Application of Land Surface Temperature in Drought Monitoring

Prepared by: Begum Rushi, NASA SERVIR Science Co-ordination Office, Huntsville, AL
e-mail: begumrabeya.rushi@nasa.gov

6.1 Introduction

Drought mitigation and response require continuous monitoring of metrological information. Due to unavailability of temporally and spatially consistent ground observations, satellite remote sensing data enable such monitoring. Land Surface Temperature (LST) products from the Moderate Resolution Imaging Spectroradiometer (MODIS) sensor on board of the Terra and Aqua satellites can be used in drought monitoring where field observation data are scarce. In previous exercises, we have learned about SPI based on precipitation. It has also been discussed about temperature condition index (TCI) and how this indicator can be correlated with vegetation condition index (VCI). In this exercise, we will discuss about a more comprehensive drought index called Vegetation-Temperature-Precipitation Condition Index which is popularly known as Synthesized Drought Index (SDI) to monitor drought condition at a monthly time-step. MODIS LST, NDVI and Integrated Multi-satellite Retrievals for GPM (IMERG) precipitation products have been used in SDI calculation which gives a proper idea of precipitation deficits, soil thermal stress and vegetation growth status in drought process. Therefore, this method contributes to the understanding of drought in a comprehensive way.

6.2 Description of Method

6.2.1 Indicator Calculations

TCI, VCI as well as Precipitation Condition Index (PCI) are required to calculate SDI. Since we know from the previous discussion that:

$$VCI = \frac{NDVI - NDVI_{\min}}{NDVI_{max} - NDVI_{min}}$$

Where NDVI is the corresponding pixel value after filtering. $NDVI_{max}$ and $NDVI_{min}$ are respectively maximum NDVI and minimum NDVI of the corresponding pixels in same month for the entire NDVI records (In this study, 2015–2017). Here, original index value is required. That is why percentage has not been calculated.

For this study, TCI is based on MODIS LST and it can be calculated as:

$$TCI = \frac{LST_{max} - LST)}{LST_{max} - LST_{min}}$$

where LST, LST_{max} and LST_{min} are the values of LST, maximum LST and minimum LST of each pixel respectively in same month during the study period (2015-2017).

Another component, PCI, can calculated as:

$$PCI = \frac{PRCP - PRCP_{\min}}{PRCP_{max} - PRCP_{min}}$$

where PRCP, $PRCP_{max}$ and $PRCP_{min}$ are the values of precipitation, maximum precipitation and minimum precipitation of each pixel respectively in same month during the study period (2015-2017).

6.2.2 Principal component Analysis

After calculating these indices, principal component analysis (PCA) has been applied to calculate SDI. Systematic dimensionality of a data set can be reduced using this PCA method (Richards and Jia, 2006; Deng et al., 2008). It creates an orthogonal transformation of possibly correlated inputs to create principal components as new variables (Eastman and Fulk, 1993; Avena et al., 1999). These new variables are chosen to represent the maximum possible extent of variability contained in the original data (Valdés-Pineda et al., 2016). Singular Value Decomposition method has been used in this case in which a series of new data (called eigenchannels) are computed by multiplying the eigenvector for the original input data. The VCI, TCI and PCI for each month are considered as inputs and this method creates the same number of principal component bands. Since the first principal component (PC-1) always contains more than 75% information from inputs, here PC-1 is considered as SDI. In this method, SDI index has been created for each month for the entire study period (2015-2017). The wider the temporal span will be, the more results will be representative of the actual condition. For this training a short time period (Year 2015-2017) has been considered.

Prerequisites:

- Installation of R and RStudio

  (Software and package installation guide has been provided in **Appendix-A**)
- Google Earth Engine (GEE) Account (Optional) (Instruction to work on GEE environment has been provided in **Appendix-B)**

Data Required:

- Daily LST (MOD11A1) (Wan, 2015)
- 8-Day LST (MOD11A2) (Wan, 2015)
- Monthly Vegetation Indices (NDVI) Data (MOD13A3) (Didan, 2015)
- Monthly Precipitation Data (IMERG) (Huffman, 2017)

6.3 Methodology

The flowchart of methodology is provided below:

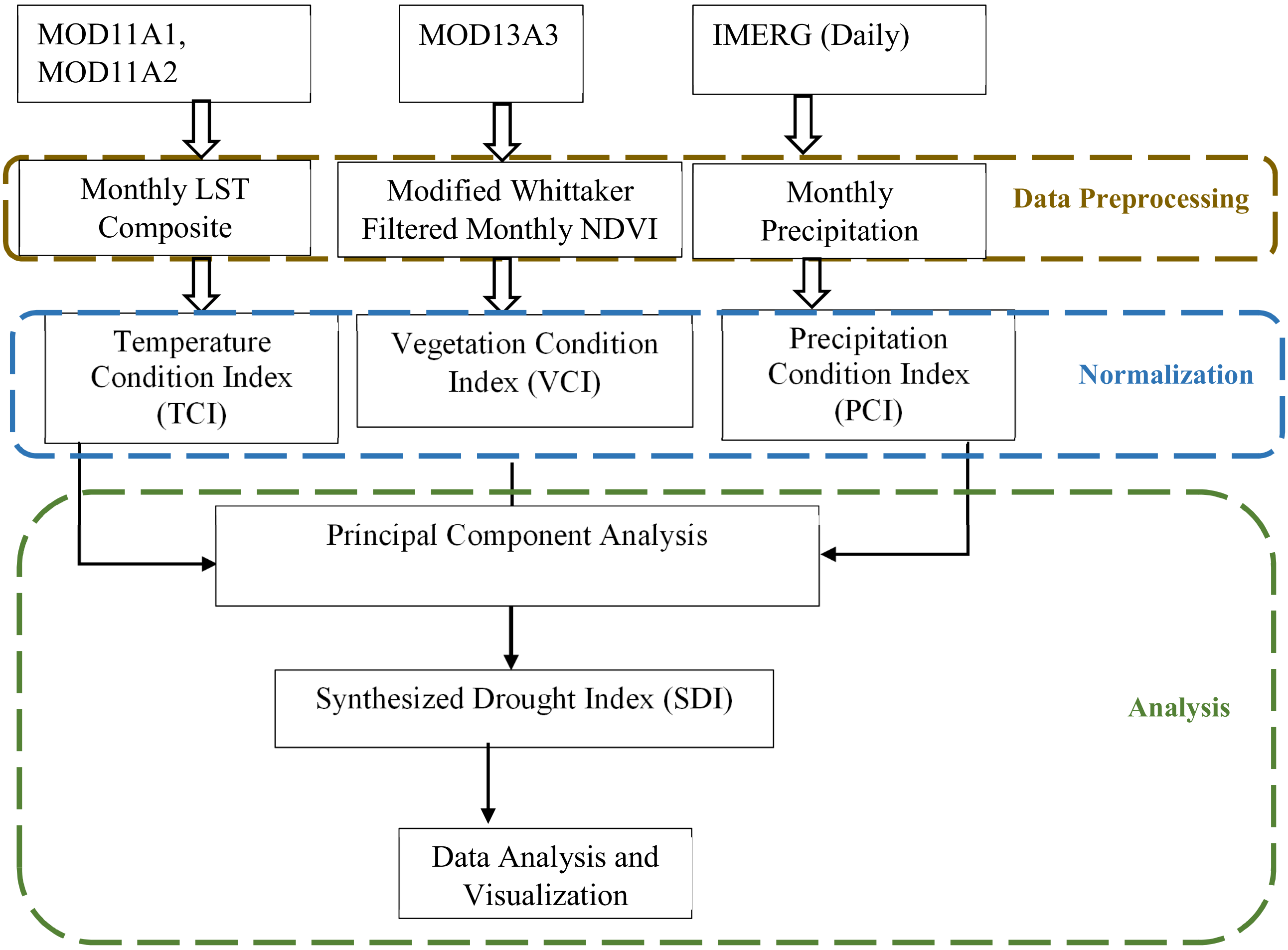


Figure 6.1: Flowchart of Methodology

6.4 Downloading and Processing Data into Local Machine

6.4.1 Downloading Data into Local Machine

IMERG precipitation data can be downloaded from different sources (e.g. DAAC, ClimateSERV). In this case, monthly precipitation data has been used.

The most up-to-date and recent MODIS based LST, NDVI data can be accessed through NASA data archive called The Land Processes Distributed Active Archive Center (LP DAAC). In the scripts folder, (e.g. directory: C:\Training_Materials\Scripts_BD_Workshop\) a script named **download_MODIS.R** which can be used to download MODIS from LPDAAC. An user can parse command-line arguments into the script. Few default arguments have been set up beforehand, but by providing arguments possible customizations can be made. Possible arguments can be:

-i Input Folder Directory (e.g. C:/test/MODIS_Input/). If you don't provide any input, it will create a sub-folder named 'MODIS_Input' under 'test' folder in your C drive. In this folder, it will download raw hdf formatted data in your local machine.

-r Output folder directory. (e.g. C:/test/MODIS_processed/). If you don't provide any input, it will create a sub-folder named 'MODIS_ processed under 'test' folder in your C drive. In this folder, it will mosaic and crop tiles of interested area extent and produce results in Geotiff.

-s Start Date (integer) (e.g. 2017001)

-e End Date (integer) (e.g. 2017032)

- p Product ID (e.g. MOD13A2)

- a Country Name (e.g. Bangladesh)

By typing the following, an user can get a comprehensive guide about all the necessary arguments.

Rscript download_MODIS.R -h

To run the script from command line, one has to type:

Rscript download_MODIS.R -i Input directory -r Output directory -s Start Date -e End Date -p Product ID -a Country name

Example:

Rscript download_MODIS.R -i C:/test/MODIS_Input/ -r C:/test/MODIS_processed/ -s 2017001 -e 2017032 -p MOD13A2 -a Bangladesh

For users who would like to download data for a smaller area extent, there is another script named **download_MODIS_v2.R** where extents of the area of interest (latitude and longitude in decimal degrees) are needed to be parsed as arguments.

From command line, an user has to type:

Rscript download_MODIS_v2.R -i Input directory -r Output directory -s Start Date -e End Date -p Product ID -m Minimum Longitude -x Maximum Longitude -t Minimum Latitude -b Maximum Latitude

Example:

Rscript download_MODIS_v2.R -i C:/test/MODIS_Input/ -r C:/test/MODIS_processed/ -s 2017001 -e 2017032 -p MOD13A2 -m 88.08 -x 92.67 -t 20.67 -b 26.44

In these script, using multiple R packages (e.g. MODIS, rgdal, raster) data has been downloaded, processed and converted into geotiff format to use it for further analysis. All the arguments are not mandatory. If an user would like to keep everything as default or partially change some arguments, he/she would still be able to run the script.

6.4.2 Creating Temporal Composite of MODIS Product (LST, NDVI)

This study has been conducted at monthly time-step. Data inputs from varied temporal resolution were composited at monthly scale using a script named "**temporalcomposite_MODIS_product.R**". Since we used the monthly MODIS NDVI product in this study, no temporal conversion is not required for calculating VCI. If required, an user can use the same script to create temporal composite of MODIS NDVI at different time-step.

In this script, `temporalComposite` function has been used for necessary processing.

```
## create monthly mean value composites

monthly_composite<-temporalComposite(ndvi, cdoy, timeInfo = extractDate(cdoy, asDate =
                                                          TRUE)$inputLayerDates, interval = c(opt$time_step),
                        fun = mean, na.rm = TRUE)

```

Inputs at this step are geotiffs files which have been created from previous step. From command line, an user has to type:

Rscript temporalcomposite_MODIS_product.R

-i Input directory -r Output directory -f Output file name -t Time Step

Example:

Rscript temporalcomposite_MODIS_product.R

-i C:/test/MODIS_processed/temporalComposite/

-r C:/test/MODIS_Monthly_Composite/ -f Monthly_NDVI_Composite.tif -t month

If an user would like to have yearly or fortnight data, time step argument would "year" and "fortnight" respectively.

6.4.3 Creating Drought Indices

At this step, all indices would be calculated. Outputs would be saved in geotiff format. "Drought_indices_MODIS_product.R" script will be used. This script has four different segments.

a. At first segment, it will read all the monthly composite files (MODIS LST, MODIS NDVI, and IMERG Precipitation).
b. Next, it calculates TCI, VCI, PCI and SDI respectively and saves the results as geotiff files.
c. In this script, `princomp` function has been used to calculate the coordinates of all three indices on the principal components. There are several techniques to calculate principal components. Singular value decomposition has been used in this function.

```
pca.cal <- princomp(~raster.point$TCI+raster.point$VCI+raster.point$PCI, data = raster.point,
                    cor = F, scores = TRUE)
```

From command line, one has to type the following:

Rscript Drought_indices_MODIS_product.R -i Input directory -r Output directory -Y Start Year -N End Year

Example:

Rscript Drought_indices_MODIS_product.R

-i C:/Training_Materials/Monthly_Composite/

-r C:/ Training_Materials/Monthly_Composite/Indices/ -Y 2015 -N 2017

6.5 Analysis and Visualization of Drought Indices

The drought classification schemes for all the indices are given below:

Index	Index Value for the Most Favorable Condition	Index Value for the Least Favorable Condition
TCI	1	0
VCI	1	0
PCI	1	0
SDI	>0.5	<0.2

Figure 6.2 shows SDI values for the month of August (wet month) for Year 2015-2017:

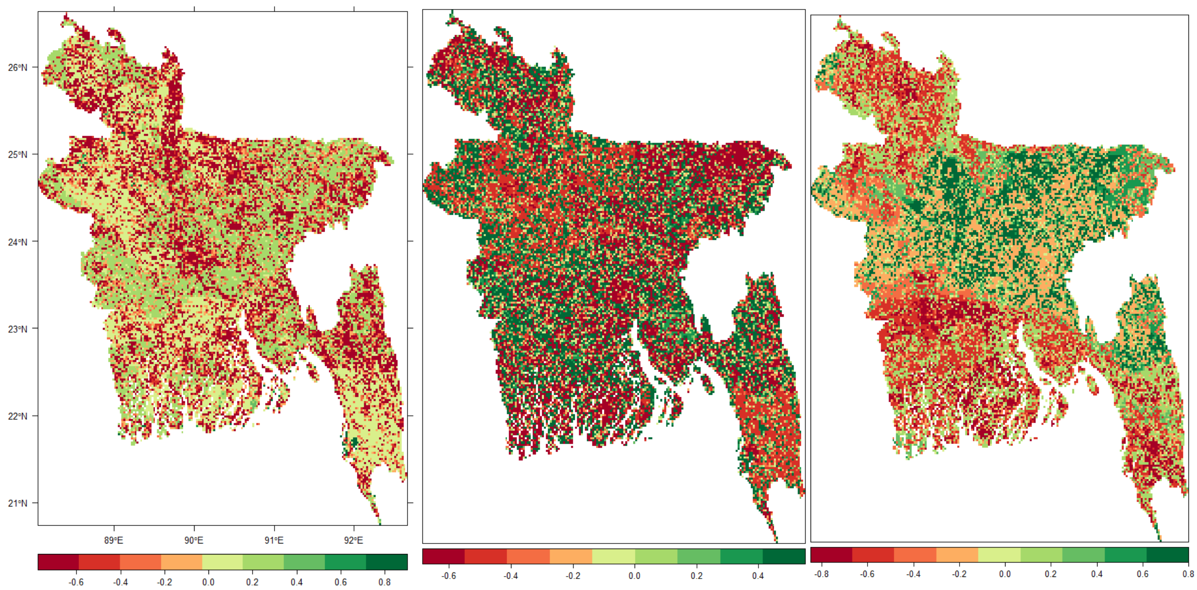


Figure 6.2: Drought Monitoring in Bangladesh using SDI for the Month of August (2015-2017 Respectively)

Whereas, figure 6.3 depicts SDI for the month of March (dry month):

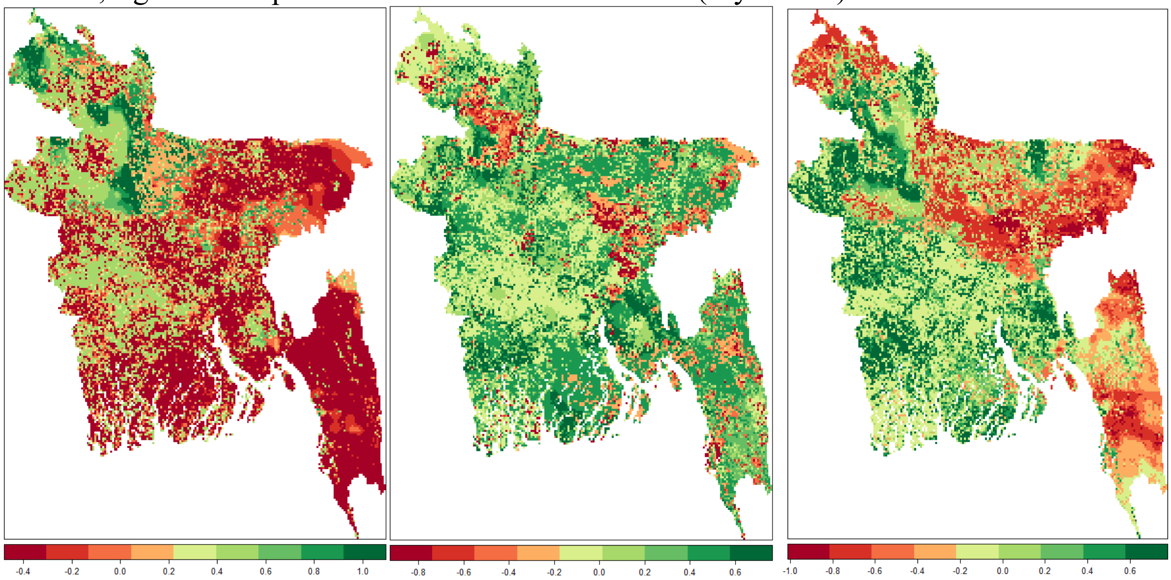


Figure 6.3: Drought Monitoring in Bangladesh using SDI
for the Month of March (2015-2017 Respectively)

If we consider, a small area such as Manda upazilla, we witness presence of extreme drought condition under SDI monitoring for the month of April in 2016.

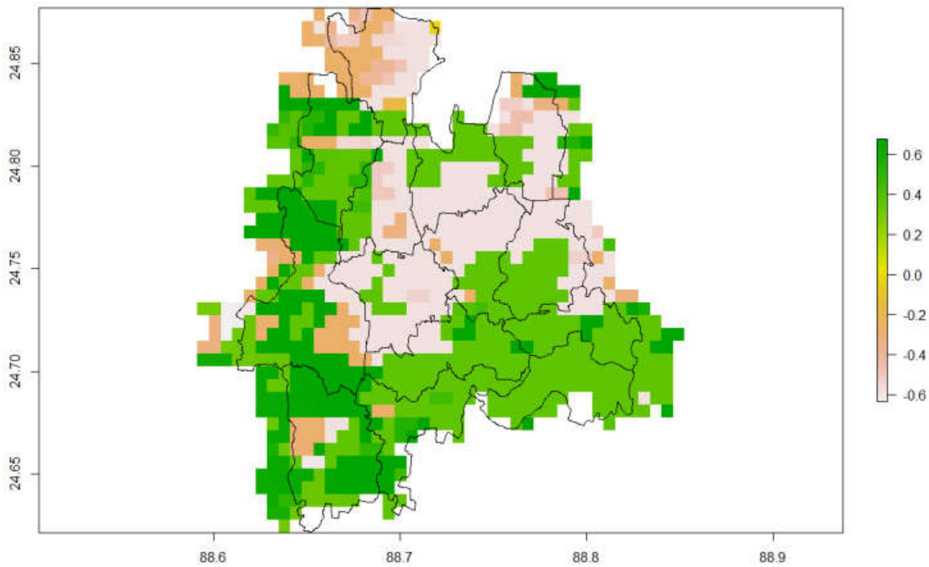
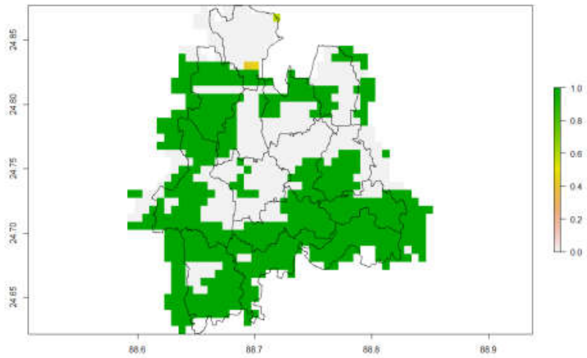
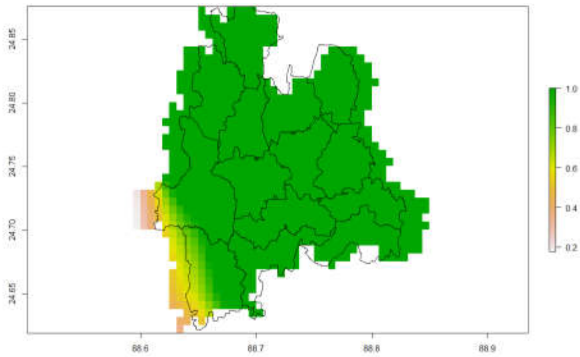


Figure 6.3: Drought Monitoring using SDI for Manda Upazilla

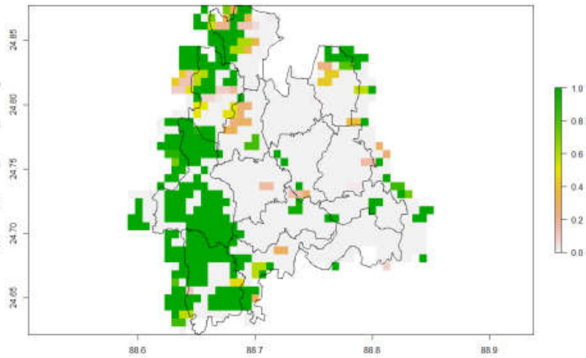
If we compare Figure 6.3 with Figure 6.4 which depicts TCI, PCI and VCI indices, SDI is able to capture drought condition more specifically compared to other three indices.



TCI



PCI



VCI

Figure 6.4: SDI for Month of April in 2016

Graphical representations of the results depend on the researcher as well as objectives of the study. Depending on the availability and user preference, QGIS, ArcGIS as well as programming tools such as R, python can be used to create maps using outputs from this study. For this study, R has been used to analyze and visualize products. All these scripts as well as few preprocessing scripts can be found at **GitHub repository** of this training.

References

Avena, G.C., Ricotta, C., Volpe, F., 1999. The influence of principal component analysis on the spatial structure of a multispectral dataset. International Journal of Remote Sensing 20 (17), 3367–3376.

Deng, J.S., Wang, K., Deng, Y.H., Qi, G.J., 2008. PCA-based land-use change detection and analysis using multitemporal and multisensor satellite data. International Journal of Remote Sensing 29 (16), 4823–4838

Didan, K.. MOD13A2 MODIS/Terra Vegetation Indices 16-Day L3 Global 1km SIN Grid V006. 2015, distributed by NASA EOSDIS LP DAAC, https://doi.org/10.5067/MODIS/MOD13A2.006

Eastman, J.R., Fulk, M., 1993. Long sequence time-series evaluation using standardized principal components. Photogrammetric Engineering and Remote Sensing 59 (6), 991–996.

Huffman, G., 2017. GPM IMERG Final Precipitation L3 1 day 0.1 degree x 0.1 degree V05, Edited by Andrey Savtchenko, Greenbelt, MD, Goddard Earth Sciences Data and Information Services Center (GES DISC), [Accessed on March 28 2018] 10.5067/GPM/IMERGDF/DAY/05

Richards, J.A., Jia, X., 2006. Remote Sensing Digital Image Analysis: An Introduction. Springer Verlag, Berlin, Germany

Valdés-Pineda, R., Demaría, E.M., Valdés, J.B., Wi, S. and Serrat-Capdevila, A., 2016. Bias correction of daily satellite-based rainfall estimates for hydrologic forecasting in the Upper Zambezi, Africa. Hydrology and Earth System Sciences, 437, pp.1-28.

Wan, Z., S. Hook, G. Hulley. MOD11A1 MODIS/Terra Land Surface Temperature/Emissivity Daily L3 Global 1km SIN Grid V006. 2015, distributed by NASA EOSDIS LP DAAC, https://doi.org/10.5067/MODIS/MOD11A1.006

Wan, Z., S. Hook, G. Hulley. MOD11A2 MODIS/Terra Land Surface Temperature/Emissivity 8-Day L3 Global 1km SIN Grid V006. 2015, distributed by NASA EOSDIS LP DAAC, https://doi.org/10.5067/MODIS/MOD11A2.006

Appendix-A:

A.1 R and R Studio Installation

Please follow the steps below for R and R Studio installation:

Windows

If you already have R and RStudio installed

- Open RStudio, and click on "Help" > "Check for updates". If a new version is available, quit RStudio, and download the latest version for RStudio.
- To check the version of R you are using, start RStudio and the first thing that appears on the terminal indicates the version of R you are running. Go on the CRAN website and check whether a more recent version is available. If so, please download and install it. You may also want to consider removing your old version of R. You can check here for more information.

If you don't have R and RStudio installed

- Download R from the CRAN website.
- Run the .exe file that was downloaded
- Go to the RStudio download page
- Under *Installers* select **RStudio x.yy.zzz - Windows XP/Vista/7/8** (where x, y, and z represent version numbers)
- Double click the file to install it
- Once it's installed, open RStudio to make sure it

MacOS X

If you already have R and RStudio installed

- Open RStudio, and click on "Help" > "Check for updates". If a new version is available, quit RStudio, and download the latest version for RStudio.
- To check the version of R you are using, start RStudio and the first thing that appears on the terminal indicates the version of R you are running. Go on the CRAN website and check whether a more recent version is available. If so, please download and install it. You may also want to consider removing your old version of R. You can check here for more information.

If you don’t have R and RStudio installed

- Download R from the CRAN website.
- Select the .pkg file for the version of OS X that you have and the file will download
- Double clik on the downloaded file to install R
- Go to the RStudio download page
- Under *Installers* select **RStudio x.yy.zzz - Mac OS X 10.6+ (64-bit)** (where x, y, and z represent version numbers)
- Double click the file to install RStudio
- Once it’s installed, open RStudio to make sure it works

Linux

- Follow the instructions for your distribution from CRAN, they provide information to get the most recent version of R for your distribution. For most distributions, you could use your package manager (e.g., for Debian/Ubuntu run sudo apt-get install r-base, and for Fedora sudo yum install R), but the versions provided by this approach are usually out of date. In any case, make sure you have at least R 3.3.1
- Go to the RStudio download page
- Under *Installers* select the version that matches your distribution, and install it with your preferred method (e.g., with Debian/Ubuntu sudo dpkg -i rstudio-x.yy.zzz-amd64.deb at the terminal).
- Once it’s installed, open RStudio to make sure it works

A.2. Package Installation

A.2.1 Package Installation (Online)

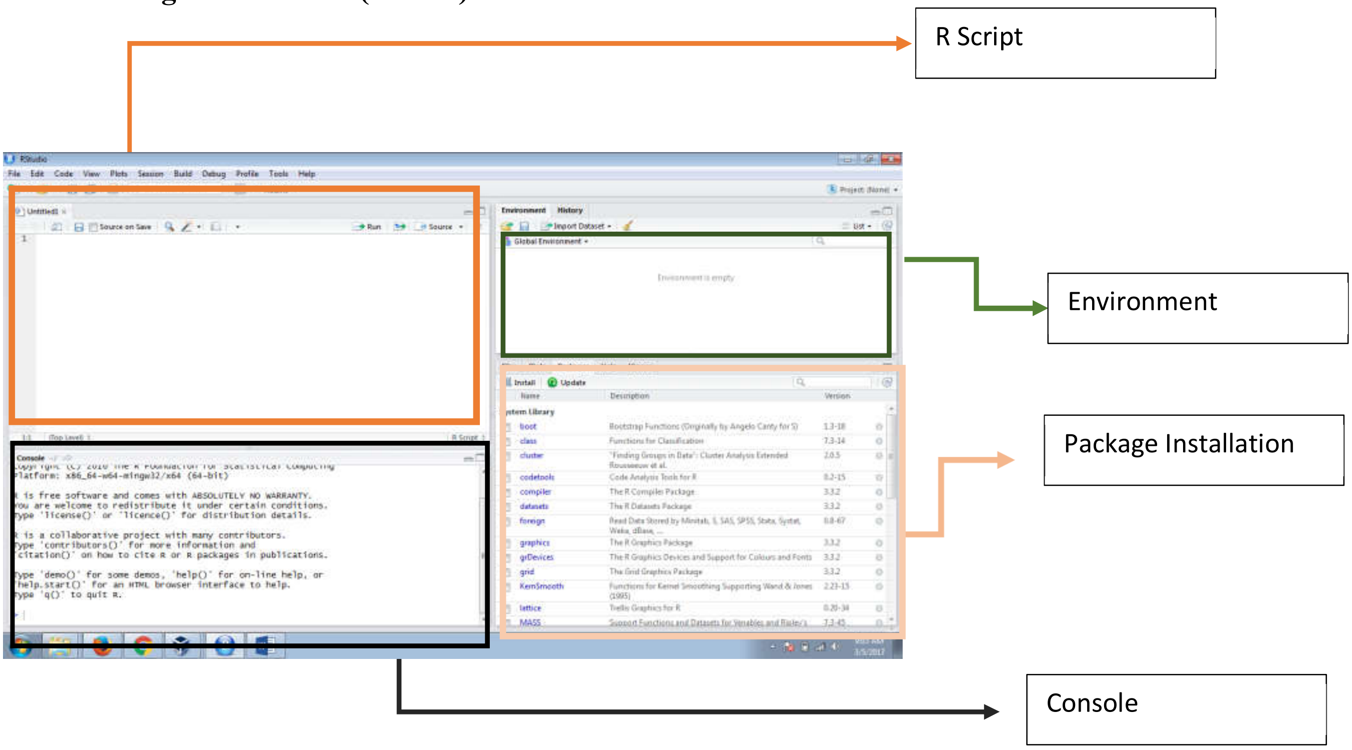


Figure A.1: RStudio

R Studio has four segments/sections (Figure A.1). After downloading R and R Studio, click "Install" under Package Installation section and search for necessary packages. To run the bias correction scripts, the following packages are needed:

- lubridate
- mapview
- MODIS
- optparse
- rts
- raster
- RCurl
- gdalUtils
- rgdal
- signal

- zoo

Or, in console section you can write:

install.packages(c("package name"))

After installing all packages your bias correction model is ready to run.

A.2.2 Package Installation (Offline)

For this training all pre-downloaded packages for the environment are available in win.binary format under ../Training_Materials/Rpackages/. There is a script named "**Rpackages_install.R**" under ../Training_Materials/R_extra/. If you run the script from command line, you would be able to install all the packages without internet connection.

From command line, an user has to type:

Rscript Rpackages_install.R

Before running the script, one has to make sure that working directory in the script is set up to the folder where all packages are available.

```
setwd("C:/Training_Materials/Rpackages/")
```

Appendix – B: Cloud based Data Access and Processing

➢ Accessing Satellite Product through Google Earth Engine (GEE)

Google Earth Engine (GEE) is a web-based platform where users are able to access data (Figure), perform data analysis and visualization using JavaScript API and Python API.

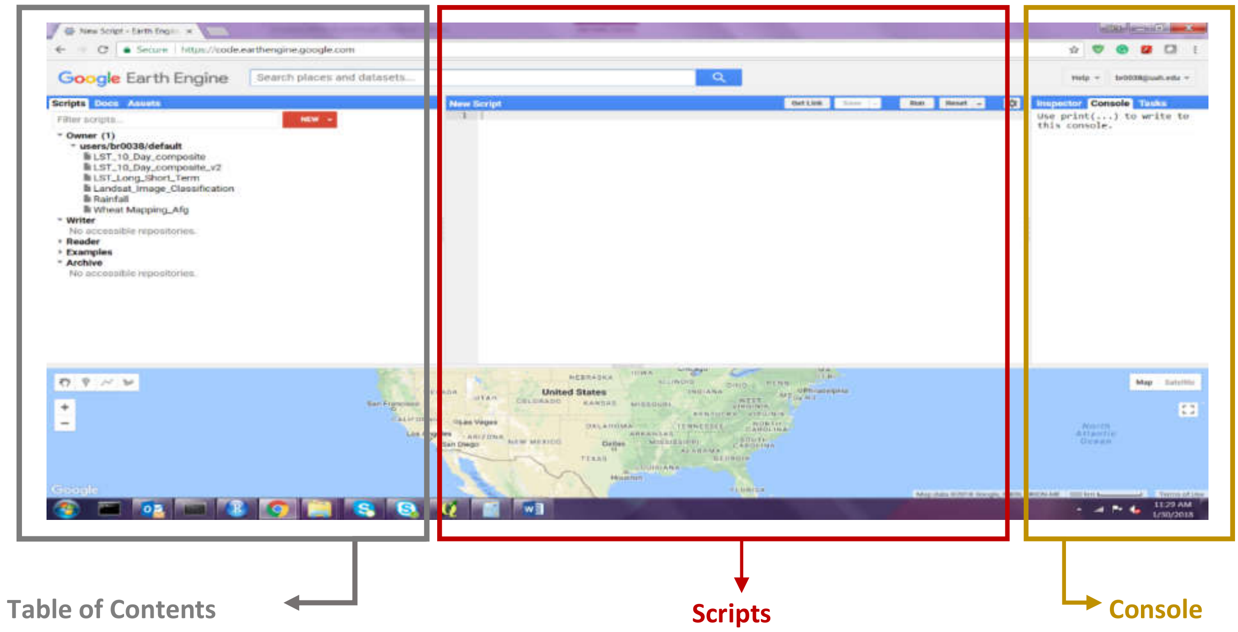


Figure B.1: GEE Platform

You can access different satellite and climate datasets in GEE. Details about alll GEE datasents can be found here: https://earthengine.google.com/datasets/

On the search menu, it requires to type the name of dataset you would like to access to import into your code editor for analysis.

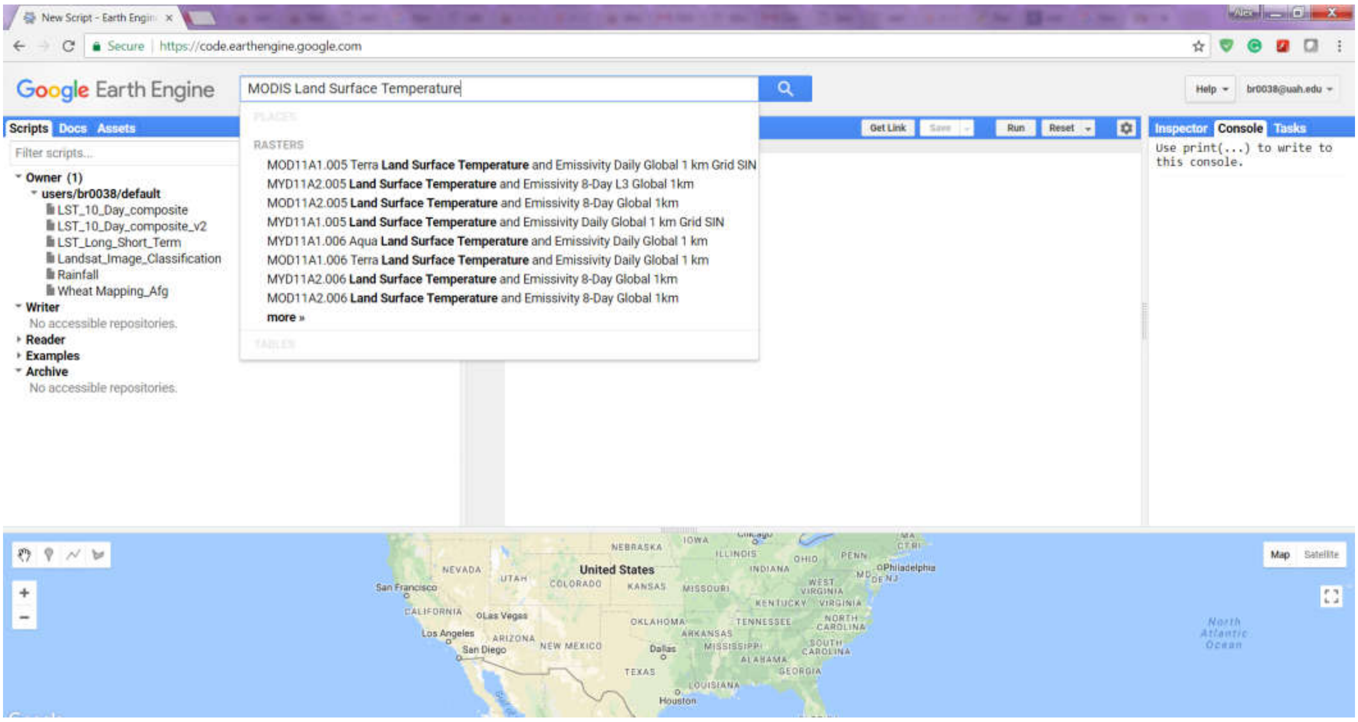


Figure B.2: Data Query in GEE

➢ Data Processing in Web Environment

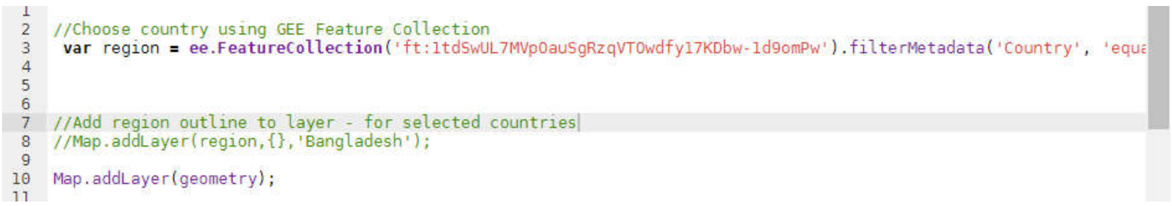
For instance, temporal composite at 10 day of MODIS LST has been shown here. You can access the code:

https://code.earthengine.google.com/78fe852423f9330b1b242e3580ded8a5

Different parts of the above code are explained below:

a. Enter the name of the MODIS Land Surface Temperature to load

b. Designating the interested region



c. Collecting bands and scale

```
// Collect bands and scale
var modisLSTday = ee.ImageCollection('MODIS/MOD11A1').select('LST_Day_1km');
var modisLSTnight = ee.ImageCollection('MODIS/MOD11A1').select('LST_Night_1km');
var modLSTday = modisLSTday.map(function(img) {
  return img.multiply(0.02).subtract(273.15).copyProperties(img,['system:time_start','system:time_end']);
});
var modLSTnight = modisLSTnight.map(function(img) {
  return img.multiply(0.02).subtract(273.15).copyProperties(img,['system:time_start','system:time_end']);
});
```

d. Selecting study period

```
// Select dates
var collection05night = ee.ImageCollection(modLSTnight.filterDate('2001-01-01', '2017-12-31'));
var collection05day = ee.ImageCollection(modLSTday.filterDate('2001-01-01', '2017-12-31'));
```

e. Cropping the data for the region

```
// Cropping the data for the region

var collection = collection05day.filterBounds(region);
```

f. Collecting Metadata Properties

```
// Get a list of all metadata properties.
var properties = collection.propertyNames();
print('Metadata properties: ', properties); // ee.List of metadata properties

// Get the date range of images in the collection.

var dates = ee.List(collection.get('date_range'));
var dateRange = ee.DateRange(dates.get(0), dates.get(1));
print('Date range: ', dateRange);
```

g. Calculating 10 Day Composite Data

ST_10_Day_composite Get Link Save Run Reset

```
//Calculating the 10 Day Composite

var startyear = 2001;
var endyear = 2017;
// Run the loop
for (var yr= startyear; yr <= endyear; yr = yr + 1) {
  var startdate = 1;
  //var enddate = 365;
  var enddate = ((yr%4)===0) ? 366 : 365;
  //print('This is enddate ' +enddate);

  for (var dt= startdate; dt <= enddate; dt = dt + 10) {

  // Use ee.Filter.calendarRange to filter by year and month
  var img = collection.filter(ee.Filter.calendarRange(yr,yr,'year'))
  .filter(ee.Filter.calendarRange(dt,dt+9,'day_of_year'));
  // reduce image collection with mean()
  if (dt == 1){
    var mean = img.mean().rename('LST_'+dt+'_' +(dt+9)+'_'+yr);
  }
  else {
    mean = mean.addBands(img.mean().rename('LST_'+dt+'_' +(dt+9)+'_'+yr))
  }
  }
  print(mean)

```

h. Exporting the output

```

 // Export the image, specifying scale and region.
  Export.image.toAsset({
  image: mean,
  description: 'LST_10Day_' + yr,
  assetId: 'projects/servir-hkh/MODIS_LST_10_Day_Composite/LST_10Day_'+yr,
  scale: 1000,
  region: geometry,
  maxPixels:3E11,
  pyramidingPolicy: {
    '.default': 'mean',
  }
  });

 }


```

- **Outputs**

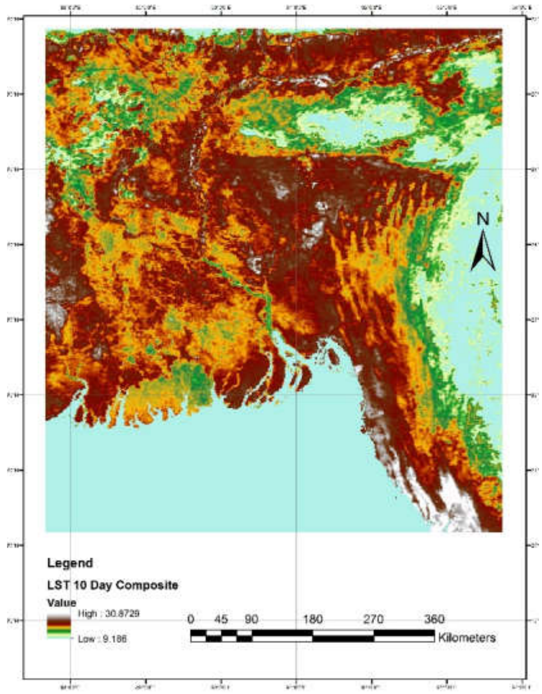


Figure B.3: LST 10 Day Composite for 2017 for Bangladesh

Code is designed in a way that 10 days composites of each year are stacked in a single raster brick. Suppose for 2001, 36 raster files are generated for 10 day composites of LST. Final product for 2001 (LST_10Day_2001) is a raster brick (stack) comprising 36 raster files. Next, we will take the output a further analyse the data in QGIS.