National Aeronautics and Space Administration
Goddard Earth Science Data Information and Services Center (GES DISC)

README Document for Annual Summary of Artificial Light At Night from VIIRS/S-NPP at CONUS County and Census Tract V1

Last Revised April 3, 2023

Goddard Earth Sciences Data and Information Services Center (GES DISC)
http://disc.gsfc.nasa.gov
NASA Goddard Space Flight Center
Code 610.2
Greenbelt, MD 20771 USA

Prepared By:

Qian Xiao

Name
Department of Epidemiology,
Human Genetics and Environmental Sciences,
The University of Texas Health Science Center
in Houston

April 3, 2023

Date

Reviewed By:

Feng Ding April 3, 2023

Reviewer Name
GES DISC
GSFC Code 619

Date

Goddard Space Flight Center
Greenbelt, Maryland

Revision History

Revision Date	Changes	Author
4/3/2023	Initial Version	Qian Xiao

Table of Contents

1.0 Introduction

This document provides detailed information about the satellite-based data on artificial light at night (ALAN). We used the Suomi National Polar-orbiting Partnership (S-NPP) Visible Infrared Imaging Radiometer Suite (VIIRS) Day/Night Band (DNB) nighttime lights (NTL) product (VNP46A4, DOI: 10.5067/VIIRS/VNP46A4.001) in NASA's Black Marble suite to derive annual summary of ALAN levels throughout the Contiguous US (CONUS) at both county and tract level for the period of 2012-2020.

1.1 Dataset/Mission Instrument Description

The yearly VNP46A4 data products are retrieved from the NASA Level-1 and Atmosphere Archive & Distribution System Distributed Active Archive Center (https://ladsweb.modaps.eosdis.nasa.gov/missions-and-measurements/products/VNP46A4/). Corresponding user guide can be found at the LAADS website provided. (https://ladsweb.modaps.eosdis.nasa.gov/api/v2/content/archives/Document%20Archive/Science%20Data%20Product%20Documentation/VIIRS_Black_Marble_UG_v1.2_April_2021.pdf)

1.2 Data Disclaimer

1.2.1 Data Citation and Acknowledgment

Qian Xiao, Jun Wang, Cici Bauer, Meng Zhou, Yue Lyu, Jiachen Lu, Kehe Zhang (2023), Annual Summary of Artificial Light At Night from S-NPP/VIIRS at CONUS County and Census Tract, NASA Goddard Space Flight Center, Goddard Earth Sciences Data and Information Services Center (GES DISC), Accessed: [Data Access Date], 10.5067/TZY5MHFMYLKQ

1.2.2 Contact Information

PIs:

Qian Xiao, Ph.D Email: qian.xiao@uth.tmc.edu

Associate Professor, Department of Epidemiology, Human Genetics and Environmental Sciences, The University of Texas Health Science Center in Houston

1.3 What's New?

N/A

1.3.1 Version 1

This is the first version/version 1 of the product.

1.4 Quality Issues

The VNP46A4 data product provides yearly composites generated from daily cloud-free, and atmospheric-, terrain-, vegetation-, snow-, lunar-, and stray light-corrected radiance data (VNP46A2). we averaged observations from all angles upon snow free land surface throughout the year to generate the annual ALAN composite (500 m spatial resolution). It is worth noting that several factors, such as snow/cloud misclassification, severe aerosol pollution, and aurora borealis radiance, might potentially introduce positive bias to the daily VNP46A2 product. Urban centers might also show high NTL variation in daily scales. However, this impact should be minimal to the annual ALAN composite generated in this study, since temporal and spatial averaging are used to remove/smooth out the variation. Thus, it presents a mean status of the ALAN in yearly scale.

Imputation for census-tract level ALAN: ALAN data were missing for a small number of tracts (N=67): 65 tracts in the NYC area for the period of 2012-2014, one tract (St Mary, Louisiana) for year 2012, 2018, 2019, and 2020, and one tract (Monroe, Florida) for all years. For the 65 tracts in the NYC, we utilized the black marble VNP46A1 dataset to fill in these missing areas over the CONUS. The VNP46A1 dataset consists of daily at-sensor VIIRS DNB radiance data gridded to a resolution of 0.005 degrees and includes a cloud mask generated by the NASA VIIRS cloud mask algorithm. To ensure the accuracy of our results, we implemented a strict filtering criterion (QF_Cloud_Mask=0) to remove any cloud-contaminated pixels. For the other two tracks, because they had no residents or low population, and located in shallow oceans, we imputed them as 0. We created a flag variable indicating whether ALAN values were based on gap-filling methods, imputation or not. There is no missing value of radiance mean of county level ALAN.

2.0 Data Organization

The datasets contain annual summary of ALAN at both county and census tract level from year 2012 to year 2020 by projecting on administrative boundaries of counties and census tracts using the 2010 census information.

2.1 File Naming Convention

ALAN_VIIRS_CONUS_yyyy_to_yyyy_geo_level.xlsx

Where:
o yyyy = 4 digit year number [2012 - 2020].

o geo = can be one of county, census_tract.

Filename example: ALAN_VIIRS_CONUS_2012_to_2020_county_level.xlsx

2.2 File Format and Structure

County or census tract level data are separate files and both in xlsx text format with each row representing ALAN measures (see data dictionary in 3.2 for details) at a specific county/tract in a specific year.

2.3 Key Science Data Fields

The data has ALAN measurements including radiance mean, minimum, maximum, median, quantiles and standard deviation (see data dictionary in 3.2 for details).

3.0 Data Contents

3.1 Data Set Attributes (File Metadata)

N/A

3.2 Variable Data Attributes

Attribute	**Description**	**Type**	**Unit**
Geo_ID	Tract_ID: Census Tract FIPS; county_ID: County FIPS	String	Tract/county
county	County name	String	county
state	State name	String	state
year	Year, range from 2012 to 2020	Integer	year
nraster	Number of pixels within that county/tract	Integer	number

rad_mean	Radiance mean within that county/tract	Numeric	nanowatt per steradian per square centimeter (W·sr^-1·cm-^2)
rad_mean_imputed	Imputation value for radiance mean within that county/tract	Numeric	nanowatt per steradian per square centimeter (W·sr^-1·cm-^2)
Imputed	Yes: This radiance is imputed value; No: This radiance is original value	Boolean	unitless
rad_min	Maximum radiance within that county/tract	Numeric	nanowatt per steradian per square centimeter (W·sr^-1·cm-^2)
rad_max	Minimum radiance within that county/tract	Numeric	nanowatt per steradian per square centimeter (W·sr^-1·cm-^2)
rad_median	Radiance median within that county/tract	Numeric	nanowatt per steradian per square centimeter (W·sr^-1·cm-^2)
rad_q25	Lower quartile radiance within that county/tract	Numeric	nanowatt per steradian per square centimeter (W·sr^-1·cm-^2)
rad_q75	Upper quartile radiance within that county/tract	Numeric	nanowatt per steradian per square centimeter (W·sr^-1·cm-^2)
rad_sd	standard deviation	Numeric	nanowatt per steradian per square centimeter (W·sr^-1·cm-^2)
rad_IQR	Interquartile range equals rad_q75 minus rad_q25	Numeric	nanowatt per steradian per square centimeter (W·sr^-1·cm-^2)

3.3 Geolocation Fields

These fields appear for every data observation

Attribute	**Description**	**Type**
year	From 2012 to 2020	integer
County_id	County FIPS	Five-digit String
Tract_id	Census Tract FIPS	Five-digit String

3.4 Dimensions

Attribute	Description	Dimensions
year	From 2012 to 2020	9
County_id	County FIPS	3108
Tract_id	Census Tract FIPS	72538

4.0 Products/Parameters

We prepared two datasets, “ALAN_VIIRS_CONUS_2012_to_2020_county_level.xlsx” and “ALAN_VIIRS_CONUS_2012_to_2020_census_tract_level.xlsx”, for county- and tract-level data respectively. Data dictionary in 3.2 listed all variables and descriptions. Both datasets were in long format, with each row representing ALAN summary at a specific county/tract in a specific year.

4.1 Data Fields

N/A

4.2 Fill Values

See section 1.4.

4.3 Quality Control

See section 1.4.

5.0 Options for Reading the Data

The data files are XLSX text files. The Microsoft Excel and its alternative software are the best to read the data. Many popular editor tools can also read the data.

6.0 GES DISC Data Services

If you need assistance or wish to report a problem:
Email: gsfc-dl-help-disc@mail.nasa.gov
Voice: 301-614-5224
Fax: 301-614-5268
Address:
Goddard Earth Sciences Data and Information Services Center NASA Goddard Space Flight Center Code 619, Greenbelt, MD 20771 USA

6.1 How To Articles

The GESDISC web site contains many informative articles under the “How To Section”, “FAQ” (frequently asked questions), “News”, “Glossary”, and “Help”. A sample of these articles includes:

Earthdata Login for Data Access

How to Download Data Files from HTTPS Service with wget

How to Obtain Data in NetCDF Format via OpeNDAP

Quick View Data with Panoply

How to Read Data in NetCDF Format with R

How to Read Data in HDF-5 or netCDF Format with GrADS

How to read and plot NetCDF MERRA-2 data in Python

How to Subset Level-2 Data

How to use the Level 3 and 4 Subsetter and Regridder

7.0 More Information

An R shiny dashboard for visualizing ALAN data is on (currently only county-level data are

available): https://spatiotemporal-data-science.shinyapps.io/ALAN/

The following R code snippet shows how to read the variables of lat, lon, and radiance from NetCDF4 file and convert to raster file. The raster file can be used for extracting radiance at county- or census-tract levels and mapping.

```
# netCDF file to process
year <- 2012 # example year
ncfile <- paste0("VIIRS_Black_Marble_VNP46A4.",year,".nc4")
# directory folder
filepath <- paste0("data/BLACK MARBLE 2012_TO_2020/processed/",year,"/")
# description file of netCDF
sinkfile <- paste0(filepath,year,"_US_nc.txt")
# raster file processed from netCDF
rasterfile <- paste0(filepath,year,"_US_LAN.tif")


################################
# Read in NetCDF file ####
################################
nc_data_us <- nc_open(filename = ncfile)
# print the attributes
print(nc_data_us)
# variable names
names(nc_data_us$var)

# save the data description to a txt file
{
   sink(sinkfile)
   print(nc_data_us)
   sink()
}

# Read data from a netCDF file
lan.radiance.us <- ncvar_get(nc_data_us, "AllAngle_Composite_Snow_Free") # radiance t all
angle snow-free
lan.lat.us <- ncvar_get(nc_data_us, "latitude", verbose = F)
lan.lon.us <- ncvar_get(nc_data_us, "longitude")

# close the netCDF file
nc_close(nc_data_us)

################################
# Create Raster file #####
```

```
##############################

# rasterize
# trick: need to transpose the data, but do not flip
nc_raster_us <- raster(t(lan.radiance.us)
                       , xmn = min(lan.lon.us), xmx = max(lan.lon.us)
                       , ymn = min(lan.lat.us), ymx = max(lan.lat.us)
                       crs = CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs+
                towgs84=0,0,0"))

# load packages
library(sp);library(stats);library(reshape2);library(lattice);library(latticeExtra);
library(raster) # package for raster manipulation
library(rgdal) # package for geospatial analysis
library(rasterVis);library(dplyr);library(stringr);library(broom);library(reshape);library(geosphere)

# choose which year of data to process

year <- "2012"
# read in raster file
nc_raster <- raster::raster(nc_raster_us)

####################################
# Extract county level raster    #####
####################################

# read in county level shapefile
county_spdf <- readOGR(
   dsn = " tl_2020_us_county"
   , layer = "tl_2020_us_county"
   , verbose = FALSE
)

## start to extract raster by county ----------

# save a temp file in case of interruption or stuck
tempfilename <- paste0('raster_extr_county_temp_',year,'.rds')

# project coordination system of raster on shapefile
raster_crs <- proj4string(nc_raster)
county_spdf <- spTransform(county_spdf, raster_crs)
### identify order of the shapefile data
county_spdf@data$ID <- 1:nrow(county_spdf)
extr_ID <- county_spdf@data$GEOID
```

```
Nrun <- length(extr_ID)

raster_extr <- data.frame()

# if stuck or interrupted, restart from this step;
# start_j is the position of GEOID to restart from
if (file.exists(tempfilename)){
   print("Found temp file - resuming from geoid")
   raster_extr <- readRDS(tempfilename) # read in temporary file
   startindex <- raster_extr[nrow(raster_extr),] # show the last observation
   print(startindex) # print out
   (start_j <- startindex$j) # set j(j-th GEOID) to start from the last observed j
   raster_extr <- raster_extr[raster_extr$j<start_j,] # remove last observed j
   print(raster_extr[nrow(raster_extr),]) #print the last observation to see if deleted correctly
}

for (j in 1:Nrun){
# if stuck or interrupted, restart from j=start_j(below) instead from j=1(above).
#for (j in start_j:Nrun){
   GEOID <- extr_ID[j]
   print(paste0("Start process No.",j," ",GEOID))
   extr <- raster::extract(nc_raster, county_spdf[county_spdf$GEOID == GEOID,]
                                      , method = 'simple' # extract single value
                                      , df = TRUE # output dataframe
   )
   if(nrow(extr)!=0){
      extr$j <- j
      extr$ID <- GEOID
      colnames(extr) <- c("GEOID", "radiance","j")
      raster_extr <- rbind(raster_extr, extr)
   } else raster_extr <- raster_extr

   saveRDS(raster_extr, tempfilename)      # save to temporary file
   gc()
}

if(nrow(raster_extr)!=0){
   # get summary of extracted raster
   rad_summary <- raster_extr %>% group_by(GEOID) %>%
      mutate(log.rad = log(radiance))%>%
      summarise(nraster=n()
                , rad_mean = mean(radiance, na.rm = TRUE)
                , rad_min = min(radiance, na.rm = TRUE)
                , rad_max = max(radiance, na.rm = TRUE)
```

```
                    , rad_median = median(radiance, na.rm = TRUE)
                    , rad_sd = sd(radiance, na.rm=TRUE)
                    , rad_q25 = as.vector(quantile(radiance, prob = 0.25, na.rm = TRUE))
                    , rad_q75 = as.vector(quantile(radiance, prob = 0.75, na.rm = TRUE)))%>%
      mutate(rad_IQR = rad_q75-rad_q25)
}
length(unique(rad_summary$GEOID)) # 3233
length(unique(extr_ID)) # 3234

rad_summary$year <- year

saveRDS(rad_summary, paste0(getwd(),"/rad_summary_county_US_",year,"_2020shp.rds"))

########################################
# Extract census tract level raster #####
########################################
nc_raster <- raster::raster(nc_raster_us) # the same raster file for county-level extraction
raster_crs <- proj4string(nc_raster)

statefips <- c(paste0("0",c(1:2,4:6,8:9)), c(10:13,15:42,44:51,53:56,60,66,69,72,78))
filenames <- paste0("tl_2020_", statefips, "_tract")

## start to extract tract-level data ----------

raster_extr_us <- data.frame()

tempfilename <- paste0("raster_extr_tract_temp_",year,".rds")

if (file.exists(tempfilename)){
   print("Found temp file")
   raster_extr_us <- readRDS(tempfilename)
   print("the last row of extracted raster")
   startindex <- raster_extr_us[nrow(raster_extr_us),] # identify the last statefips was processing
   print(startindex)
   raster_extr_us <- raster_extr_us %>% filter(stateFIPS!=startindex$stateFIPS) # remove the last
processing state
   starti <- which(statefips == raster_extr_us[nrow(raster_extr_us),]$stateFIPS) # the last state
completed extraction
   print("restart from")
   print(filenames[starti+1])
}


## extract by state
```

```
for (i in 1:length(filenames)
        ##### run this follow if code ends abruptly
        #for (i in (starti+1):length(filenames)
){
   # read in shape file
   # (load polygon shapes as a reference to extract raster)
   census_spdf <- readOGR(
      #dsn = paste0(getwd(),"/Shapefiles/",filenames[i])
      dsn = paste0(getwd(),"/Shapefiles/2020/",filenames[i])
      , layer = filenames[i]
      , verbose = FALSE
   )

   # extract raster by census tract
   ### transform crs of the shapefile to be the same as raster
   census_spdf <- spTransform(census_spdf, raster_crs)

   extr_GEOID <- census_spdf$GEOID
   Nrun <- length(extr_GEOID)

   raster_extr <- data.frame()

   ## extract by census tract within state
   for (j in 1:Nrun){

      GEOID <- extr_GEOID[j]
      print(paste0("Start processing for state ",statefips[i]," ", j, "-th GEOID ",GEOID))

      extr <- raster::extract(nc_raster, census_spdf[census_spdf$GEOID == GEOID,]
                                          , method = 'simple' # extract single value
                                          , df = TRUE # output dataframe
      )
      if(nrow(extr)!=0){
         extr$stateFIPS <- statefips[i]
         extr$ID <- GEOID
         colnames(extr) <- c("GEOID", "radiance", "stateFIPS")
         raster_extr <- rbind(raster_extr, extr)
      }
      else raster_extr <- raster_extr
      gc()
   }

   raster_extr_us <- rbind(raster_extr_us, raster_extr)
   saveRDS(raster_extr_us, tempfilename) # save to temp file
```

```
}

if(nrow(raster_extr_us)!=0){
   # get summary of extracted raster
   rad_summary <- raster_extr_us %>% group_by(GEOID) %>%
      mutate(log.rad = log(radiance))%>%
      summarise(nraster=n()
                     , rad_mean = mean(radiance, na.rm = TRUE)
                     , rad_min = min(radiance, na.rm = TRUE)
                     , rad_max = max(radiance, na.rm = TRUE)
                     , rad_median = median(radiance, na.rm = TRUE)
                     , rad_sd = sd(radiance, na.rm=TRUE)
                     , rad_q25 = as.vector(quantile(radiance, prob = 0.25, na.rm = TRUE))
                     , rad_q75 = as.vector(quantile(radiance, prob = 0.75, na.rm = TRUE)))%>%
      mutate(rad_IQR = rad_q75-rad_q25)
}

length(unique(rad_summary$GEOID)) # 83776 (using 2020 shp)

rad_summary$year <- year

saveRDS(rad_summary, paste0(getwd(),"/data/BLACK MARBLE
2012_TO_2020/processed/",year,"/rad_summary_tract_US_",year,"_2020shp.rds"))
```

8.0 Acknowledgments

9.0 References