HyAirshed

A R package to approximate daily and seasonal airsheds of point coordinates using the HYSPLIT trajectory model and to estimate fire intensity within airsheds

We use the term "airshed" to refer to the approximate area containing aerosols that can affect the air quality of a point location (a single lat/lon) through meteorology. The approach uses the frequency of back trajectories over a grid with a specified resolution to approximate daily, seasonal, and composite airsheds. This manual contains a part of the methodology used in Liu et. al (in revision)[1] to approximate the airsheds of Delhi, Bengaluru, and Pune during two-month periods of high outdoor fire activity. We used back trajectories from the NOAA Hybrid Single Particle Lagrangian Integrated Trajectory (HYSPLIT) model[2] to approximate the daily and seasonal airsheds of the aforementioned cities and the MODIS fire radiative power (FRP) from the Collection 6 MOD14A1/MYD14A1 active fire products[3] as a proxy for fire intensity.

---

[1] Liu T, Marlier ME, DeFries RS, Westervelt DM, Xia KR, Fiore AM, Mickley LJ, and Cusworth DH (in revision) Contributions of outdoor biomass burning to air pollution in three Indian cities: Delhi, Bengaluru, and Pune.

[2] https://ready.arl.noaa.gov/HYSPLIT_traj.php

[3] http://modis-fire.umd.edu/pages/ActiveFire.php?target=DailyMOD14MYD14

Table of Contents

1. Operating the HYSPLIT model

Web-based computation of the HYSPLIT back trajectories ("Compute archive trajectories") is available through the Air Resources Laboratory (ARL) at http://ready.arl.noaa.gov/hypub-bin/trajtype.pl?runtype=archive.

1.1 Type of trajectories

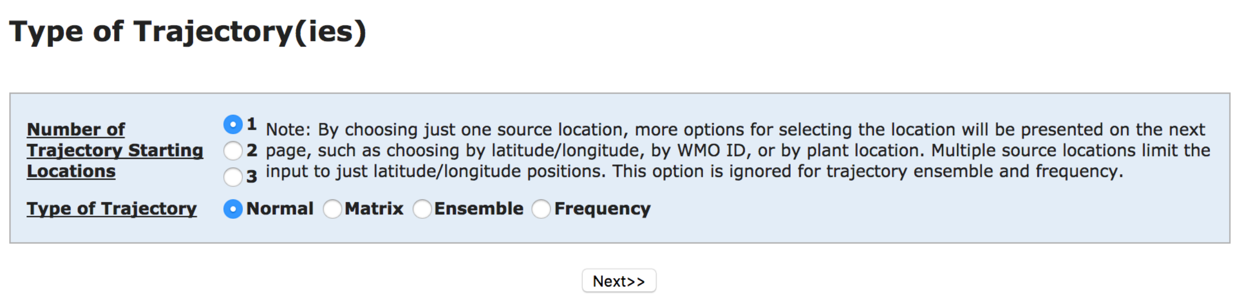

Figure 1. Type of trajectories.

Select **number of starting locations** (1-3) and use the default **type of trajectory**: "Normal" (Figure 1). Other types of trajectories require further editing of R functions.

1.2 Meteorology and starting locations

Select **meteorology** and note the spatial and temporal limitations of each meteorology dataset (Figure 2). In Liu et al. (in revision), we used the GDAS, 0.5 degree, global dataset, available 09/2007 to present. Then, enter **starting locations** by using 1. an interactive map; 2. coordinates (decimal degrees); 3. coordinates (degrees, minutes, seconds); 4. city (from a list); 5. airport or World Meteorological Organization (WMO) ID. Note that for more than 1 starting location, only option 2 (coordinates in decimal degrees) can be used as input (Figure 3).

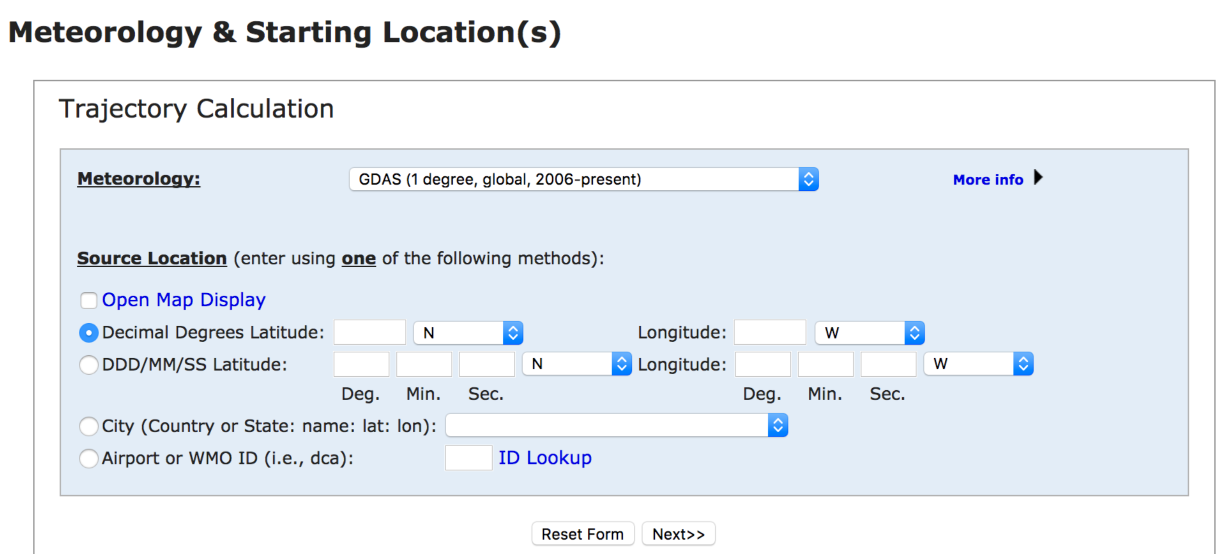

Figure 2. Meteorology and starting locations (for 1 starting location).

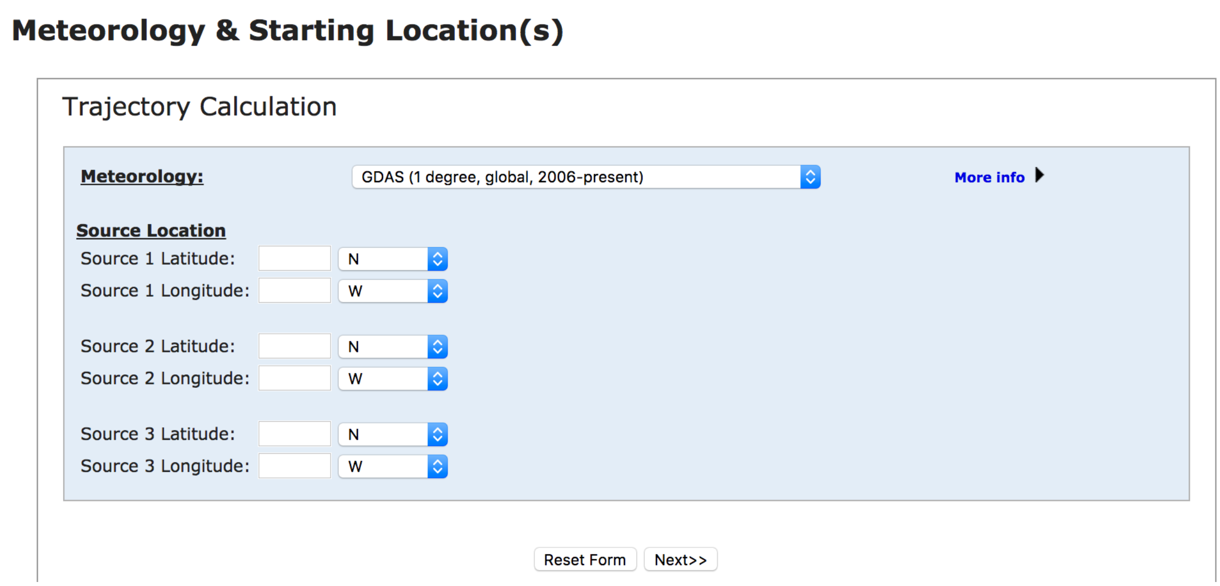

Figure 3. Meteorology and starting locations (for 2-3 starting locations).

1.3 Meteorology file

Select **meteorology file**, while is named according to the date of file. Meteorology files for certain days may be missing. If so, back trajectories cannot be run for those days. Record the days that are available and successfully run in the table **hysplitDays.csv**[4] by month and year. Use **c()** to record days that are not connected. For example, if the 5th-10th days and 20th day are available in a month, input: c(5:10, 20).

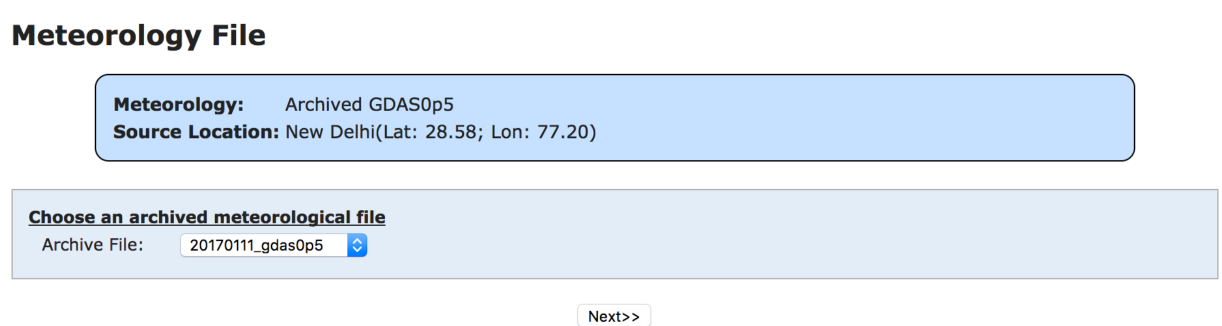

Figure 4. Meteorology file. For this example, we selected the January 11, 2017 GDAS 0.5 degree meteorology file and New Delhi as the starting location.

1.4 Model run parameters

Many parameters need to be set for the model run: 1. **trajectory direction** (backward, or back trajectory); 2. **start time**, in UTC (select hour to start back trajectory and convert local time to UTC); 3. **total run time**, in hours (72 hours is optimal for the R function, but a run time < 72 hours can be used); and 4. **level height**, in meters (starting altitude of model run at starting location; 1-3 heights can be selected) (Figure 5). If not mentioned, the parameters have either been populated or

[4] A template of **hysplitDays.csv** is located in the "template" folder of the **HyAirshed** package on GitHub. Contents of the **HyAirshed** package can be accessed at: https://github.com/tianjialiu/HyAirshed-RPackage.

the default should be used. In addition, select **GIS shapefile** in display options (Figure 6). Then, run the trajectory!

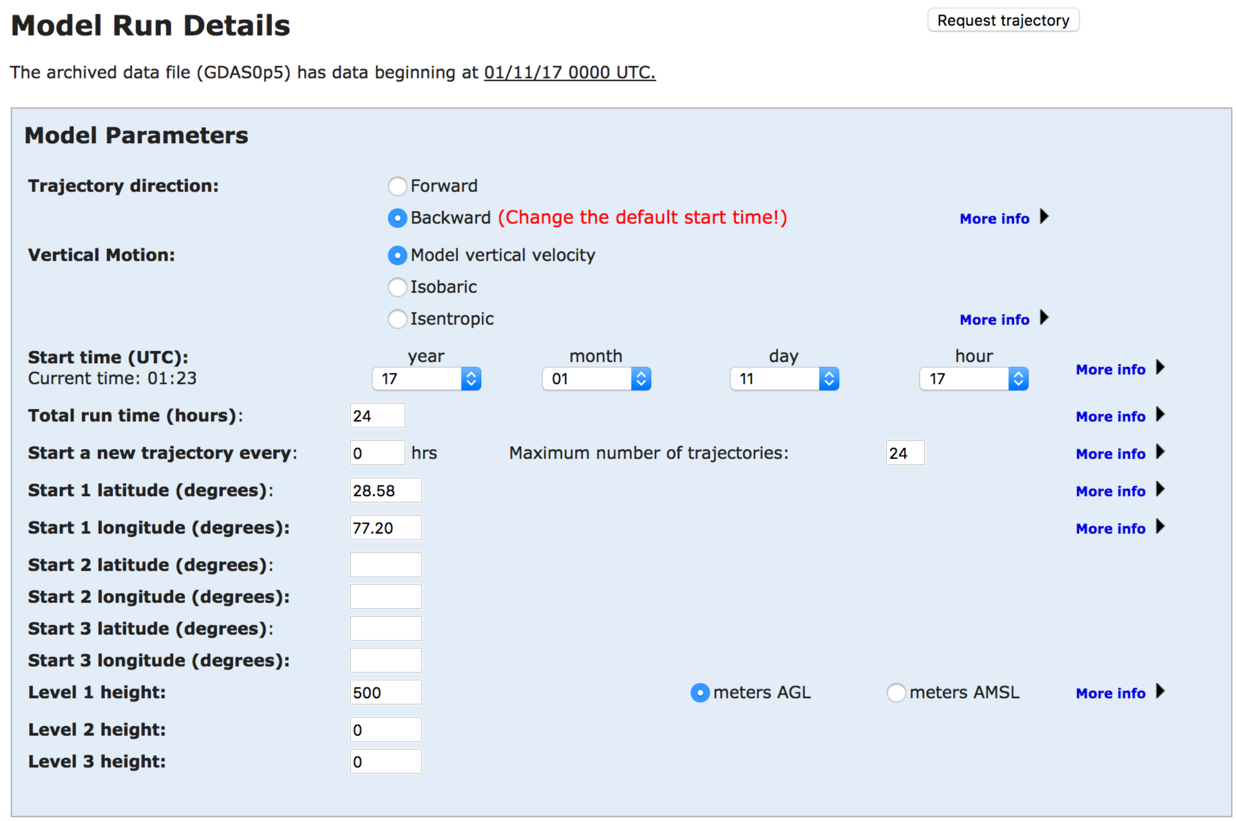

Figure 5. Model parameters in model run details.

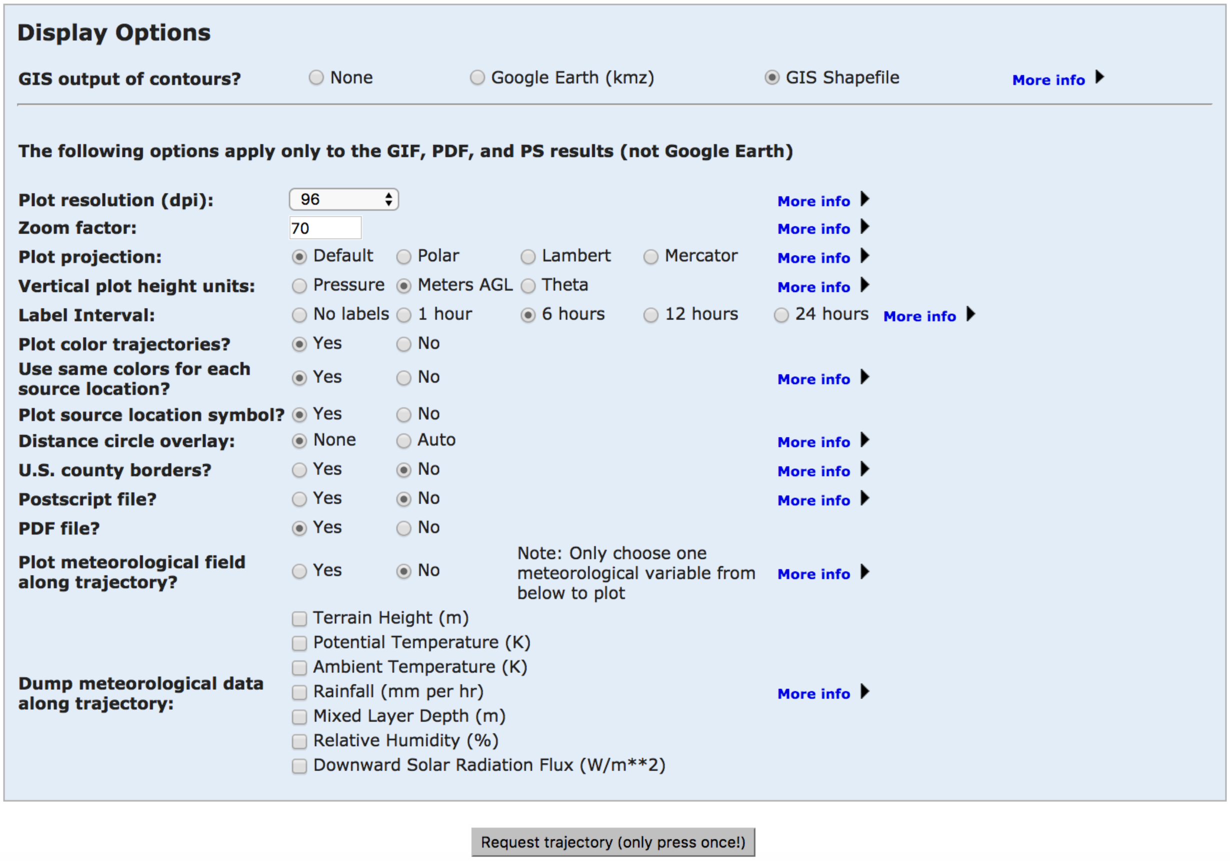

Figure 6. Display options in model run details.

1.5 HYSPLIT trajectory model results

Wait for the job to finish and save both the PDF plot (.pdf) and the zipped GIS shapefiles (.gis) file (Figure 7). If a warning is shown instead, the back trajectory cannot be run for that day. Put the PDF plots and GIS shapefiles in separate folders and make different subfolders for each month and year of files, exactly as shown in the schematic in Figure 8.

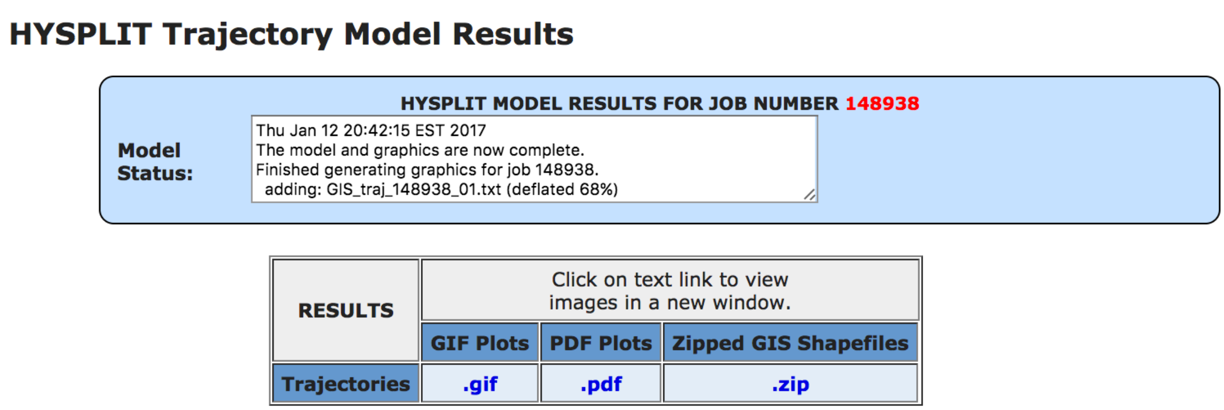

Figure 7. HYSPLIT trajectory model results.

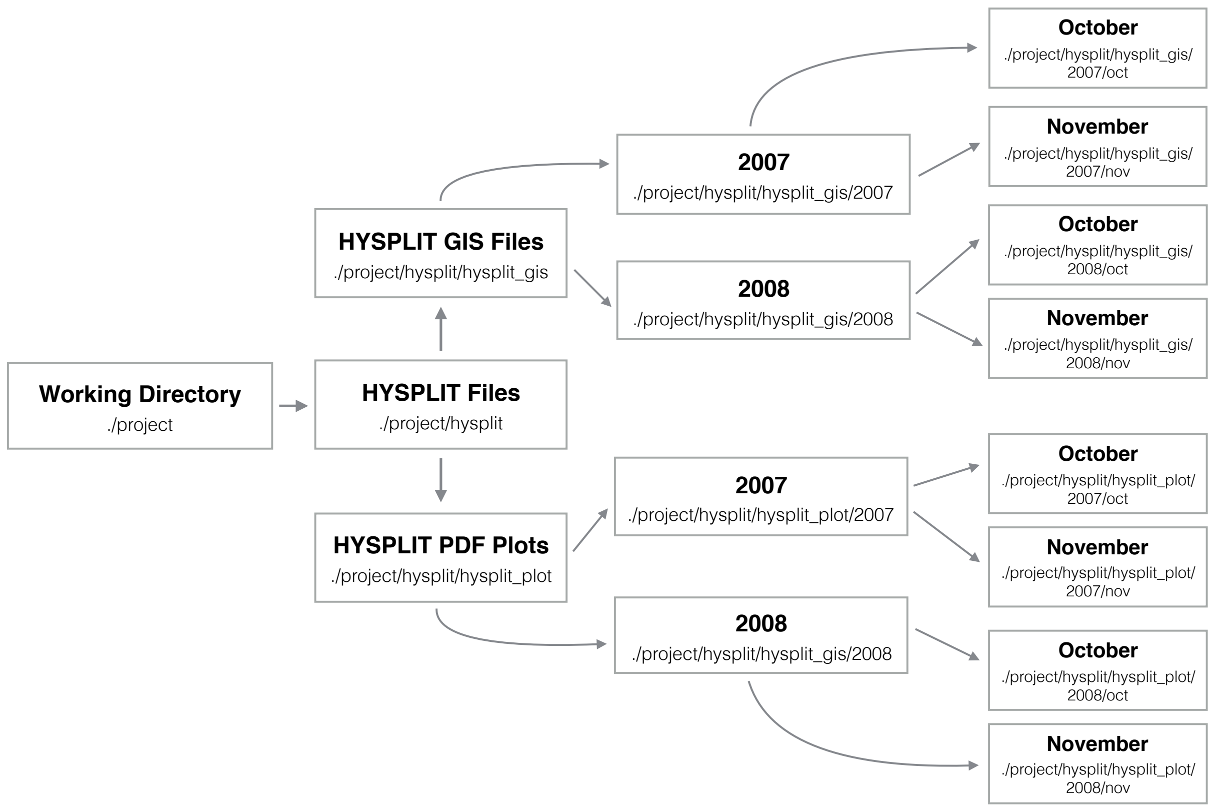

Figure 8. Hierarchy for folders to organize HYSPLIT files. Example for October and November in 2007-2008. Directory paths shown are for Mac OS.

2. Obtaining MODIS FRP rasters

MODIS datasets are archived at NASA Reverb (https://reverb.echo.nasa.gov/reverb/#utf8=%E2%9C%93&spatial_map=satellite&spatial_type=rectangle). In Liu et al. (in revision), we used the sum of Terra and Aqua FRP to capture the fire intensity at both overpass times. The overpass time of the Terra sensor is 10:30 am local time and that of the Aqua sensor is 1:30 pm local time.[5]

2.1 Downloading MODIS granules: Using Reverb and Terminal

Draw a bounding box around the area of interest in **spatial search** (Figure 9). Enter MOD14A1 for Terra sensor active fire product (repeat all steps for MYD14A1, which is the product code for the Aqua sensor) in **search terms**. Enter the start and end date of study in **temporal search**.

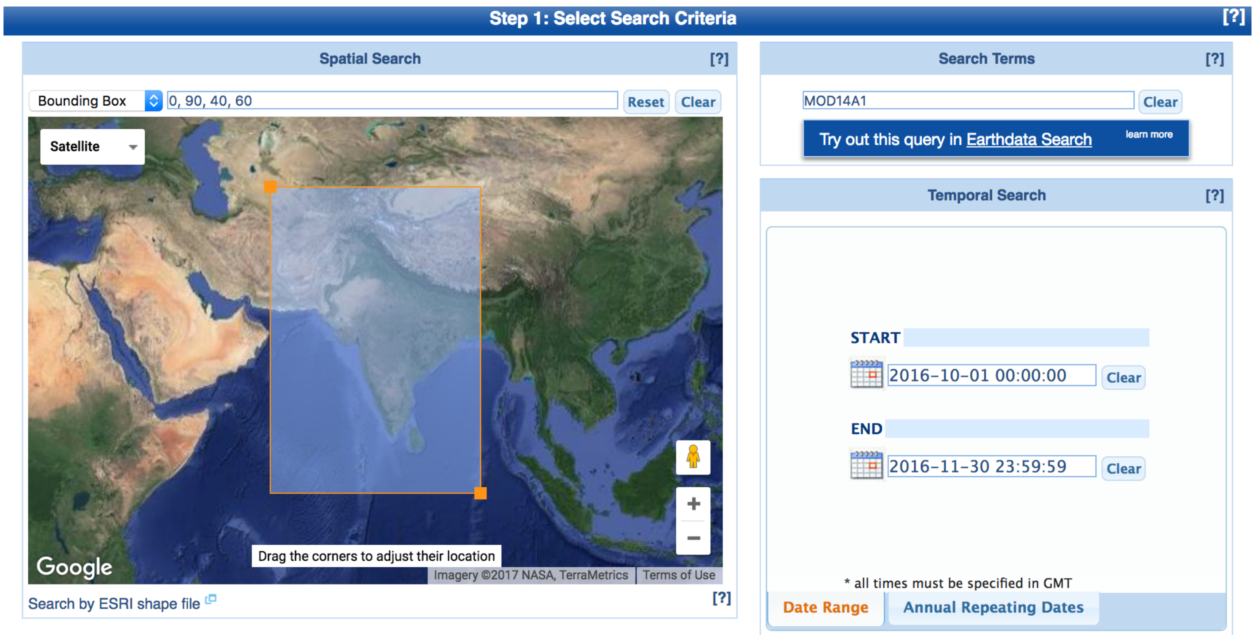

Figure 9. Step 1 in Reverb search. An example of a bounding box drawn for India, selection of MOD14A1 (Terra) datasets in October and November, 2016.

Select "MODIS/Terra (Aqua) Thermal Anomalies/Fire Daily L3 Global 1km SIN Grid V006" and search for granules (Figure 10).

[5] https://nsidc.org/data/modis/terra_aqua_differences

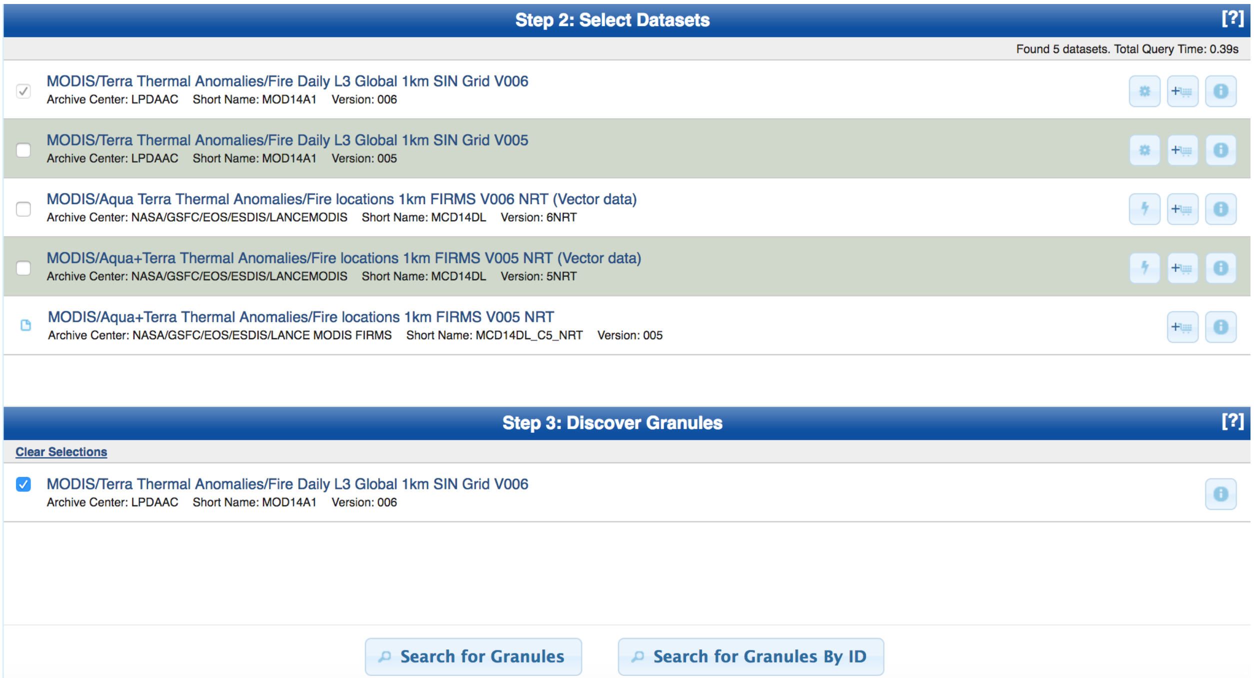

Figure 10. Steps 2 & 3 in Reverb search.

Add all granules to the shopping cart by clicking on the small shopping cart icon with "All" labeled underneath (Figure 11). Then, view items in cart.

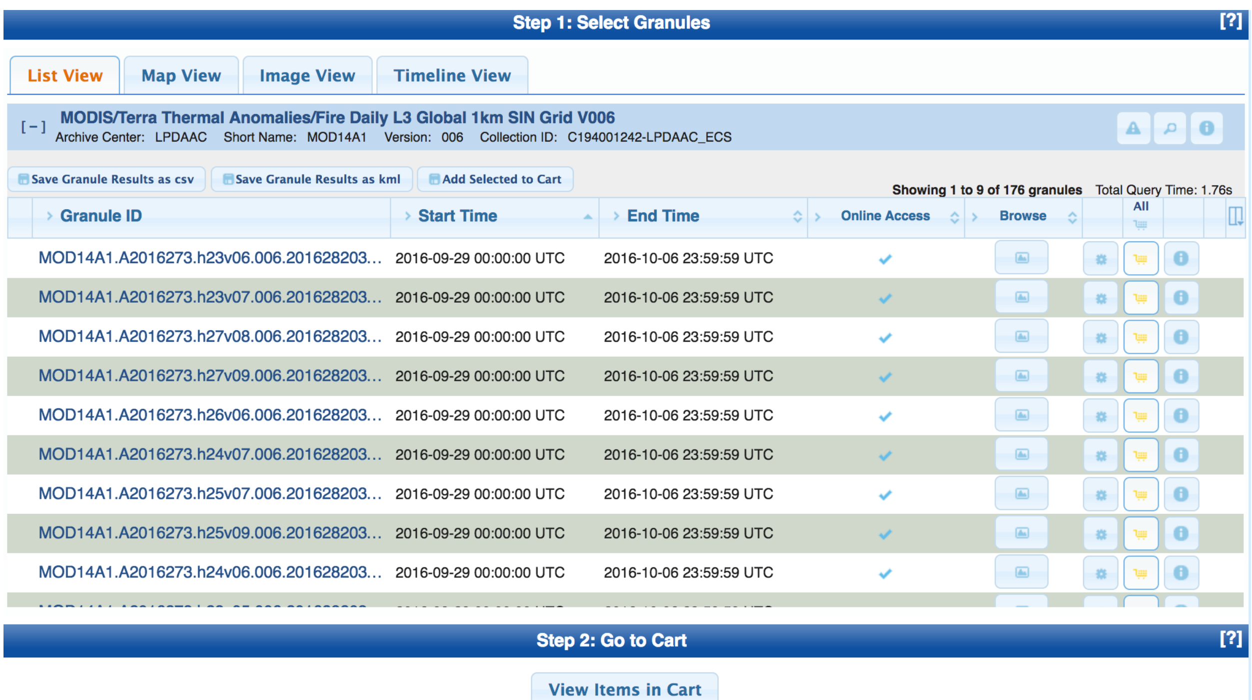

Figure 11. Select and add granules to cart.

Download the text file containing the urls for the MODIS granules (Figure 12).

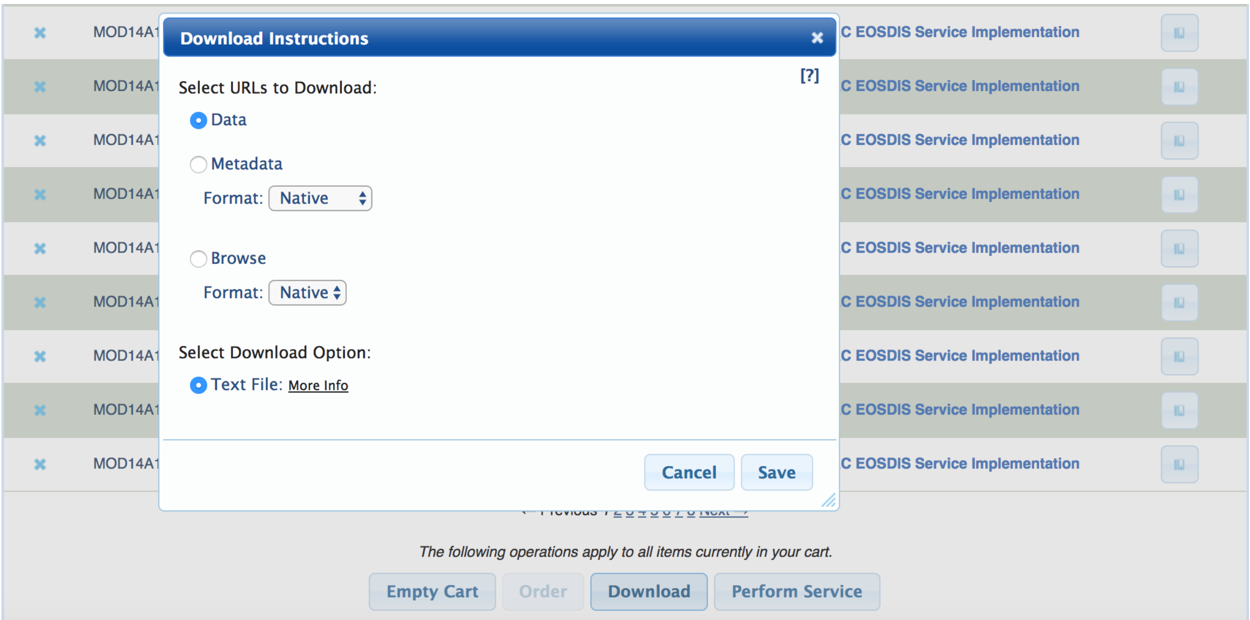

Figure 12. Download granule urls to a text file.

Make a temporary subdirectory to process the .hdf files. In Terminal, set that folder as the working directory and use the **wget** command to download the .hdf files of the MODIS granules. Add the text file to the working directory. (For Mac OS users, first download Xcode[6] if not on machine.) For example, in Terminal, run the lines:

```
cd /Users/TLiu/Desktop/project/temporary
wget -i data_url_script_2017-01-12_221013.txt
```

Shorten the names of the files using the **batch_renameHDF.sh** file (add file to working directory folder) with the following command in Terminal:

```
sh batch_renameHDF.sh
```

batch_renameHDF.sh is simply the following lines of code stored in a .sh file:

```
for file in M*hdf
do
    mv "${file}" "${file/006.*/hdf}"
done
```

Revise "006" if using MODIS Collections other than Collection 6.

[6] https://itunes.apple.com/us/app/xcode/id497799835?mt=12

2.2 Batch process: Using the MODIS Reprojection Tool

The MODIS Reprojection Tool (https://lpdaac.usgs.gov/tools/modis_reprojection_tool) takes granules and mosaics them together into a single file. It can also reproject from sinusoidal to Mercator, among other projections, and convert from .hdf to GeoTiff. Install the MODIS Reprojection Tool in the same subdirectory as the raw .hdf files by setting the working directory as that folder and running the downloaded file **mrt_install** in Terminal. Alternatively, dragging **mrt_install** into Terminal after setting the directory works as well. A prompt asking y/n to install should pop up. For example,

```
cd /Users/TLiu/Desktop/project/raw_files/2016
/Users/TLiu/Desktop/project/raw_files/2016/mrt_install
```

The template .prm file **batchMODIS.prm** includes sample parameters generated from the MODIS Reprojection Tool. The resampling method uses nearest neighbor. The output datum is WGS84, and the output projection is Mercator at 1000 m x 1000 m pixel size (Note that if dealing with areas for later calculations, the real area of the output pixels will not be 1-km^2 due to the poleward distortion of the Mercator projection.[7]) If study area is not India, either revise the bounding box coordinates or input granules of the same 8-day composite in the MODIS Reprojection Tool (go to MRT > bin > **ModisTool.jar**), select parameters, and save the .prm file.

Then, run the **batch_process.sh** loop and generate all daily max FRP .tif files in a given year from the 8-day .hdf composites. Revise "DATELIST" in **batch_process.sh** as necessary if subsetting particular days in a given year; replace MOD or MYD if switching between Terra and Aqua. Use the following line of code the run batch process:

```
sh batch_process.sh
```

Store the FRP .tif files and raw .hdf files from Section 2.1 in the hierarchical structure described in Figure 13.

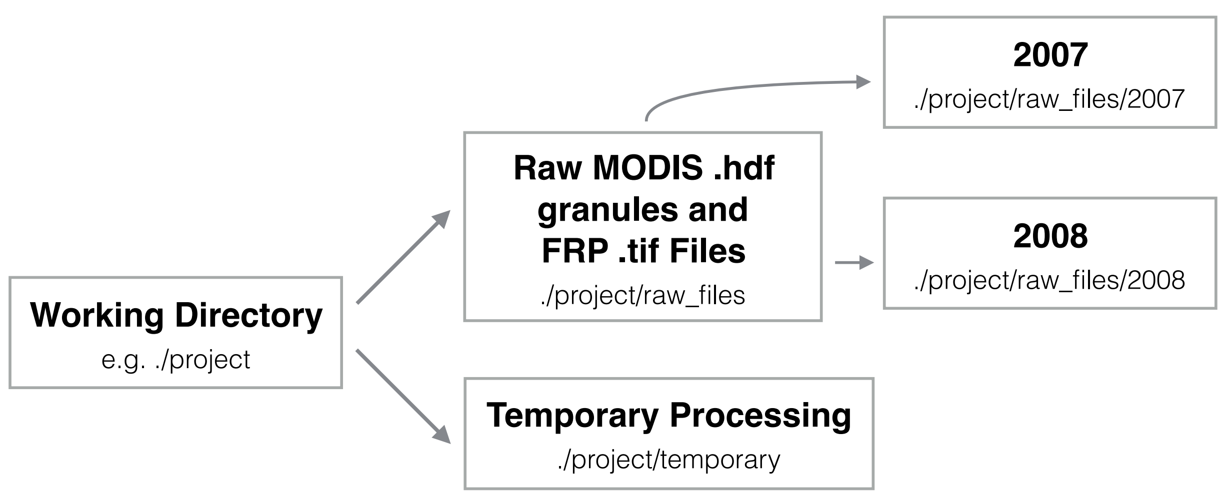

Figure 13. Hierarchy of MODIS raw .hdf and Terra and Aqua FRP .tif files. An example for 2007-2008.

[7] To get the real area, reproject the raster to an equal-area projection, such as Lambert.

2.3. Stack Terra + Aqua FRP: Using RStudio and QGIS

The next step is to stack and sum daily Terra and Aqua FRP using RStudio[8], and mask any ocean FRP (FRP picks up offshore oil flares; there are also anomalous blocks of FRP in the ocean in certain years). Use of the R functions and scripts described in the following sections requires the installation of these packages: **raster**, **rgeos**, **maptools**, **mapproj**, **maps**, **rgdal**, **spdep**, **sp**, **fields**, and **devtools**. To use the packages and attach them to the workspace, use **library()**.

2.3.1. Ocean mask

First, we need to create an ocean mask using the **ne_10m_admin_0_countries** shapefile from Natural Earth (http://www.naturalearthdata.com/downloads/10m-cultural-vectors/10m-admin-0-countries/) by using QGIS[9] and R. Put the unzipped shapefile in a subdirectory named "shapefiles" under the project home directory (The shapefile is the default to plot airsheds for the **plot.airshed**, **plot.airshedM**, and **plot.airshedC** functions.) In QGIS, load the vector shapefile **ne_10m_admin_0_countries.shp**, and use rasterize (Raster > conversion > Rasterize (Vector to Raster)) with MAPCOLOR7 as the attribute field. For "raster size in pixels," use a width of 10000 and height of 5000 (Note: higher resolution is more desirable, but may take longer to process in QGIS). Now, land has a value > 0 and ocean (and large bodies of water) = 0. Save as **countries.tif** in the "rasters" folder. In R, install the package **HyAirshed**[10] and its functions with the following commands:

```
install.packages("devtools")
library("devtools")
install_github("tianjialiu/HyAirshed-RPackage")
library("HyAirshed")
```

The **oceanMask** function can be used to obtain the **ocean_mask.tif** (Figure 14). The function **oceanMask** is used as follows:

```
oceanMask(...)
```

Note: This function should take a few minutes to finish.

[8] R (https://cran.rstudio.com/) must be installed to use RStudio (https://www.rstudio.com/products/rstudio/download/).

[9] QGIS (http://www.qgis.org/en/site/forusers/download.html) is an open source Geographic Information System (GIS) that can be used on both PC and MacOS.

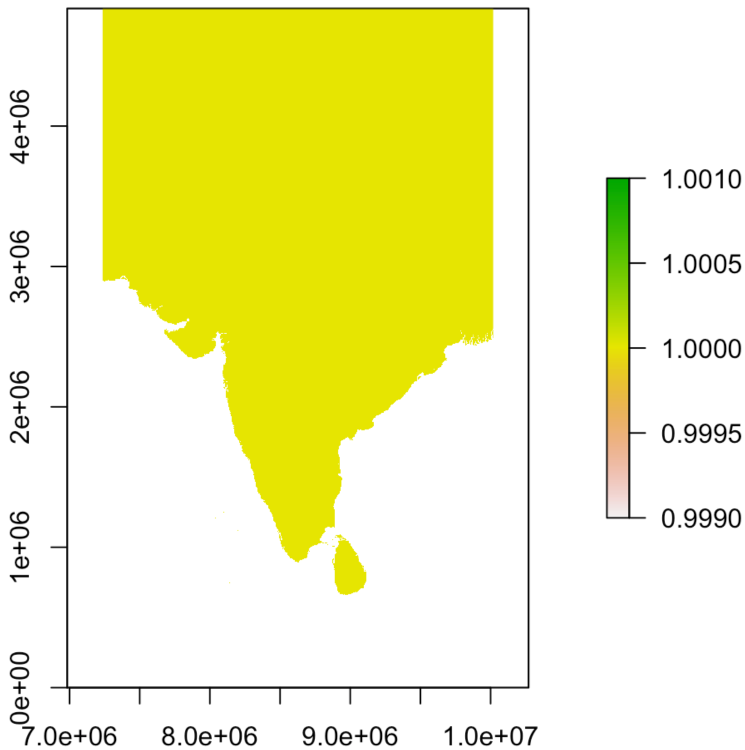

Figure 14. ocean_mask.tif. Visualization for plot(ocean_mask) after command is run in R console. Example of a ocean_mask.tif for India in Mercator projection (bounding box in degrees: x.min=60, x.max=90, y.min=0, y.max = 40).

2.3.2. Stack daily FRP

To stack Terra and Aqua FRP run the function **dailyFRPstack**. The function **dailyFRPstack** is used as follows:

```
dailyFRPstack(year_select, sDay, eDay, frp_mode="combined",
ocean_mask=TRUE, ...)
```

Arguments:

`year_select` = numeric vector. Years to be run
`sDay` = character. First day of starting 8-day composite*
`eDay` = character. First day of starting 8-day composite*
`frp_mode` = character: "terra", "aqua", or "combined". Choose Terra-only, Aqua-only, or Terra+Aqua FRP. [default is "combined"]
`ocean_mask` = logical. Mask ocean? [default is TRUE]

*Possible inputs for sDay and eDay: 1, 9, 17, 25, 33, 41, 49, 57, 65, 73, 81, 89, 97, 105, 113, 121, 129, 137, 145, 153, 161, 169, 177, 185, 193, 201, 209, 217, 225, 233, 241, 249, 257, 265, 273, 281, 289, 297, 305, 313, 321, 329, 337, 345, 353, 361

The output is stored as .tif files in subfolders named after the year of the file (e.g. FRP2007001.tif would be located in rasters > frp_daily > 2007). Subfolders are automatically generated by **dailyFRPstack**. Note: Outputs from the different modes are located in separate subfolders in the "rasters" directory: combined Aqua+Terra output in "frp_daily," Terra only in "frp_daily_terra," and Aqua only in "frp_daily_aqua." This function is intensive and can take hours to complete if spanning multiple years.

2.3.3. Stack monthly and seasonal FRP

To stack daily FRP from the **dailyFRPstack** output to monthly or seasonal, use the function **modeFRPstack**. In this case, we refer to seasonal as any number of days from 1-day to a 2-year period that is supported by the function. The function **modeFRPstack** is used as follows:

```
modeFRPstack(loc="city", day_select=NA, month_select=NA
year_select, frp_mode="seasonal", season_name="season",
next_yr=FALSE, ...)
```

Arguments:

`loc` = Name of location, starting point, used in naming subdirectories for mode="seasonal" [default is "city"]
`day_select` = numeric vector, integers, [1,365]. Days to be run, in Julian days, non-leap years, useful if mode="seasonal"
`month_select` = numeric vector. Months to be run, useful if mode="monthly"
`year_select` = numeric vector. Years to be run
`frp_mode` = character: "terra", "aqua", or "combined". Choose Terra-only, Aqua-only, or Terra+Aqua FRP. [default is "combined"]
`stack_select` = numeric vector: if "monthly" [1,12], if "seasonal", [1,365], format is Julian days in non-leap year
`season_name` = character. Name of season, used in the names of the output subfolder and file [default is "season"]
`next_yr` = numeric [1,365]. Days in next year, format is Julian days in non-leap year, useful for stacking days in winter months, only for "seasonal" mode [default is FALSE]
`...` see global arguments: **`rasDir`**, **`pointDir`**, **`ask_home`**

The output FRP .tif files will be in the format of FRP[yyyymm]monthly.tif for the "monthly" mode and FRP[yyyy][season_name].tif for "seasonal" mode. Files are organized in yearly subfolders in the "monthly" mode. Seasonal FRP rasters are stored in the "rasters" subfolder of the point location (see argument "loc") directory. This prevents overwriting of files for seasons of the same name but those that were defined differently. For example, in Liu et al. (in revision), we defined summer airsheds in the following way: Delhi (April-May), Bengaluru (February-March), and Pune (March-April). Monthly FRP rasters are stored in the "rasters" subdirectory of the project home directory.

3. Approximating and visualizing airsheds: monthly, seasonal, and composite

3.1 Preprocessing HYSPLIT trajectories

The HYSPLIT output used is the gis folders (shapefile and text file with hourly coordinates) and the pdf plots. Pdf plots are used to organize the gis folders, which are unzipped from .zip files and difficult to put in sequential order. The function **HyProcess** accesses the the pdf files and orders the text files in the gis folders. The coordinates in the text files are extracted and saved in .csv tables in the point location folder. Separate files are generated for latitude and longitude and each altitude. This process allows the user to run more than 3 altitudes to approximate airsheds. The function **HyProcess** is used as follows:

```
HyProcess(loc, year_select, hy_alt, nHours, alt.adj=0, ...)
```

Arguments:

`loc` = character. Name of location, starting point, used in naming subdirectories and files
`year_select` = numeric vector. Years to be run
`hy_alt` = character string or vector. Name(s) of hysplit heights to be run, up to 3
`nHours` = numeric. Total hours in back trajectories
`alt.adj` = numeric. Adjustment if more than one location [default is 0]
`...` see global arguments: **`hyDir`**, **`pointDir`**, **`ask_home`**

3.2 Approximating airsheds

To approximate the seasonal and/or monthly airshed of a point location, use the function **airshed**. The seasonal and monthly airsheds are approximated using the density of back trajectories per grid cell. Grid cells whose density is higher than a set threshold (e.g. >10% total number of trajectories) are considered as part of the seasonal airshed. A composite airshed for seasonal airsheds spanning multiple years can also be constructed with this function by again using a density threshold. The function **airshed** is used as follows:

```
airshed(loc, lon, lat, day_select=NA, month_select=NA,
year_select,
tYears=year_select[1]:year_select[length(year_select)],
season_name="season", mode="seasonal", hy_alt, alt.adj=0,
sPix=0.3, nPix=100, cutoff=10, composite=TRUE, ...)
```

Arguments:

`loc` = character. Name of location, starting point, used in naming subdirectories and files
`lon` = longitude of starting point
`lat` = latitude of starting point
`day_select` = numeric vector, integers, [1,365]. Days to be run, in Julian days, non-leap years, useful if mode="seasonal"
`month_select` = numeric vector. Months to be run, useful if mode="monthly"
`year_select` = numeric vector. Years to be run

tYears = Total range of years in study, ignoring missing years [default = year_select[1]:year_select[length(year_select)], which is the years from the first year to last year in year_select
season_name = character. Name of season, used in the names of the output folder and file [default is "season"]
mode = character: "monthly", "seasonal", or "both". [default is "both"]
hy_alt = character string or vector. Name(s) of hysplit heights to be run
sPix = numeric. Size of pixel, in degrees [default is 0.3]
nPix = numeric. Number of pixels in horizontal and vertical direction [default is 100]
cutoff = numeric. Threshold, in percent, for pixels to be considered part of seasonal airshed [default to 10]
composite = logical. Create composite airshed? [default is TRUE]
... see global arguments: **pointDir**, **ask_home**

Note: This function is intensive and can take a more than 10 minutes to complete if spanning multiple years, months, and heights.

3.3 Visualizing and plotting airsheds

To visualize the seasonal airshed and HYSPLIT back trajectories of a point location, use the function **plot.airshed**. The function **plot.airshed** is used as follows:

```
plot.airshed(loc, lon, lat, month_select, year_select,
tYears=year_select[1]:year_select[length(year_select)],
season_name="season",
borderName="ne_10m_admin_0_countries", hy_alt, alt.adj=0,
sPix=0.25, nPix=100, x.adj=FALSE, zoom=4.5, x.just=0,
y.just=0, traj.col=c("#a6cee3","#fb9a99","#cab2d6"),
traj.lwd=0.3, nKM=150, col.adj=c(0,0,0,0),
scale.adj=c(0,0), bar.adj=c(0,0,0,0), caption=NA,
caption.adj=c(0,0), nAlt.adj=c(0,0), legend.lab=hy_alt,
legend.adj=c(0,0), txt_size=0.75, ...)
```

Arguments:
loc = character. Name of location, starting point, used in naming subdirectories and files
lon = longitude of starting point
lat = latitude of starting point
month_select = numeric vector. Months to be run
year_select = numeric vector. Years to be run
tYears = numeric vector. Total range of years in study, ignoring missing years [default = year_select[1]:year_select[length(year_select)], which is the years from the first year to last year in year_select
season_name = character. Name of season, used in the names of the output folder [default is "season"]
hy_alt = character string or vector. Name(s) of hysplit heights to be run
sPix = numeric. Size of pixel, in degrees [default is 0.25]
nPix = numeric. Number of pixels in horizontal and vertical direction [default is 100]

borderName = character. Name of shapefile located in the "shapefiles" folder of project home directory to plot country/sub-country borders [default is “ne_10m_admin_0_countries”]
x.adj = logical. Adjust x axis latitude to 0-360? default is FALSE (-180 to 180)
zoom = numeric. Window of the plot, in degrees, from the lon, lat of the starting location in all 4 directions [default is 4.5]
x.just = numeric. Adjust window laterally, positive is right and negative is left [default is 0]
y.just = numeric. Adjust window vertically, positive is right and negative is left [default is 0]
traj.col = character string or vector in html or R color name. Colors of trajectories [default is c("#a6cee3","#fb9a99","#cab2d6")]
traj.lwd = numeric. Width of trajectory lines [default is 0.3]
nKM = numeric. Length of scale bar, in kilometers [default is 150]
col.adj = numeric vector, length=4. Adjust color bar position/size (format is c(x.min, x.max, y.min, y.max) adjustment) [default is c(0,0,0,0)]
scale.adj = numeric vector, length=2. Adjust scale bar position (format is c(x, y) adjustment), [default is c(0,0)]
bar.adj = numeric vector, length=4. Adjust timestamp bar position/size (format is c(x.min, x.max, y.min, y.max) adjustment), [default is c(0,0,0,0)]
caption = character or expression. Name of caption, [default is NA], which is no caption
caption.adj = numeric vector, length=2. Adjust caption position (format is c(x, y) adjustment) [default is c(0,0)]
nAlt.adj = numeric vector, length=2. Adjust number of trajectories label position (format is c(x, y) adjustment) [default is c(0,0)]
legend.lab = character vector. Labels for back trajectories [default is hy_alt]
legend.adj = numeric vector, length=2. Adjust legend position (format is c(x, y) adjustment), [default is c(0,0)]
txt_size = numeric. Adjust text size of plot labels, based on size of axis tick labels [default is 0.75]

This function may require some tinkering with the adjustment variables depending on the output or elements of the plot to be moved. The elements of the plot (caption, legend, timestamp, scale bar) are plotted in a window that uses x-limits of [0,10] and y-limits of [0,10] rather than the same window used for the trajectories and airsheds. The **ne_10m_admin_0_countries.shp** shapefile used to create the ocean mask is used as default to plot the background border. A sample output of this function is shown in Figure 15. The output timestamps can be converted into .jpeg files and combined into a .gif video at gifcreator.me.

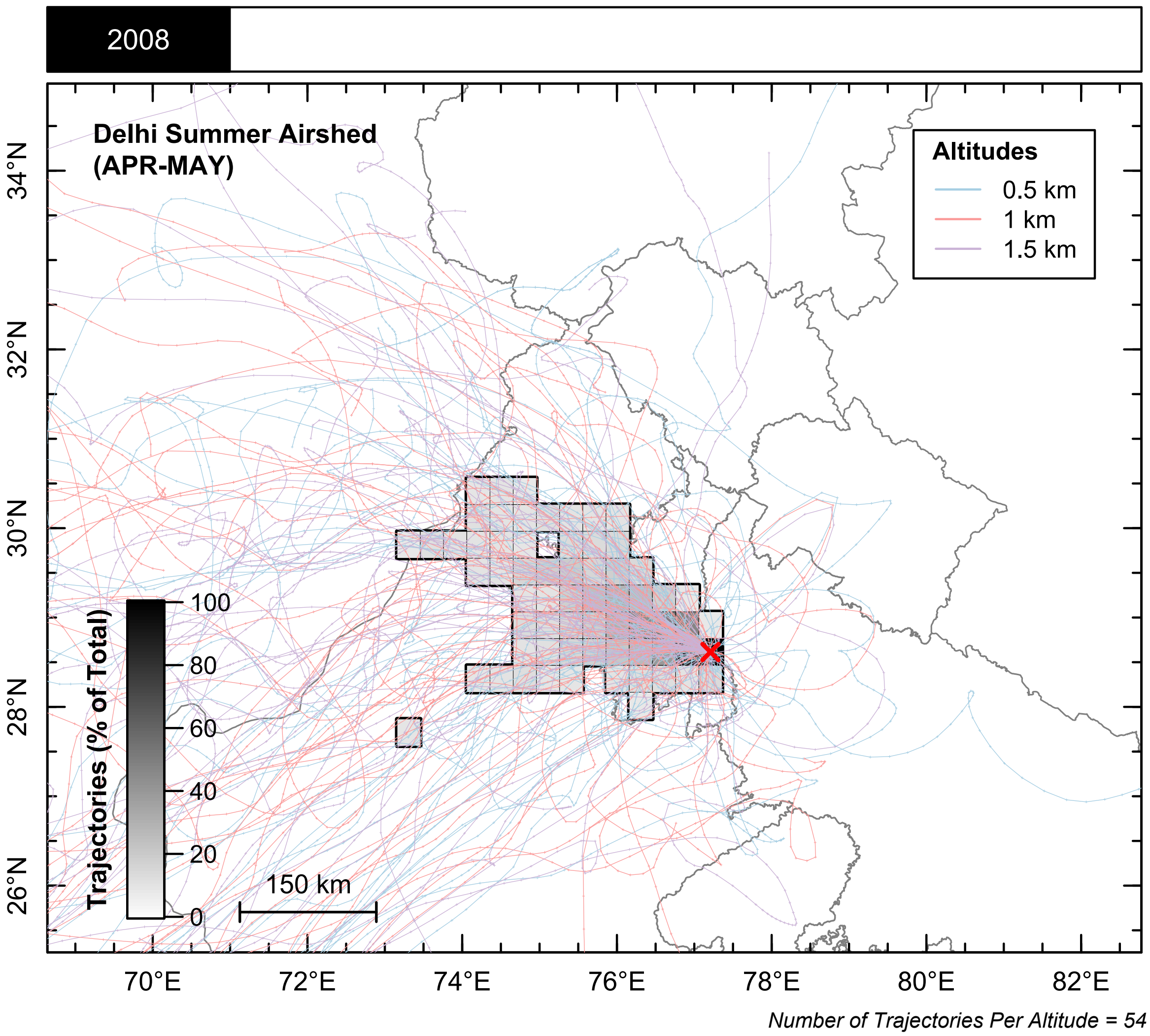

Figure 15. A sample output of the function **plot.airshed**. Summer (April-May) airshed of Delhi in 2008 with 3 heights: 0.5 km, 1 km, and 1.5 km.

A similar function **plot.airshedM** and **plot.airshedC** with the many of the same arguments as **plot.airshed** plots monthly airsheds with yearly and monthly timestamps and the composite airshed of more than one seasonal airshed, respectively. An example is shown in Figure 16 and Figure 17 for **plot.airshedM** and **plot.airshedC**, respectively. The functions **plot.airshedM** and **plot.airshedC** are used as follows:

```
plot.airshedM(loc, lon, lat, month_select, year_select,
tYears=year_select[1]:year_select[length(year_select)],
hy_alt, alt.adj=0, sPix=0.25, nPix=100,
borderName="ne_10m_admin_0_countries", x.adj=FALSE,
zoom=4.5, x.just=0, y.just=0, nKM=150,
traj.col=c("#a6cee3","#fb9a99","#cab2d6"), traj.lwd=0.25,
col.adj=c(0,0,0,0), scale.adj=c(0,0), bar.adj=c(0,0,0,0),
bar2.adj=c(0,0,0,0), caption=NA, caption.adj=c(0,0),
nAlt.adj=c(0,0), legend.adj=c(0,0), txt_size=0.75, ...)
```

The argument “bar2.adj” in **plot.airshedM** refers to adjustment of the monthly timestamp.

```
plot.airshedC(loc, lon, lat, season_name,
borderName="ne_10m_admin_0_countries", x.adj=FALSE,
zoom=4.5, x.just=0, y.just=0, nKM=150, col.adj=c(0,0,0,0),
scale.adj=c(0,0), caption=NA ,caption.adj=c(0,0),
nYrs.adj=c(0,0), legend.adj=c(0,0), txt_size=0.75, ...)
```

The argument “nYrs.adj” in **plot.airshedC** is used similarly to “nAlt.adj” but shows the number of years in the composite rather than the number of trajectories per altitude.

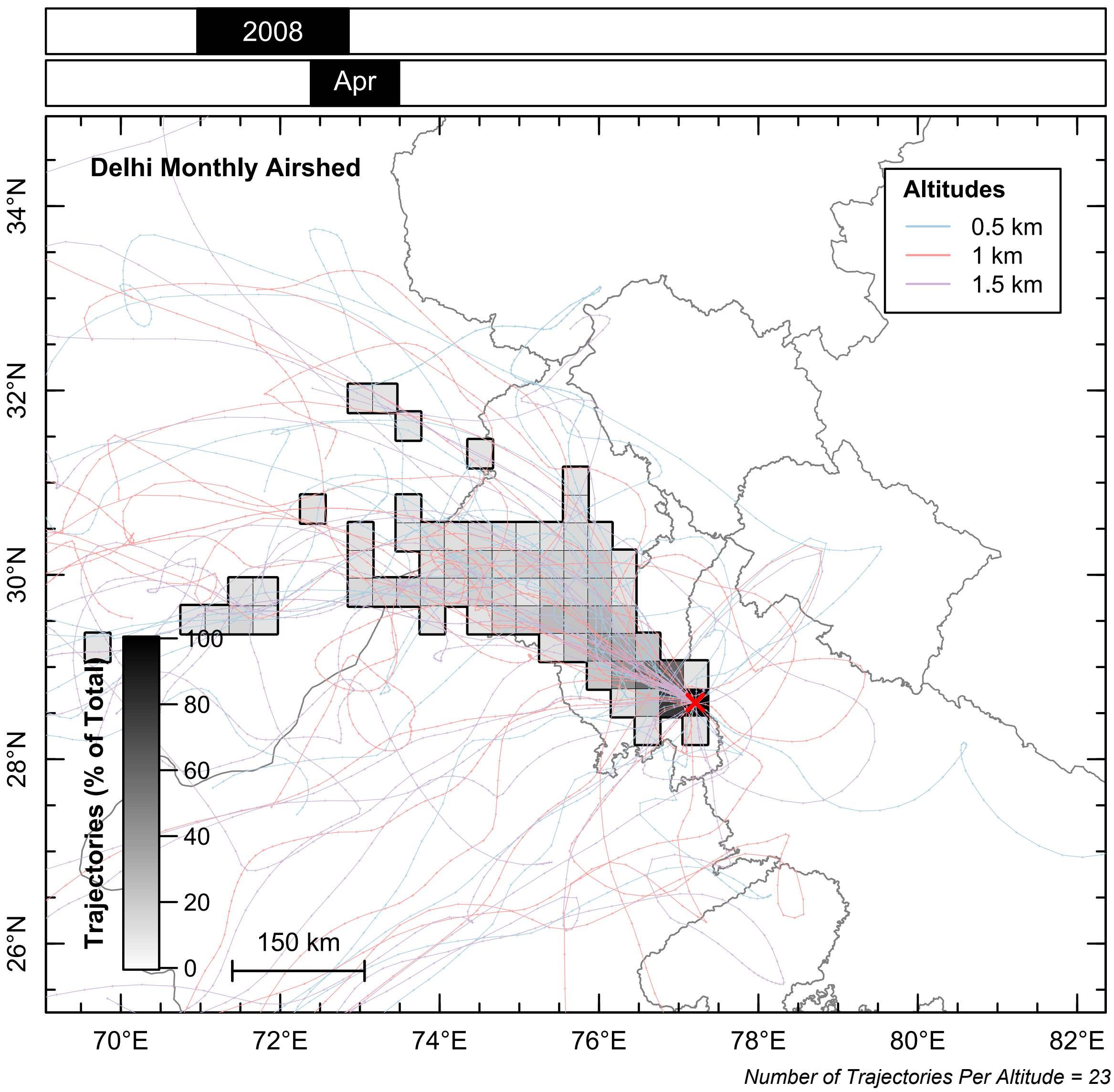

Figure 16. A sample output of the function **plot.airshedM**. April airshed of Delhi in 2008 with 3 heights: 0.5 km, 1 km, and 1.5 km.

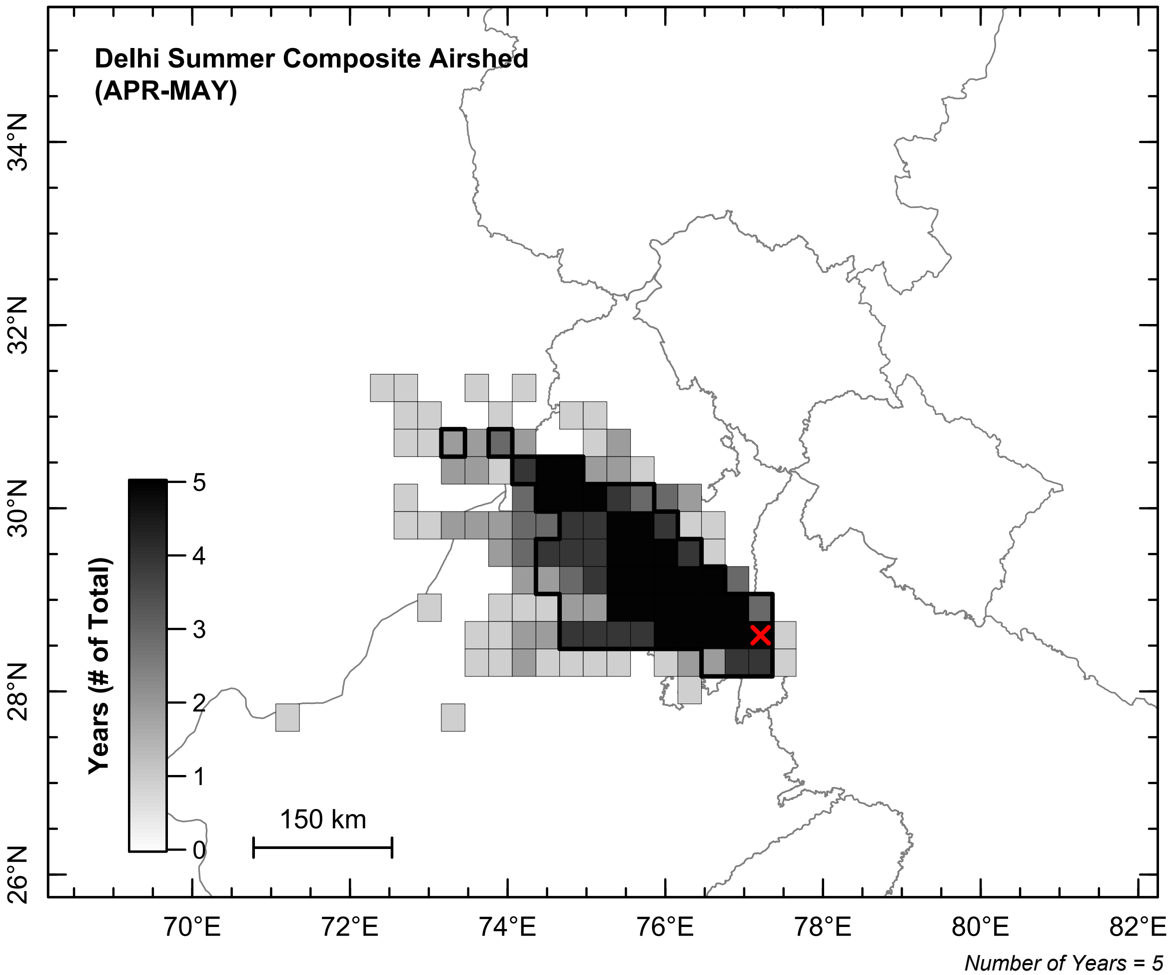

Figure 17. A sample output of the function **plot.airshedC**. Composite summer (Apr-May) airshed of Delhi in 2008-2009, 2011-2013 with 3 heights: 0.5 km, 1 km, and 1.5 km.

4. FRP in HYSPLIT-derived airsheds

4.1 Daily

Use the function **dailyHyFRP** to extract daily FRP from HYSPLIT-derived airsheds and save to summary .csv tables. The function compensates for the timing of HYSPLIT back trajectories and daily FRP. For example, for 72-hour back trajectories started from Delhi at 17:00 local time, the trajectories are divided into four parts that are used to extract the daily FRP corresponding to the timing of that part of the trajectory. Therefore, some minimal overlap in the FRP sum may exist. Make sure **hysplitDays.csv**, which includes information about which trajectory runs are not limited by meteorology, is located in the project home directory. The function **dailyHyFRP** is used as follows:

```
dailyHyFRP(loc, month_select, year_select, buf=10000,
hy_alt, alt.adj=0, local_time, frp_mode="combined", ...)
```

Arguments:

loc = character. Name of location, starting point, used in naming subdirectories and files
month_select = numeric vector. Months to be run
year_select = numeric vector. Years to be run
buf = numeric. Width of buffer, in meters, from original trajectory [default is 10000]
hy_alt = character string or vector. Name(s) of hysplit heights to be run
alt.adj = numeric. Adjustment if more than one location [default is 0]
local_time = numeric. local time of HYSPLIT starting time, in hours
frp_mode = character: "terra", "aqua", or "combined". Choose Terra-only, Aqua-only, or Terra+Aqua FRP. [default is "combined"]

The total and subtotal FRP are summarized in .csv files in the "daily frp" subfolder under the point location subdirectory. Be careful when dealing with trajectories in early January; the FRP for the last few days of December of the previous year must also be in the "rasters" folder if trajectories trace back to December. Note: This function is the most intensive and can take hours to complete if spanning multiple years, months, and heights.

4.2 Monthly and seasonal

Use the function **modeHyFRP** to extract monthly or seasonal FRP from HYSPLIT-derived airsheds and save to a summary .csv table. The function **modeHyFRP** is used as follows:

```
modeHyFRP(loc, month_select, year_select,
season_name="season", mode="seasonal", frp_mode="combined",
...)
```

Arguments:

loc = character. Name of location, starting point, used in naming subdirectories and files
month_select = numeric vector. Months to be run
year_select = numeric vector. Years to be run
season_name = character. Name of season, used in the name of the output table when mode="seasonal" [default is "season"]
frp_mode = character: "terra", "aqua", or "combined". Choose Terra-only, Aqua-only, or Terra+Aqua FRP. [default is "combined"]

mode = character: "monthly" or "seasonal". [default is "seasonal"]

Outputs are stored in the "frp_airshed" subfolder of the point location (see argument "loc") subdirectory. An example of the output for **modeHyFRP** when mode="seasonal" is shown in Figure 18. The output in the "monthly" mode is flexible and can be run for months in more than one season. For example, if the monthly airsheds of the summer season are only available in 2008-2009, 2011-2013 and the monthly airsheds are available for 2007-2013 for Delhi, the month_select arguments for **modeHyFRP** should be monthly_select=c(4:5, 10:11) and year_select=2007-2013; the function inputs NA for months not selected or monthly airsheds not available for that month and year. The output is shown in Figure 19. Note: This function is intensive and can take over an hour to complete if spanning multiple years and months.

	frp_win
2007	668561.4
2008	604010.1
2009	458104.8
2010	451952.5
2011	640103.9
2012	636721.3
2013	414418.5

Figure 18. A sample .csv output (as read in R) of the function **modeHyFRP** for FRP in the winter Delhi airshed in 2007-2013.

	jan	feb	mar	apr	may	jun	jul	aug	sep	oct	nov	dec
2007	NA	NA	NA	NA	NA	NA	NA	NA	NA	442770.4	165141.6	NA
2008	NA	NA	NA	19184.4	23854.2	NA	NA	NA	NA	299131.0	239285.7	NA
2009	NA	NA	NA	51353.0	63230.7	NA	NA	NA	NA	295658.9	117995.1	NA
2010	NA	NA	NA	NA	NA	NA	NA	NA	NA	111355.4	302455.9	NA
2011	NA	NA	NA	8829.2	44296.7	NA	NA	NA	NA	328037.9	238463.2	NA
2012	NA	NA	NA	5099.3	169312.7	NA	NA	NA	NA	236571.8	351411.9	NA
2013	NA	NA	NA	2214.1	90282.6	NA	NA	NA	NA	96389.9	241286.0	NA

Figure 19. A sample .csv output (as read in R) of the function **modeHyFRP** for FRP in the Delhi April and May airsheds in 2008-2009, 2011-2013 and October and November airsheds in 2007-2013.

5. Global arguments for functions in HyAirshed

When running each function in **HyAirshed**, R prompts the user whether the current directory is the project home directory. If running multiple functions in a R script, "ask_home" can be set to FALSE in each function to prevent the default interactive dialogue. Four subdirectories of the working directory can also be set to different paths: MODIS granules and FRP files (rawDir), HYSPLIT files (hyDir), raster files (rasDir), and point location files (pointDir). The folders, subfolders, and files of these subdirectories cannot be changed unless the functions are edited.

`ask_home` = logical. Ask for home directory in interactive mode? [default is TRUE]
`rawDir` = character. [default is "./raw_files/"], which is a sub directory of the project home directory for Aqua and Terra FRP .tif files from the MODIS Reprojection Tool
`hyDir` = character. [default is "./hysplit/"], which is a sub directory of the project home directory for HYSPLIT gis and pdf plots
`pointDir` = character. [default is paste0("./loc/")], which is a sub directory of the project home directory named after the point location, based on the argument "loc"
`rasDir` = character. [default is "./rasters/"], which is a sub directory of the project home directory for daily and monthly FRP rasters

An example of the complete hierarchy of folders is shown in Figure 20.

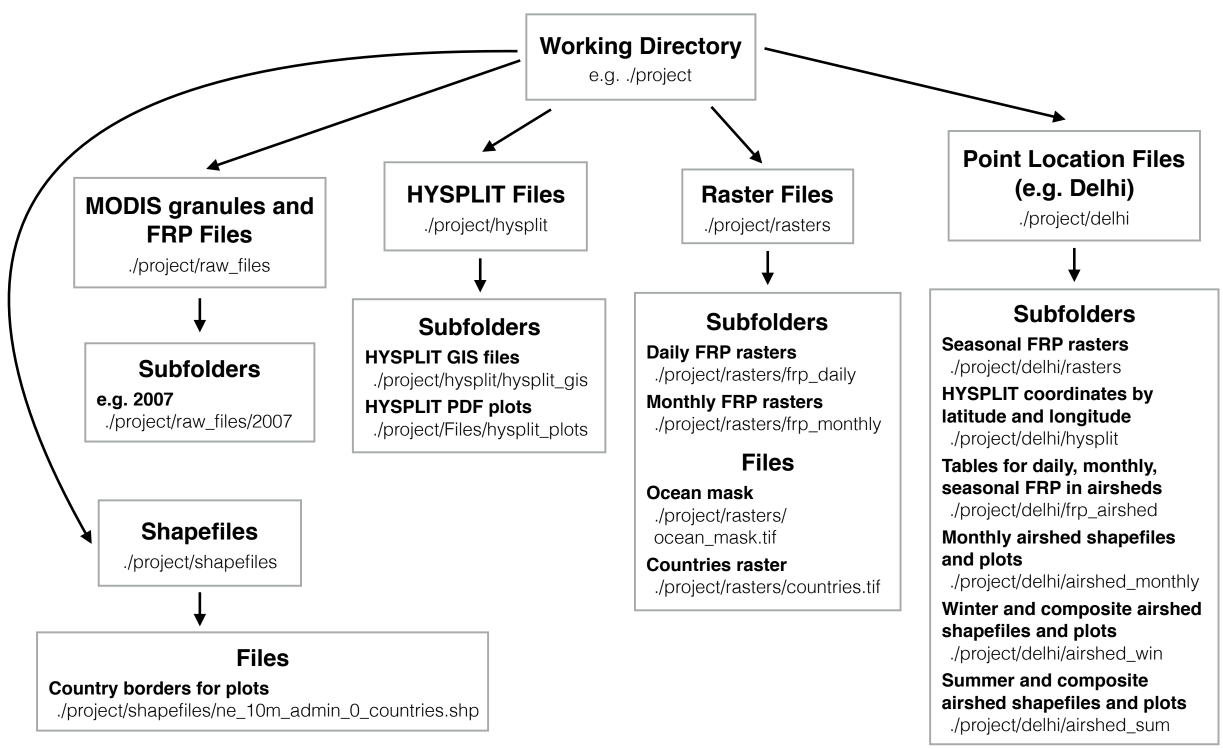

Figure 20. An example of the complete hierarchy of folders.