Supplementary Material S1: Input data, software requirements, guide, and codes

Article: A Near Real-Time and Free Tool for the Preliminary Mapping of Active Lava Flows during Volcanic Crises: The Case of Hotspot Subaerial Eruptions

Francisco Javier Vasconez [1,*], Juan Camilo Anzieta [2,3], Anais Vásconez Müller [1], Benjamin Bernard [1] and Patricio Ramón [1]

[1] Instituto Geofísico, Escuela Politécnica Nacional, Ladrón de Guevara E11-253, Quito 170525, Ecuador; fjvasconez@igepn.edu.ec (FJV); avasconez@ igepn.edu.ec (AVM); bbernard@igepn.edu.ec (BB); pramon@igepn.edu.ec (PR)
[2] Department of Earth Sciences, Simon Fraser University, Surrey, BC V5A 1S6, Canada; janzieta@sfu.ca (JCA)
[3] Escuela de Ciencias Físicas y Matemática, Pontificia Universidad Católica del Ecuador, Quito 170525, Ecuador
* Correspondence: fjvasconez@igepn.edu.ec

A. Procedure to download the FIRMS data

1. For downloading the FIRMS products, click on https://firms.modaps.eosdis.nasa.gov/download/ or https://firms2.modaps.eosdis.nasa.gov/download/. NASA provides two servers for downloading data because sometimes one of these servers is overloaded and generates errors.
2. Click on *Create a New Request*. Choose *Custom Region* (Figure S1a). Click on *Use Map*, select your area of interest (AOI) by using the draw polygon tool (Figure S1b) and save. Another option is to type the geographic coordinates of your AOI into the box (Figure S1a).
3. Check the VIIRS S-NPP and VIIRS NOAA-20 boxes. Define the dates of interest, e.g., between the onset and the processing day for ongoing eruptions, or from the beginning to the end for past eruptions (Figure S1a). Choose comma-separated text (.csv) and include your email address (Figure S1a). Finally, check the *send email confirmation for this data request* box and submit (Figure S1a).
4. After few seconds, you will receive a confirmation email (Figure S1c), where you can review the selected AOI (*View map* link) and can confirm the chosen dates. Afterwards, you will receive another two emails with the download links per satellite (i.e., SV for S-NPP and J1V for NOAA-20).
5. Click on the links and unzip the folders. Each folder contains two files, Readme.txt and thermal data.csv. When the thermal data is very extensive, more than one folder per satellite is sent. In such uncommon cases you must create a new dataset with the combination of the two files (i.e., one per satellite) under the same parameters (i.e., 14 attributes). The thermal dataset contains the following parameters: ***latitude, longitude,*** *bright_ti4 (or brightness), scan,* ***track, acq_date****,* ***acq_time****, satellite, instrument, confidence, version, bright_ti5 (or bright_t31),* ***frp*** *and type (or daynight)* (Figure S1d). Only few attribute names change from one satellite to the other. Of these 14 columns, our codes use latitude, longitude, track, acq_date, acq_time and frp. Importantly, our codes were made to automatically identify the .csv FIRMS' datasets by reading the patterns "SV" and "J1V" in the files' name. If the user wants to change the file name, we strongly recommend keeping SV and J1V as part of the names (e.g., Kilauea_2018_SV.csv). Other R-users can alter the reading part of the codes as needed.

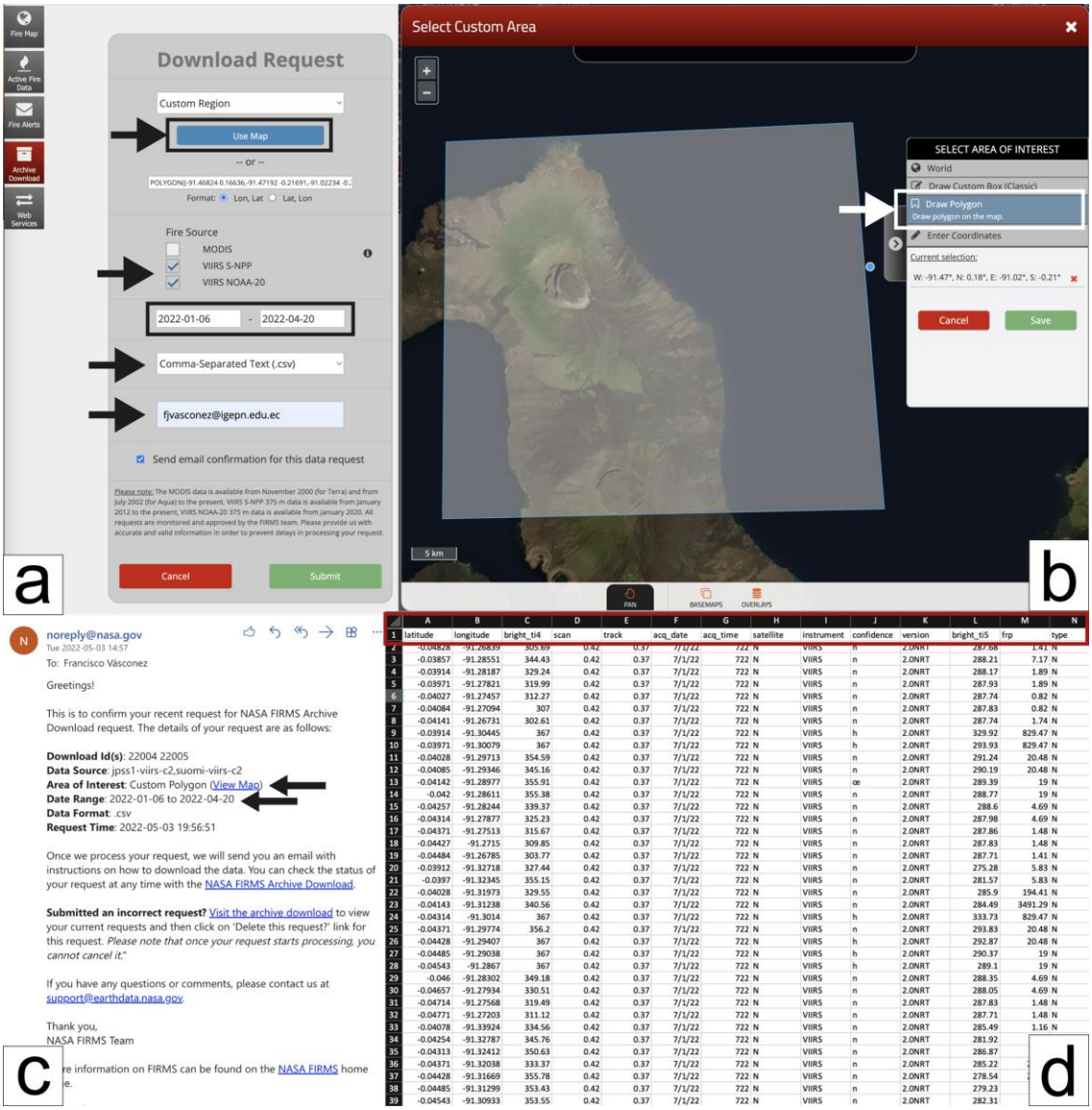

Figure S1. (a) FIRMS' download request interface, (b) "Use Map" tab for drawing the area of interest, (c) Email confirmation for checking the AOI and dates, (d) FIRMS' dataset which contains 14 parameters.

B. Required software: R and RStudio (optional)

1. Click on https://cran.r-project.org/bin/windows/base/ for windows or https://cran.r-project.org/bin/macosx/ for macOS, download and install. For installing just follow each step explained during the installation process.
2. (Optional / recommended) Click on https://www.rstudio.com/products/rstudio/download/, choose *free* and download the RStudio Desktop according to your computer system and install. Strongly recommended.

After installing the programs be sure that they are working correctly (for instance running the FRP_statistical script), if it does not work and an error occurs, be sure that all the requirements are also installed, for instance XQuartz for macOS (https://www.xquartz.org/). All the needed requirements are listed in the links previously provided.

C. Quick guide for the use of the tool (for further details see the accompanied manuscript) and the two R-scripts (FRP_statistical and LavaFlow_mapper)

1) Create a dedicated folder or working directory which will contain the FIRMS datasets (in .csv format) and the two R-codes (i.e., FRP_statistical.R and LavaFlow_mapper.R). Importantly, for past eruptions earlier than 2020 there is no NOAA-20 data, so we strongly recommend duplicating part of the S-NPP data in a new file called "J1V" to avoid errors. The duplicated data (3 rows only) should surpass the track and FRP default filters (i.e., track<=0.5 and frp>=35). Another option is to not run/comment all the lines that use the NOOA-20 dataset (taken into account that some parts of the code use both datasets so they will still need to be modified to use the S-NPP data only).

2) In R the user must install the following packages "tictoc", “leaflet”, “leaflegend”, “sp”, “viridisLite”, “viridis”, “mapview”, “geosphere” and “webshot”. For simplicity, we strongly suggest using RStudio, where, after opening our scripts, the software will automatically identify the need for these packages and will ask the user to install them. By clicking on install packages the installing process in the “Jobs” tab will start.
3) Run LavaFlow_mapper: for the onset and first days of ongoing eruptions, we strongly recommend using the default filters, track<=0.5 and FRP-filter >=35 (lines: 26 & 37). Additionally, the user can include a reference point in decimal WGS84 coordinates to estimate the maximum linear extension of the lava flows per satellite and through time (lines: 71-72). To run the code in RStudio is, click on the Source button; or press command (ctrl in Windows) + shift + enter. You can also run the code line by line (crtl + enter or Run button) for more flexibility of outputs (the total time registered by tictoc for the code will increase if run by hand).
4) In R, an interactive thermal map should be opened in your browser and a plot will appear for the estimated maximum extension. In RStudio the interactive thermal map will appear in the “viewer tab” and the estimated maximum distance of the lava flows in the “Plot tab”. The interactive map can be exported as .png and the plot as .png and .pdf files, as needed. Additionally, four tables with the resulting filtered data and the estimated length of the lavas will be automatically saved in the working directory (or modify the code for other directories).
5) After a week or when the FIRMS datasets are large enough to produce a robust cumulative statistical analysis (i.e., the cumulative statistical parameters tend to stabilize); the user should run the FRP_statistical code to choose a new FRP-threshold which will fit the conditions of the eruption being investigated. The resulting cumulative statistical analysis of data collected every 12-hours will be saved in the working directory as separate tables (one per satellite) and the resulting composite figure could be exported as .png or .pdf files as needed.
6) After having chosen a proper filter for your case study, the default FRP-filter of 35 should be replaced in the LavaFlow_mapper (lines: 26 & 37 in the file) and the code should be run again to produce a more reliable temporal-thermal map.

C1. FRP_statistical code: also available in https://vhub.org/tools/lavaflowmapper/wiki **with the nine case studies**

```
#####"FRP_statistical" version 1.0######
###Tested in R 4.1.3 and RStudio 2022.02.01######
####################CALL LIBRARIES############ Be sure to install and call all the dependent packages before
running
library(tictoc)

tic() #start timer
#################CLEAN VARIABLES AND TABS#################
rm(list=ls()) #clean global environment
graphics.off() #clean previous plots

#############SET WORKING DIRECTORY AND READ FIRMS DATASETS#################
setwd(dirname(rstudioapi::getActiveDocumentContext()$path)) #set working directory automatically / this can
be changed by R-users if needed
SNPP<-read.csv(list.files(pattern = "SV(.*)csv$"),sep=",",dec=".",header=TRUE) #read S-NPP data
raw_times1=sprintf("%04d", unlist(SNPP["acq_time"]))#extract times and pad 0 if needed
raw_dates1=sprintf(unlist(SNPP["acq_date"]))#extract dates
datetime_raw1=Map(paste,raw_dates1,raw_times1)#merge dates and times
datetime1=strptime(datetime_raw1,format = "%d/%m/%y %H%M")#dates and times as POSIX objects
#datetime1=strptime(datetime_raw1,format = "%Y-%m-%d %H%M")#dates and times as POSIX objects uncomment this
line if previous one does not work
FRP1=as.numeric(unlist(SNPP["frp"]))#extract FRP

###IMPORTANT!!! For past eruptions earlier than 2020 there is no NOAA-20 data, we strongly recommend to du-
plicate part of the SNPP data (3 rows) to avoid errors, another option is to comment all the lines that use
the NOOA dataset####
NOAA<-read.csv(list.files(pattern = "J1V(.*)csv$"),sep=",",dec=".",header=TRUE) #read NOAA-20 data
raw_times2=sprintf("%04d", unlist(NOAA["acq_time"]))#extract times and pad 0 if needed
raw_dates2=sprintf(unlist(NOAA["acq_date"]))#extract dates
datetime_raw2=Map(paste,raw_dates2,raw_times2)#merge dates and times
datetime2=strptime(datetime_raw2,format = "%d/%m/%y %H%M")#dates and times as POSIX objects
#datetime2=strptime(datetime_raw2,format = "%Y-%m-%d %H%M")#dates and times as POSIX objects uncomment this
line if previous one does not work
FRP2=as.numeric(unlist(NOAA["frp"]))#extract FRP

par(mfrow=c(2,2))
par(mar=c(3,5,3,3))
####################PLOT HISTOGRAMS#########################
```

```
stat_names=c('mean','minimum','5th percentile','1st quartile','median','3rd quartile','95th percentile','max-
imum') #names of the statistical values
stat_vals1=round(c(mean(FRP1),quantile(FRP1,c(0,0.05,0.25,0.5,0.75,0.95,1))),1) #define quantiles of interest
and mean
stat_legend=Map(paste,stat_names,stat_vals1,sep=': ') #legend formatting for histogram
hist1=hist(FRP1,xlim = c(0,stat_vals1[7]),col="orange",breaks =
seq(from=min(FRP1),to=max(FRP1)+20,by=10),main=sprintf("Histogram of S-NPP
FRP\n%s",paste(min(datetime1),max(datetime1),sep = ' to '))) #create a histogram with breaks of size 20
abline(v=mean(FRP1),lty=2, lwd=1.5) #add vertical black dashed line for the mean
abline(v=quantile(FRP1,0.75),lty=3, lwd=1, col="blue") #add vertical blue dotted line for the 3rd quartile
legend('topright',legend=stat_legend,cex=1,ncol = 1,bty="n") #legend for the histogram single column, if two
columns ncol = 2

stat_vals2=round(c(mean(FRP2),quantile(FRP2,c(0,0.05,0.25,0.5,0.75,0.95,1))),1)#define quantiles of interest
and mean
stat_legend=Map(paste,stat_names,stat_vals2,sep=': ') #legend formatting for histogram
hist2=hist(FRP2,xlim = c(0,stat_vals2[7]),col="purple3", breaks =
seq(from=min(FRP2),to=max(FRP2)+20,by=10),main=sprintf("Histogram of NOAA-20
FRP\n%s",paste(min(datetime2),max(datetime2),sep = ' to ')))#create a histogram with breaks of size 20
abline(v=mean(FRP2),lty=2, lwd=1.5) #add vertical black dashed line for the mean
abline(v=quantile(FRP2,0.75),lty=3, lwd=1, col="blue") #add vertical blue dotted line for the 3rd quartile
legend('topright',legend=stat_legend,cex=1,ncol = 1,bty="n") #legend for the histogram single column, if two
columns ncol = 2

############CUMULATIVE STATISTICAL FRP PARAMETERS EVERY 12 HOURS TO S-NPP#########
stat_namesb=c('mean','1st quartile','median','3rd quartile')#names of the statistical values
init_time1=strptime(paste(as.Date(min(datetime1)),"00:00:00"),format = "%Y-%m-%d %H:%M:%S")#extract first
date of period
end_time1=strptime(paste(as.Date(max(datetime1)),"24:00:00"),format = "%Y-%m-%d %H:%M:%S")#extract last date
of period
times_stats_12a=seq(from=init_time1,to=end_time1,by=43200)#create half-day datetime sequence
halfday_cumu_stats_FRPS1=setNames(data.frame(matrix(nrow = length(times_stats_12a)-1,ncol =
5)),c('timesdates12h',stat_namesb))#initialize statistics dataframe every 12 hours
for (i in 1:length(times_stats_12a)-1){#calculate and assign cumulative 12 hours statistics
  statistics_12a=round(c(mean(FRP1[datetime1<times_stats_12a[i+1]]),quan-
tile(FRP1[datetime1<times_stats_12a[i+1]],c(0.25,0.5,0.75))),1)
  halfday_cumu_stats_FRPS1[i,1]=as.character(times_stats_12a[i+1])
  halfday_cumu_stats_FRPS1[i,2]=statistics_12a[1]
  halfday_cumu_stats_FRPS1[i,3]=statistics_12a[2]
  halfday_cumu_stats_FRPS1[i,4]=statistics_12a[3]
  halfday_cumu_stats_FRPS1[i,5]=statistics_12a[4]
}
plot(times_stats_12a[2:length(times_stats_12a)],halfday_cumu_stats_FRPS1[,2],type = 'l',lwd=1.5,ylim =
c(0,max(halfday_cumu_stats_FRPS1$mean,na.rm = TRUE)),xlab='',xaxt="n",ylab='cumul. mean
FRP',main=sprintf("Cumulative statistic of S-NPP FRP every 12
hours\n%s",paste(min(datetime1),max(datetime1),sep = ' to ')))
axis.POSIXct(1, at=seq(times_stats_12a[2], max(times_stats_12a), by="week"), format="%b %d") #x-axis ticks
per week. User can changed the x-axis to day, month or year as needed
for(i in 3:5){
  lines(times_stats_12a[2:length(times_stats_12a)],halfday_cumu_stats_FRPS1[,i],col=i-1)
}
axis(4, las=3)
grid(nx = NA, ny=NULL, col = "lightgray", lty = "dotted",lwd = par("lwd")) #add grid
legend('topright',legend = c(stat_namesb),col=c(1:4),cex=1,lty=rep(1),lwd=1.5,bty="o", bg = "white") #plot
legend

###############CUMULATIVE STATISTICAL FRP PARAMETERS EVERY 12 HOURS TO NOAA-20#########
init_time2=strptime(paste(as.Date(min(datetime2)),"00:00:00"),format = "%Y-%m-%d %H:%M:%S")#extract first
date of period
end_time2=strptime(paste(as.Date(max(datetime2)),"24:00:00"),format = "%Y-%m-%d %H:%M:%S")#extract last date
of period
times_stats_12b=seq(from=init_time2,to=end_time2,by=43200)#create half-day datetime sequence
halfday_cumu_stats_FRPS2=setNames(data.frame(matrix(nrow = length(times_stats_12b)-1,ncol =
5)),c('timesdates12h',stat_namesb))#initialize statistics data frame every 12 hours
for (i in 1:length(times_stats_12b)-1){#calculate and assign cumulative 12 hours statistics
  statistics_12b=round(c(mean(FRP2[datetime2<times_stats_12b[i+1]]),quan-
tile(FRP2[datetime2<times_stats_12b[i+1]],c(0.25,0.5,0.75))),1)
  halfday_cumu_stats_FRPS2[i,1]=as.character(times_stats_12b[i+1])
  halfday_cumu_stats_FRPS2[i,2]=statistics_12b[1]
  halfday_cumu_stats_FRPS2[i,3]=statistics_12b[2]
  halfday_cumu_stats_FRPS2[i,4]=statistics_12b[3]
  halfday_cumu_stats_FRPS2[i,5]=statistics_12b[4]
}
plot(times_stats_12b[2:length(times_stats_12b)],halfday_cumu_stats_FRPS2[,2],type = 'l',lwd=1.5,ylim =
c(0,max(halfday_cumu_stats_FRPS2$mean,na.rm = TRUE)),xlab='',xaxt="n",ylab='cumul. mean
FRP',main=sprintf("Cumulative statistic of NOAA-20 FRP every 12
hours\n%s",paste(min(datetime2),max(datetime2),sep = ' to ')))
axis.POSIXct(1, at=seq(times_stats_12b[2], max(times_stats_12b), by="week"), format="%b %d") #x-axis ticks
per week. You can change the x-axis to day, month or year as needed
```

```
for(i in 3:5){
  lines(times_stats_12b[2:length(times_stats_12b)],halfday_cumu_stats_FRPS2[,i],col=i-1)
}
axis(4, las=3)
grid(nx = NA, ny=NULL,col = "lightgray", lty = "dotted",lwd = par("lwd")) #add grid
legend('topright',legend = c(stat_namesb),col=c(1:4),cex=1,lty=rep(1),lwd=1.2,bty="o", bg = "white") #plot
legend

################SAVE RESULTS#########
write.csv(halfday_cumu_stats_FRPS1,'halfdayFRPs_SNPP.csv',row.names = FALSE,quote = FALSE) #save the statis-
tical results of the N-SPP data
write.csv(halfday_cumu_stats_FRPS2,'halfdayFRPs_NOAA.csv',row.names = FALSE,quote = FALSE) #save the statis-
tical results of the NOAA-20 data

toc() #end timer
```

C2. LavaFlow_mapper code: also available in https://vhub.org/tools/lavaflowmapper/wiki with the nine case studies

```
######"LavaFlow_mapper" version 1.0######
###Tested in R 4.1.3 and RStudio 2022.02.01######
################CALL PACKAGES################## Be sure to install and call all the dependent packages be-
fore running
library(tictoc)
library(leaflet)
library(leaflegend)
library(sp)
library(viridisLite)
library(viridis)
library(mapview)
library(geosphere)
#library(rgdal)  #uncomment only if you want to plot a shapefile (WGS84) in the interactive thermal map.
**For comparison purposes**

tic() #start timer
##################CLEAN VARIABLES AND TABS###################
rm(list=ls()) #clean global environment
graphics.off() #clean previous plots

####################SET WORKING DIRECTORY AND READ FIRMS DATASETS#######################
setwd(dirname(rstudioapi::getActiveDocumentContext()$path)) #set working directory automatically / this
can be changed by R-users if needed
SNPP<-read.csv(list.files(pattern = "SV(.*)csv$"),sep=",",dec=".",header=TRUE) #read Suomi data
NOAA20<-read.csv(list.files(pattern = "J1V(.*)csv$"),sep=",",dec=".",header=TRUE) #read NOAA data
#LF=readOGR(dsn=".",layer="Wolf_LF_22") #uncomment only if you have a shapefile with the area covered by
the lavas; the name (here:Wolf_LF_22) must be replaced by your shapefile name in WGS84

####################FILTER AND EXTRACT THE S-NPP DATA#########################
Suomi<-subset(SNPP,track <= 0.5  & frp>=35) ## FILTERS!!! change the FRP filter according to the FRP_sta-
tistical results of your case study or use the default value of 35+-17
raw_times1=sprintf("%04d", unlist(Suomi["acq_time"]))#extract times and pad 0 if needed
raw_dates1=sprintf(unlist(Suomi["acq_date"]))#extract dates
datetime_raw1=Map(paste,raw_dates1,raw_times1)#merge dates and times
datetime1=strptime(datetime_raw1,format = "%d/%m/%y %H%M")#dates and times as POSIX objects
#datetime1=strptime(datetime_raw1,format = "%Y-%m-%d %H%M")#dates and times as POSIX objects uncomment
this line if previous one does not work
FRP1=as.numeric(unlist(Suomi["frp"]))#extract FRP
LATS1=as.numeric(unlist(Suomi["latitude"]))#extract latitudes
LONGS1=as.numeric(unlist(Suomi["longitude"]))#extract longitudes
DATE1=as.numeric(datetime1)/86400 #convert dates from seconds to days

##########READ AND EXTRACT THE NOAA-20 DATA############  IMPORTANT: For past eruptions earlier than 2020
there is no NOAA-20 data, so we strongly recommend to duplicate part of the SNPP data to avoid errors.
The duplicated data (3 rows) should surpass the track and FRP default filters.
NOAA<-subset(NOAA20,track <= 0.5 & frp>=35) ## FILTERS!!! change the FRP filter according to the FRP_sta-
tistical results of your case study or use the default value of 35+-17
raw_times2=sprintf("%04d", unlist(NOAA["acq_time"]))#extract times and pad 0 if needed
raw_dates2=sprintf(unlist(NOAA["acq_date"]))#extract dates
datetime_raw2=Map(paste,raw_dates2,raw_times2)#merge dates and times
datetime2=strptime(datetime_raw2,format = "%d/%m/%y %H%M")#dates and times as POSIX objects
#datetime2=strptime(datetime_raw2,format = "%Y-%m-%d %H%M")#dates and times as POSIX objects uncomment
this line if previous one does not work
FRP2=as.numeric(unlist(NOAA["frp"]))#extract FRP
LATS2=as.numeric(unlist(NOAA["latitude"]))#extract latitudes
LONGS2=as.numeric(unlist(NOAA["longitude"]))#extract longitudes
```

```
DATE2=as.numeric(datetime2)/86400 #convert dates from seconds to days

###############COLOR SCALE OF DATES##################
seq<-c(min(min(DATE1),min(DATE2)):(max(max(DATE1),max(DATE2))+1)) #define min and max dates
pal <- colorNumeric(palette = "Spectral", rev=TRUE,domain = seq) #use spectral palette for dates
myLabelFormat = function(...,dates=FALSE){
if(dates){
function(type = "numeric", cuts){ as.Date(cuts, origin="1970-1-1")}
} else {labelFormat(...) }} ## convert dates to numeric values to be used in the color-scale palette

############PLOT THE THERMAL ANOMALIES ON AN ESRI BASEMAP############## if the interactive map doesn't ap-
pear in the viewer tab, select lines 58-65 and click on run
m <- leaflet(Suomi) %>%
addProviderTiles("Esri.WorldImagery") %>%  #Add default basemaps (e.g. Esri.WorldImagery, OpenTopoMap,
OpenStreetMap.Mapnik, for others see http://leaflet-extras.github.io/leaflet-providers/preview/in-
dex.html)
addCircles(data=Suomi,lng=Suomi$longitude, lat=Suomi$latitude,weight = 3, radius=150, color =
~pal(DATE1), stroke = FALSE, fillOpacity = 0.7) %>% #fillOpacity changes transparency
addCircles(data=NOAA,lng=NOAA$longitude, lat=NOAA$latitude, weight = 3, radius=150, color = ~pal(DATE2),
stroke = FALSE, fillOpacity = 0.7) %>%
#addPolygons(data = LF, fillOpacity = 0.2, color = "white",weight = 4)%>% #uncomment only if you want to
compare with a map in shapefile format, LF=LavaFlow
addLegend("topright", pal = pal, values = ~seq, title = "Date",opacity=1,labFormat = myLabelFor-
mat(dates=TRUE))%>% #add color scale legend based on dates
addScaleBar() #add scale bar
m  #print the map

###################SAVE AND EXPORT RESULTS################### if needed uncomment the line 66
#mapshot(m, file = "map.png") #save the map in png format; before using it, you must install the webshot
package "webshot::install_phantomjs()". Otherwise you can export the map from the Viewer tab
write.csv(Suomi,"filter_VIIRS_S-NPP.csv") #save the filtered S-NPP data
write.csv(NOAA,"filter_VIIRS_NOAA-20.csv") #save the filtered NOAA-20 data

###############ESTIMATE LAVA FLOW DISTANCE IN KILOMETERS#########
reference_long=-91.32241 #IMPORTANT!!! longitude coordinates must be replaced by those of your case
study. If you do not have a reference point, keep the default, but the lenght estimations would be incor-
rect
reference_lat=-0.01753 # IMPORTANT!!! latitude coordinates must be replaced by those of your case study.
If you do not have a reference point, keep the default, but the lenght estimations would be incorrect
distances1={}#initialize distances vector for S-NPP
for (i in 1:length(LONGS1)){
distances1[i]=distm(c(reference_long, reference_lat), c(LONGS1[i], LATS1[i]))/1000#distance in kilometers
for S-NPP
}
distances2={}#initialize distances vector for NOAA-20
for (i in 1:length(LONGS2)){
distances2[i]=distm(c(reference_long, reference_lat), c(LONGS2[i], LATS2[i]))/1000#distance in kilometers
for NOAA-20
}

###############PLOT THE ESTIMATED LAVA FLOW DISTANCE#########
maxx<-max(c(max(distances1),max(distances2))) #maximum distance between S-NPP and NOAA to define the ver-
tical scale of the plot
plot(datetime1,distances1, type='h',lwd=2,col="orange",xlab = "",xaxt="n",ylim=c(0,maxx),ylab="Kilome-
ters",main=sprintf("Estimated lava flow dis-
tances\n%s",paste(min(c(min(datetime1),min(datetime2))),max(c(max(datetime1),max(datetime2))),sep = ' to
'))) #plot SNPP filtered data
par(new=T)
plot(datetime2,distances2, type='h',lwd=2,col="purple3",xlab = "",xaxt="n",ylab="",ylim=c(0,maxx)) #plot
NOAA filtered data
axis(4, las=3)
grid(nx = NA, ny=NULL, col = "lightgray", lty = "dotted",lwd = par("lwd")) #add grid
axis.POSIXct(1, at=seq(min(datetime1), max(datetime1), by="week"), format="%b %d") #IMPORTANT!!! here you
can set the x-axis by day, week, month or year, according to the duration of the eruption being investi-
gated. In this example it is by "week".
legend("topleft", legend=c("S-NPP", "NOAA-20"),col=c("orange", "purple3"), lty=rep(1),lwd=2, cex=0.7,
text.font=1, bty="o",box.lty = 1,bg = "white", ncol=2) #plot's legend
df1 <- data.frame(Date_SNPP=c(datetime1),Distance_SNPP=c(distances1)) #create a dataframe of the distance
per day based on S-NPP satellite
df2<-data.frame(Date_NOAA=c(datetime2), Distance_NOAA=c(distances2)) #create a dataframe of the distance
per day based on NOAA-20 satellite
write.csv(df1,"distance_SNPP.csv") #save the distance based on S-NPP data
write.csv(df2,"distance_NOAA.csv") #save the distance based on NOAA data

toc() #end timer
```