object from the file created on the previous page.

```
datadir <- file.path(dirname(tempdir()), "_modis")
mf <- file.path(datadir, "MOD09A1.A2009361.h21v08.061.2021149144347.hdf")

library(terra)
r <- rast(mf[1])
r
## class       : SpatRaster
## dimensions  : 2400, 2400, 13  (nrow, ncol, nlyr)
## resolution  : 463.3127, 463.3127  (x, y)
## extent      : 3335852, 4447802, 0, 1111951  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=sinu +lon_0=0 +x_0=0 +y_0=0 +R=6371007.181 +units=m +no_defs
## sources     : MOD09A1.A2009361.h21v08.061.2021149144347.hdf:MOD_Grid_500m_Surface_
↪Reflectance:sur_refl_b01
##               MOD09A1.A2009361.h21v08.061.2021149144347.hdf:MOD_Grid_500m_Surface_
↪Reflectance:sur_refl_b02
##               MOD09A1.A2009361.h21v08.061.2021149144347.hdf:MOD_Grid_500m_Surface_
↪Reflectance:sur_refl_b03
##               ... and 10 more source(s)
## varnames    : MOD09A1.A2009361.h21v08.061.2021149144347
##               MOD09A1.A2009361.h21v08.061.2021149144347
##               MOD09A1.A2009361.h21v08.061.2021149144347
##               ...
## names       : sur_refl_b01, sur_refl_b02, sur_refl_b03, sur_refl_b04, sur_refl_b05,␣
↪sur_refl_b06, ...
```

Exercise: Find out at least 5 properties (path, row, date of collection etc) of the MODIS data from the information embedded in the filename.

3.2 Image properties

The code below illustrates how you can load HDF files and access image properties of a SpatRaster object.

The coordinate reference system (CRS)

```
crs(r)
## [1] "PROJCRS[\"unnamed\",\n    BASEGEOGCRS[\"Unknown datum based upon the custom␣
↪spheroid\",\n        DATUM[\"Not specified (based on custom spheroid)\",\n          ␣
↪ELLIPSOID[\"Custom spheroid\",6371007.181,0,\n                LENGTHUNIT[\"metre\",1,\
↪n                    ID[\"EPSG\",9001]]]],\n        PRIMEM[\"Greenwich\",0,\n          ␣
↪  ANGLEUNIT[\"degree\",0.0174532925199433,\n                ID[\"EPSG\",9122]]]],\n    ␣
↪CONVERSION[\"Sinusoidal\",\n        METHOD[\"Sinusoidal\"],\n        PARAMETER[\
↪"Longitude of natural origin\",0,\n            ANGLEUNIT[\"degree\",0.
↪0174532925199433],\n            ID[\"EPSG\",8802]],\n        PARAMETER[\"False easting\
↪",0,\n            LENGTHUNIT[\"metre\",1],\n            ID[\"EPSG\",8806]],\n       ␣
↪PARAMETER[\"False northing\",0,\n            LENGTHUNIT[\"metre\",1],\n            ID[\
↪"EPSG\",8807]]],\n    CS[Cartesian,2],\n        AXIS[\"(E)\",east,\n            ␣
↪ORDER[1],\n            LENGTHUNIT[\"Meter\",1]],\n        AXIS[\"(N)\",north,\n        ␣
↪    ORDER[2],\n            LENGTHUNIT[\"Meter\",1]]]"
```

And the number of cells, rows, columns

```
dim(r)
## [1] 2400 2400   13
nrow(r)
## [1] 2400
ncol(r)
## [1] 2400
# Number of layers (bands)
nlyr(r)
## [1] 13
ncell(r)
## [1] 5760000
```

The spatial resolution is about 500 m

```
res(r)
## [1] 463.3127 463.3127
```

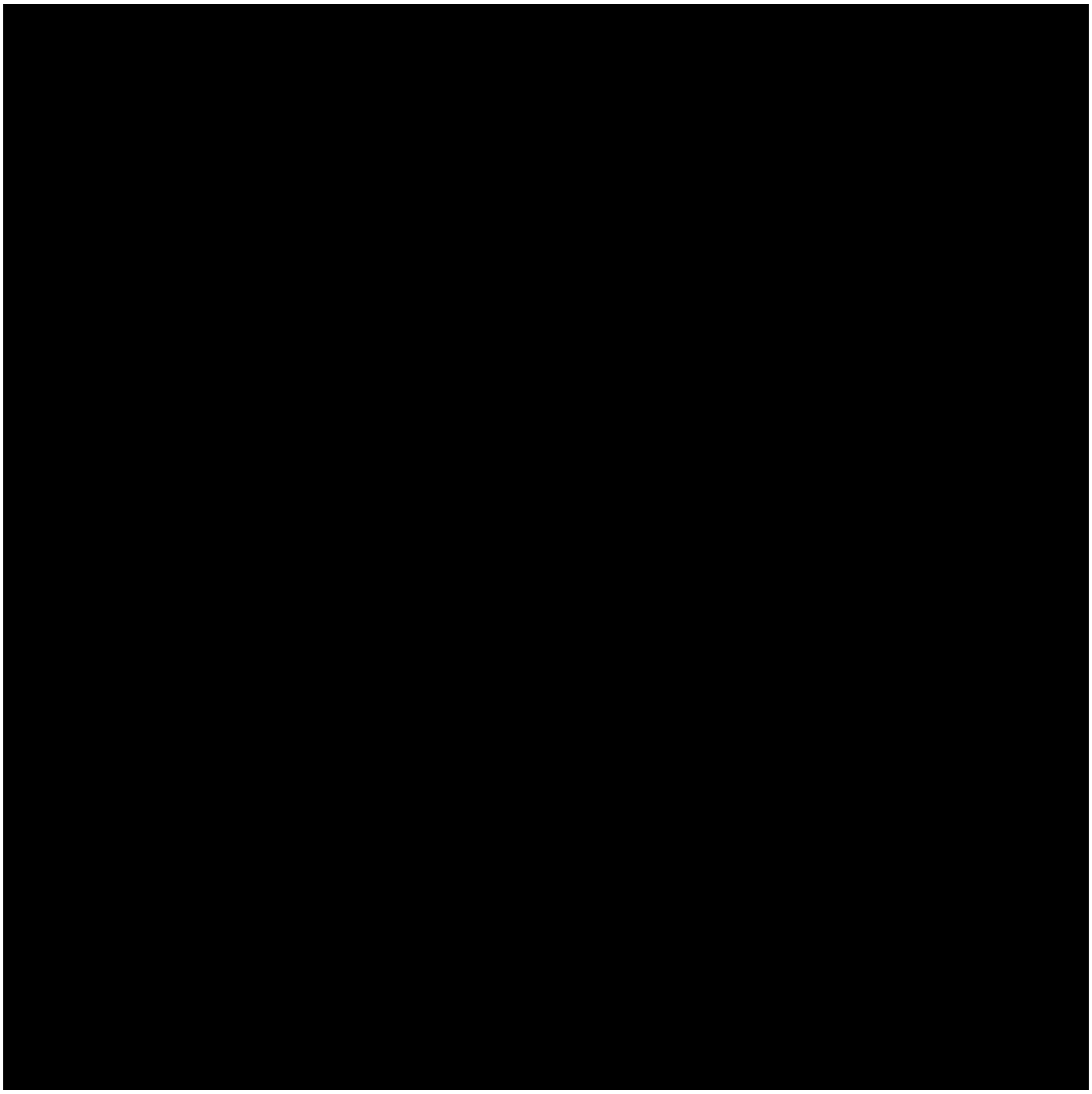
The layernames tell us what “bands” we have

```
names(r)
##  [1] "sur_refl_b01"         "sur_refl_b02"         "sur_refl_b03"
##  [4] "sur_refl_b04"         "sur_refl_b05"         "sur_refl_b06"
##  [7] "sur_refl_b07"         "sur_refl_qc_500m"     "sur_refl_szen"
## [10] "sur_refl_vzen"        "sur_refl_raz"         "sur_refl_state_500m"
## [13] "sur_refl_day_of_year"
```

3.3 Plot

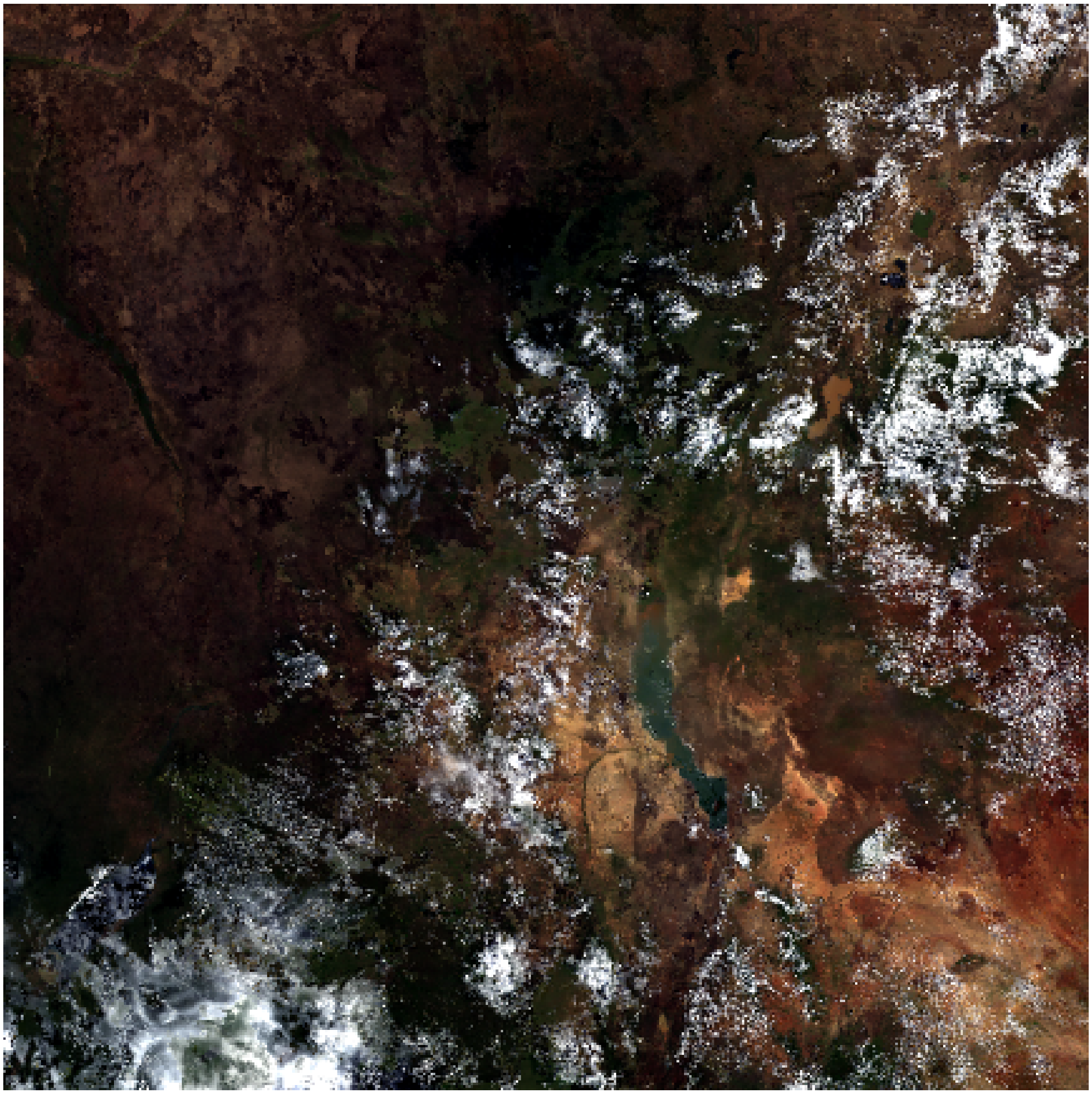
Now let's make some simple plots to see if things look reasonable.

```
# Create an image RGB composite plot
plotRGB(r, r = 1, g = 4, b = 3)
```



```
# Disappointing? apply some stretching; see `?plotRGB` for more options

plotRGB(r, r = 1, g = 4, b = 3, stretch="lin")
```



Exercise: Create False Color Composite plot using the same data. Hint: try `plotRGB` and specify the bands you need.

Exercise: Save the plots to graphics files. Hint: have a look at `?png`.

CHAPTER

FOUR

QUALITY CONTROL

Not all pixels in a MODIS image are not suitable for analysis of land areas. For example, pixel quality can be negatively affected by clouds or other atmospheric conditions. For that reason, each image needs to be pre-processed to remove the values that cannot be used.

Each MODIS file contains quality assurance (QA) information that can be used to identify the pixels to remove. The QA information is stored in a somewhat complicated bit-encoding format. This allows for very efficient storage, but it makes it much harder to use.

A single bit can represent two values — 0 (no) or 1 (yes); a combination of two bits can represent 2^2 = 4 values (00, 01, 10, and 11), and with three bits you can store 2^3 = 8 values. This video by USGS explains the QA encoding.

The table below shows State QA description (16-bit) for 500 m, 1 km and lower resolution MODIS surface reflectance products. For product specific QA attribute, follow the MODIS Surface Reflectance User's Guide.

bit	variable	value	description
1-2	cloud state	00	clear
		01	cloudy
		10	mixed
		11	not set, assumed clear
3	cloud shadow	1	yes
		0	no
4-6	land/water flag	000	shallow ocean
		001	land
		010	ocean coastlines and lake shorelines
		011	shallow inland water
		100	ephemeral water
		101	deep inland water
		110	continental/moderate ocean
		111	deep ocean
7-8	aerosol quantity	00	climatology
		01	low
		10	average
		11	high
9-10	cirrus detected	00	none
		01	small
		10	average
		11	high
11	cloud flag	1	cloud
		0	no cloud
12	fire flag	1	fire

continues on next page

Table 1 – continued from previous page

bit	variable	value	description
		0	no fire
13	snow/ice flag	1	yes
		0	no
14	is adjacent to cloud	1	yes
		0	no
15	Salt pan	1	yes
		0	no
16	Snow mask	1	yes
		0	no

Note that in R, the first bit will be referred as "1", whereas in many other languages (e.g. Python) it will follow the values as shown in table.

To interpret the pixel level QA values, we need to convert them from decimal to binary format. The `luna` package offers a function for this conversion and create a mask from the QA band. The user need to specify a matrix ("qabits") with the start and end of the quality assessment (QA) bits considered, and specify a list ("reject") with the values to be rejected (in the image) matching the rows in `qabits`. Following the table above, we will define the "reject" values to exclude pixels affected by cloud and cloud shadow.

```r
from <- c(1,3,11,14)
to   <- c(2,3,11,14)
reject <- c("01,10", "1", "1", "1")

qa_bits <- cbind(from, to, reject)
qa_bits
##      from to   reject
## [1,] "1"  "2"  "01,10"
## [2,] "3"  "3"  "1"
## [3,] "11" "11" "1"
## [4,] "14" "14" "1"
```

Pixels with bits 1 and 2 with values "01" or "10" will be rejected. All other combinations ("00" and "11" in this case) are not rejected.

We use the downloaded MODIS file that, in a previouse step, we saved in the `datadir` directory.

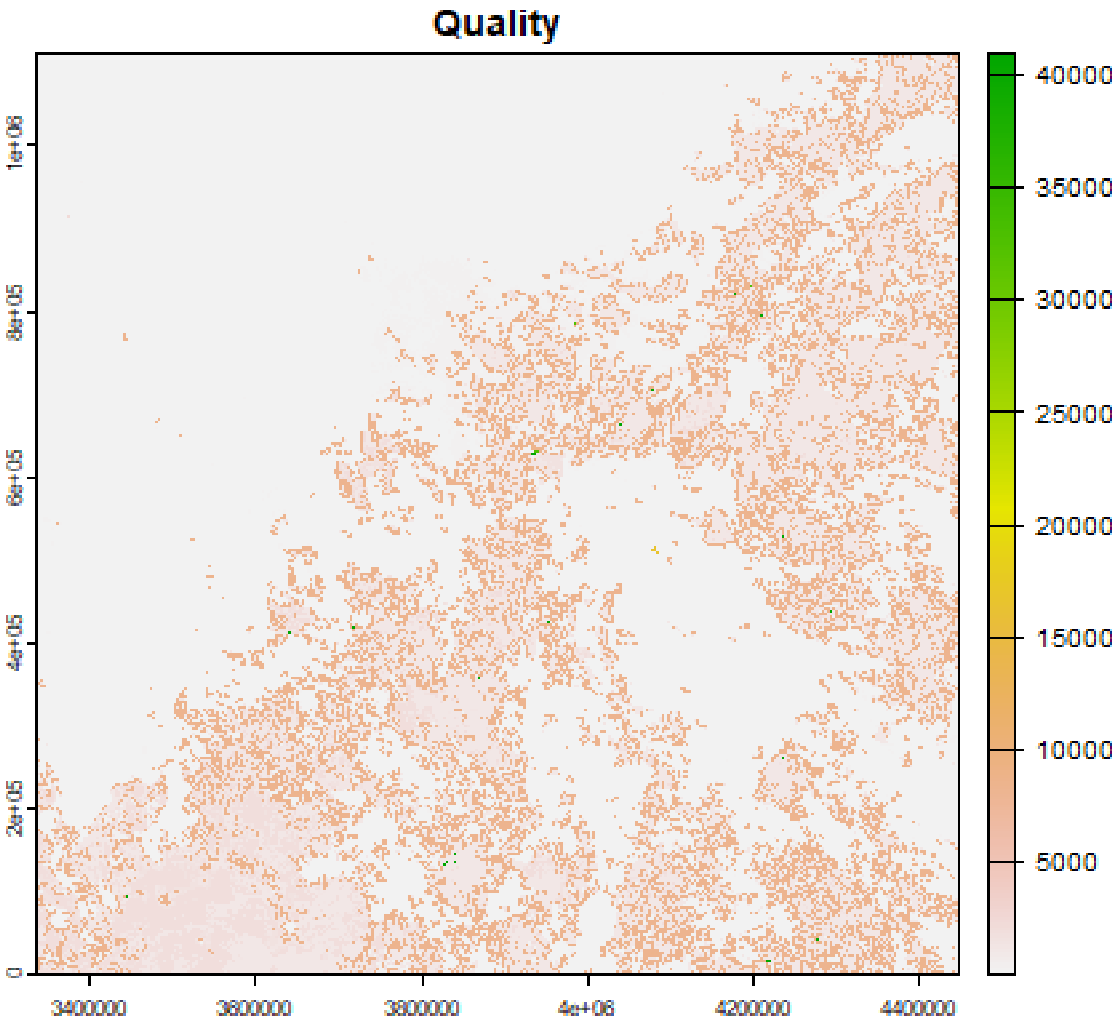
```r
library(terra)
datadir <- file.path(dirname(tempdir()), "_modis")
mf <- file.path(datadir, "MOD09A1.A2009361.h21v08.061.2021149144347.hdf")

r <- rast(mf)
```

Generate the quality mask. We will use band 12 `sur_refl_state_500m` that has the quality data.

```r
qc <- r[[12]]

plot(qc, main = "Quality")
```

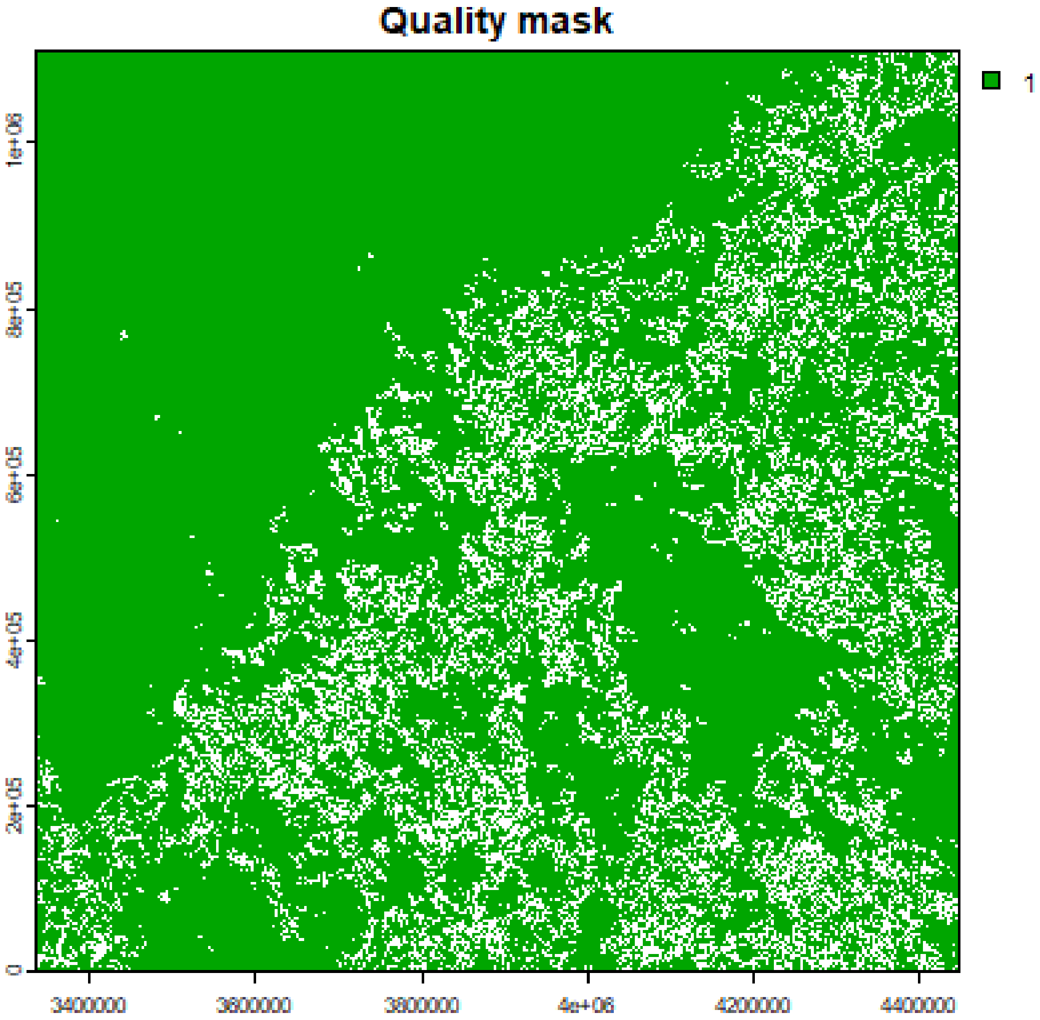


The luna package has a `modis_mask` method to create a mask from the quality band and the parameters defined above.

```
library(luna)
##
## Attaching package: 'luna'
## The following object is masked from 'package:base':
##
##     version

quality_mask <- modis_mask(qc, 16, qa_bits)

plot(quality_mask, main="Quality mask")
```

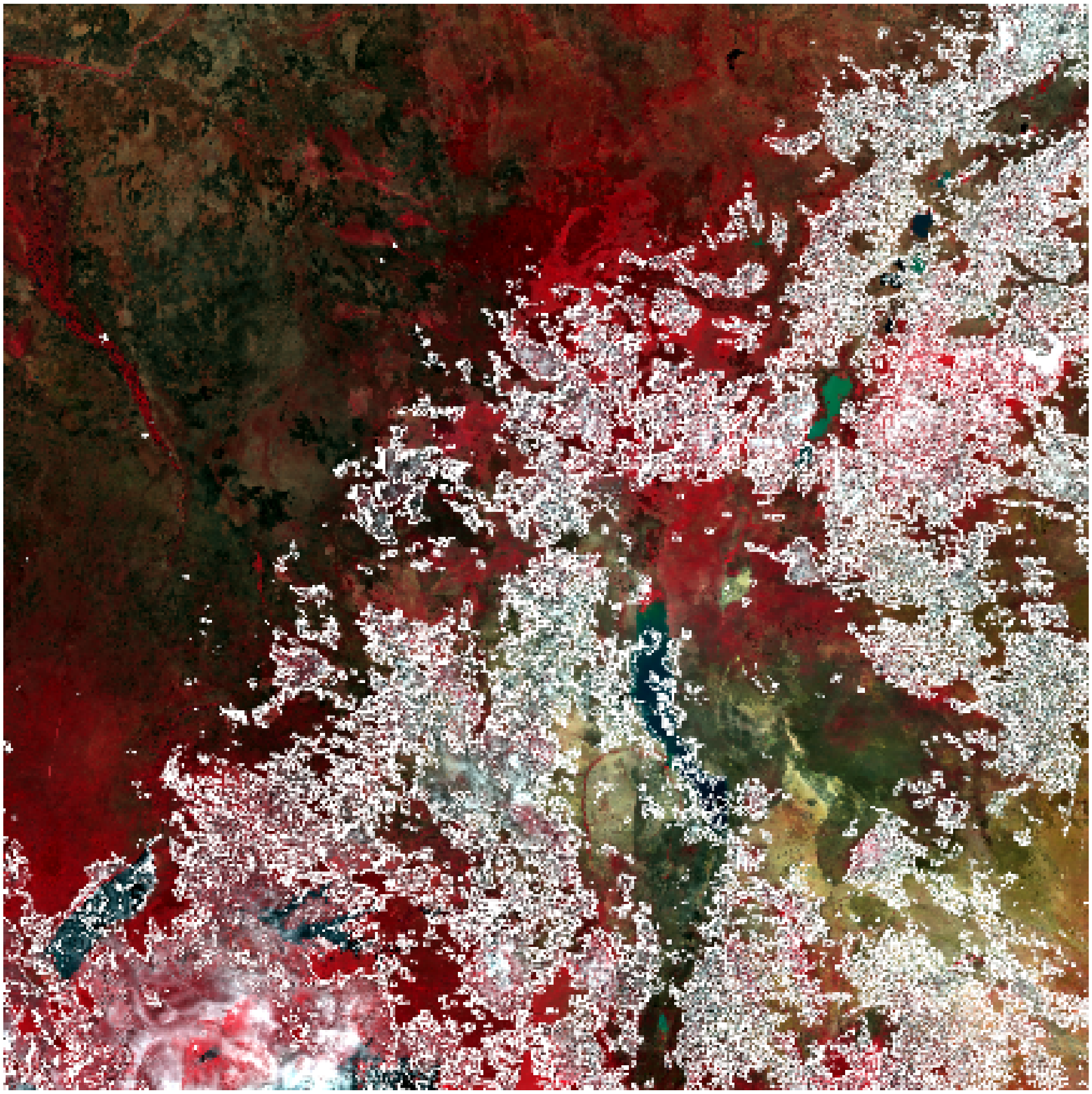


The plot shows the pixels we want to retain. Now that we have the quality mask, we can apply it to all the bands. It is always useful to visually check the result of the masking as many times pixels could be wrongly flagged as "poor" quality. Residual noises can be filtered in subsequent gap-filling and smoothing operations.

```
rmask <- mask(r, quality_mask)
```

And we can plot the results, here as a "false color composite" (NIR:Red:Green)

```
plotRGB(rmask, r = 2, g = 1, b = 4, main='False color composite', stretch="lin")
```



Finally we save the result after cloud masking.

CHAPTER

FIVE

PROCESSING

Here we crop an image to get an exact area of interest, and then we compute the Normalized Difference Vegetation Index (NDVI).

Get the polygon boundary

```
library(terra)
## terra 1.7.62
library(geodata)
ken <- gadm(country="KEN", level=1, path=".")
pol <- ken[ken$NAME_1 == "Marsabit", ]
```

Change vector boundary coordinate reference system, so that it matches that of the MODIS data.

```
datadir <- file.path(dirname(tempdir()), "_modis")
mf <- file.path(datadir, "modis_qualmasked.tif")
rmask <- rast(mf)

prj <- crs(rmask)

prj
## [1] "PROJCRS[\"unnamed\",\n    BASEGEOGCRS[\"Unknown datum based upon the custom␣
↪spheroid\",\n        DATUM[\"Not specified (based on custom spheroid)\",\n           ␣
↪ELLIPSOID[\"Custom spheroid\",6371007.181,0,\n                LENGTHUNIT[\"metre\",1,\
↪n                    ID[\"EPSG\",9001]]]],\n        PRIMEM[\"Greenwich\",0,\n         ␣
↪  ANGLEUNIT[\"degree\",0.0174532925199433,\n                ID[\"EPSG\",9122]]]],\n    ␣
↪CONVERSION[\"Sinusoidal\",\n        METHOD[\"Sinusoidal\"],\n        PARAMETER[\
↪"Longitude of natural origin\",0,\n            ANGLEUNIT[\"degree\",0.
↪0174532925199433],\n            ID[\"EPSG\",8802]],\n        PARAMETER[\"False easting\
↪",0,\n            LENGTHUNIT[\"metre\",1],\n            ID[\"EPSG\",8806]],\n        ␣
↪PARAMETER[\"False northing\",0,\n            LENGTHUNIT[\"metre\",1],\n            ID[\
↪"EPSG\",8807]]],\n    CS[Cartesian,2],\n        AXIS[\"easting\",east,\n            ␣
↪ORDER[1],\n            LENGTHUNIT[\"metre\",1,\n                ID[\"EPSG\",9001]]],\n␣
↪        AXIS[\"northing\",north,\n            ORDER[2],\n            LENGTHUNIT[\"metre\
↪",1,\n                ID[\"EPSG\",9001]]]]"

poly <- project(pol, prj)
```

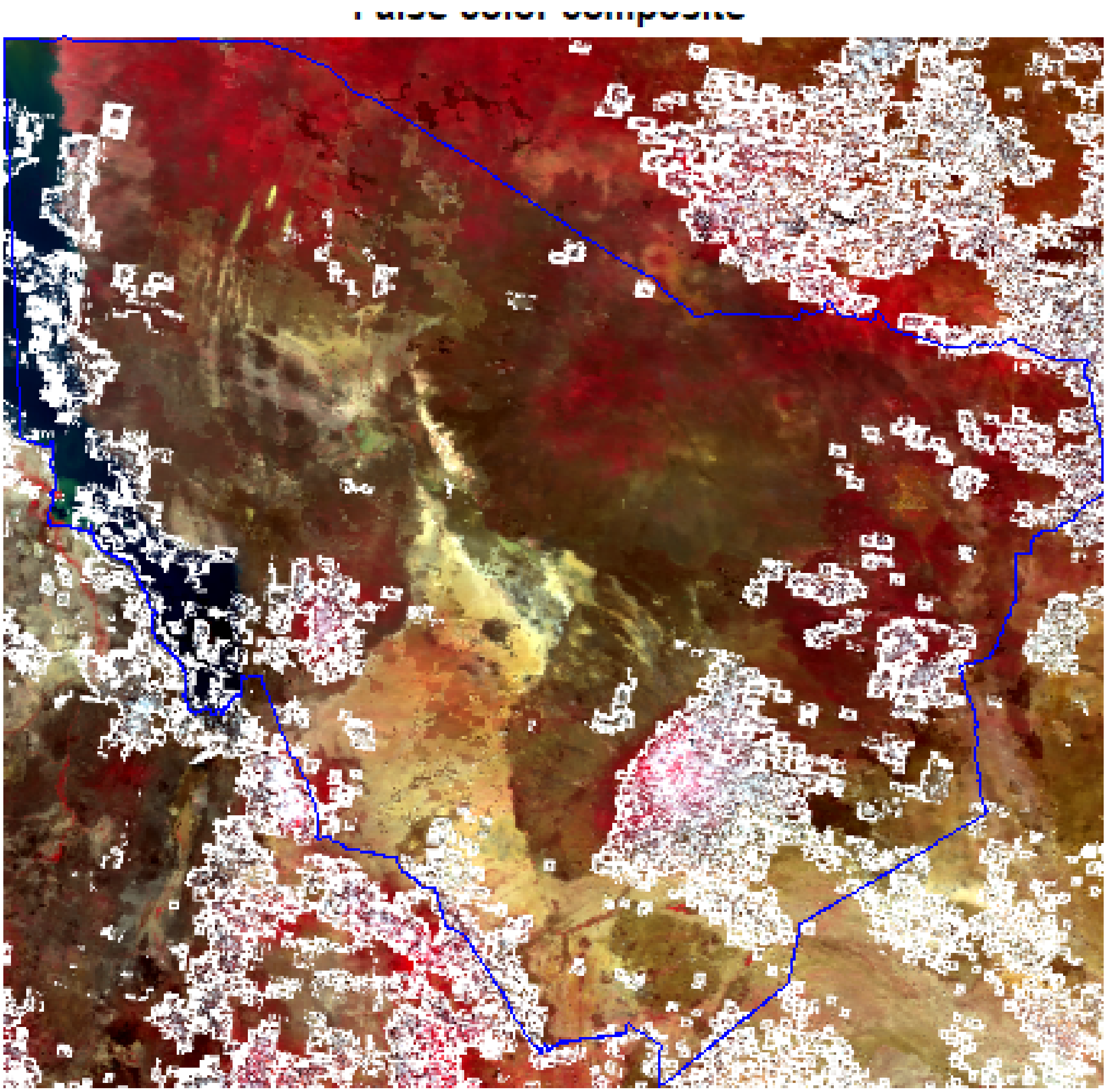
Question: Why do not we change the coordinate system of the MODIS data?

Crop the image using the transformed vector boundaries.

```
rcrop <- crop(rmask, poly)
```

Plot cropped MODIS and add the boundary. We use plotRGB to make a *false color composite* (near-infrared, red, green)

```
plotRGB(rcrop, r = 2, g = 1, b = 4,  main = "False color composite", stretch = "lin" )
lines(poly, col="blue")
```



5.1 NDVI

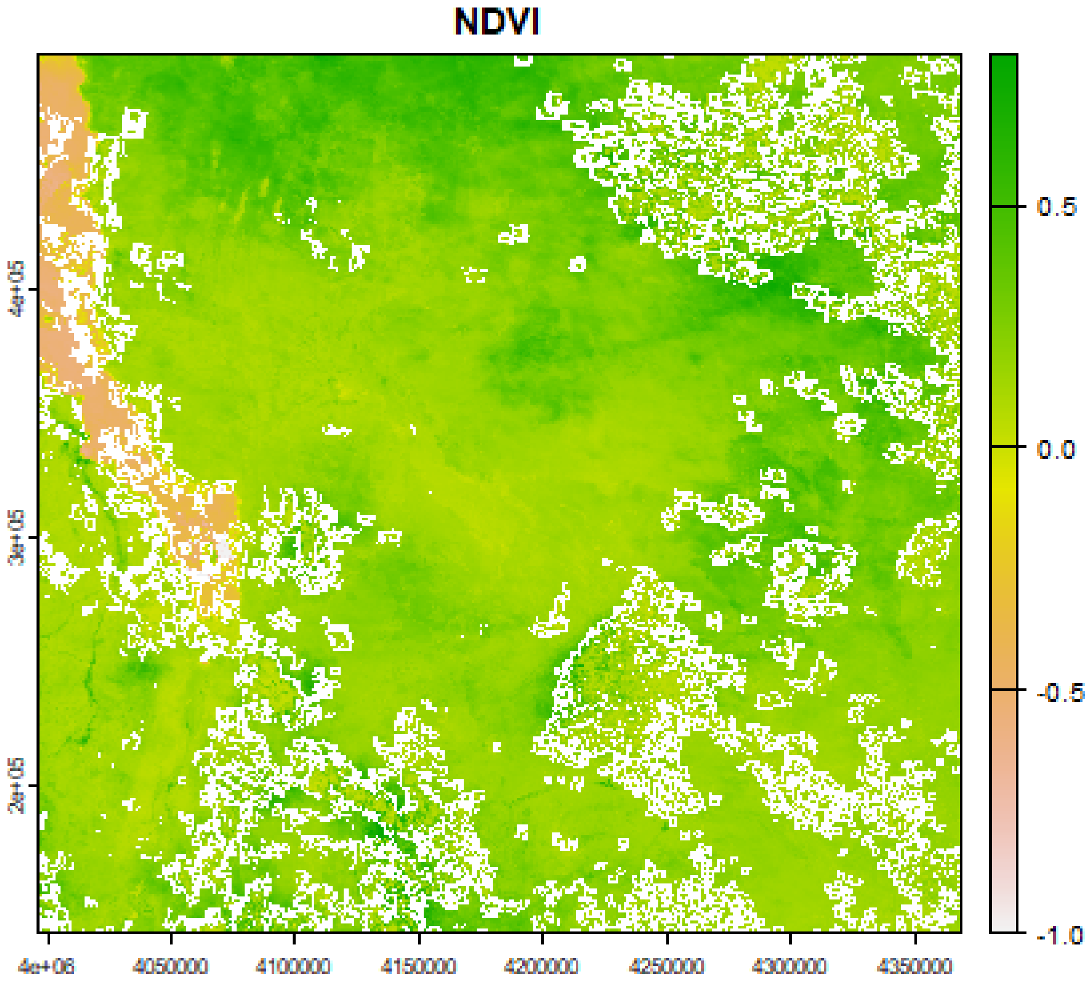
We have so far masked out bad quality pixels and clipped the image to area of interest extents. Let us now use the processed image to compute an index measure. The Normalized Difference Vegetation Index (NDVI) is a common measure of greenness. It is computed as follows

$$NDVI = \frac{(NIR - Red)}{(NIR + RED)}.$$

We expect the reflectance to be between 0 (very dark areas) and 1 (very bright areas). Due to various reasons, there may values slightly outside this rnage. First clamp values of the image between 0 and 1.

```
r <- clamp(rcrop, 0, 1)
```

```
ndvi <- (r[[2]] - r[[1]]) /(r[[2]] + r[[1]])
plot(ndvi, main="NDVI")
```



Exercise Write a function to compute `NDVI` type two-band spectral indices and compute `NDVI` using the function.