Weather data structure and processing

Tielens

2024-09-11

This is a brief introduction to the structure and attributes of weather surveillance data.

Weather surveillance radar data is systematically collected in national radar networks, and the accessibility and and storage of this data varies across countries. Typically, weather surveillance radars collect data every 5-8 minutes. National meteorological institutes can choose to make this data accessible to the public, including scientists interested in animal movement. Here, we demonstrate the downloading and processing of weather surveillance radar (WSR) data collected above the Netherlands by the KNMI (Koninklijk Nederlands Meteorologisch Instituut), to provide insight into the data structure.

In this document, we will: * download WSR data from KNMI * reading in data using dedicated R package 'biorad' * explore the data structure and resolution * demonstrate our existing data processing workflow that extracts biological signals (birds and insects) and removes noise * visualize several radar products to show the data on which we would like to deploy a classification algorithm

```
library("bioRad")
```

```
## Welcome to bioRad version 0.7.3.9000
```

```
## using vol2birdR version 1.0.1 (MistNet installed)
```

```
library("ggplot2")
library("tidyr")
library(vol2birdR)
```

```
##
## Attaching package: 'vol2birdR'
```

```
## The following object is masked from 'package:bioRad':
##
##     vol2bird_version
```

```
#library("")



setwd("/Users/elsketielens/postdoc/HiRAD/WSR/raw_scans/NEXRAD/")
```

```
setwd("/Users/elsketielens/postdoc/HiRAD/WSR/raw_scans/NEXRAD/")

download_pvolfiles(date_min=as.POSIXct("2023-10-05 23:00:00"), date_max=as.POSIXct("2023-10-05 23:40:00
```

```
## Downloading data from noaa-nexrad-level2 for radar KINX spanning over 1 days
```

```
##
## Downloading pvol for 2023/10/05/KINX/
```

```
##   |                                                                              |
```

```r
# store the names of downloaded files in my_pvolfiles
my_pvolfiles <- list.files(recursive = TRUE, full.names = FALSE, pattern="KINX")
# print to console our file names:
my_pvolfiles
```

```
##  [1] "2023/10/04/KINX/KINX20231004_230113_V06"
##  [2] "2023/10/04/KINX/KINX20231004_230643_V06"
##  [3] "2023/10/04/KINX/KINX20231004_231205_V06"
##  [4] "2023/10/04/KINX/KINX20231004_231719_V06"
##  [5] "2023/10/04/KINX/KINX20231004_232220_V06"
##  [6] "2023/10/04/KINX/KINX20231004_232742_V06"
##  [7] "2023/10/04/KINX/KINX20231004_233302_V06"
##  [8] "2023/10/04/KINX/KINX20231004_233823_V06"
##  [9] "2023/10/04/KINX20231004_190401_V06"
## [10] "2023/10/04/KINX20231004_191023_V06"
## [11] "2023/10/04/KINX20231004_191650_V06"
## [12] "2023/10/04/KINX20231004_192343_V06"
## [13] "2023/10/04/KINX20231004_193021_V06"
## [14] "2023/10/04/KINX20231004_193658_V06"
## [15] "2023/10/04/KINX20231004_194336_V06"
## [16] "2023/10/04/KINX20231004_195013_V06"
## [17] "2023/10/04/KINX20231004_195651_V06"
## [18] "2023/10/04/KINX20231004_195651_V06_MDM"
## [19] "2023/10/05/KINX/KINX20231005_230618_V06"
## [20] "2023/10/05/KINX/KINX20231005_231500_V06"
## [21] "2023/10/05/KINX/KINX20231005_232343_V06"
## [22] "2023/10/05/KINX/KINX20231005_233224_V06"
```

```r
setwd("/Users/elsketielens/postdoc/HiRAD/WSR/raw_scans/NEXRAD/")
# load polar volume data. let's just pick the first sweep
my_pvol<-read_pvolfile(my_pvolfiles[1])
```

```
## Filename = 2023/10/04/KINX/KINX20231004_230113_V06, callid = KINX
## Removed 1 SAILS sweep.
## Call RSL_keep_sails() before RSL_anyformat_to_radar() to keep SAILS sweeps.
## Reading RSL polar volume with nominal time 20231004-230113, source: RAD:KINX,PLC:TULSA,state:OK,radar
## Warning: dimensions of scan parameter 1 at elev 3.999025 do not match scan dimensions, resampling ..
## Warning: dimensions of scan parameter 2 at elev 3.999025 do not match scan dimensions, resampling ..
## Warning: dimensions of scan parameter 5 at elev 3.999025 do not match scan dimensions, resampling ..
## Warning: dimensions of scan parameter 9 at elev 3.999025 do not match scan dimensions, resampling ..
## Warning: dimensions of scan parameter 10 at elev 3.999025 do not match scan dimensions, resampling .
```

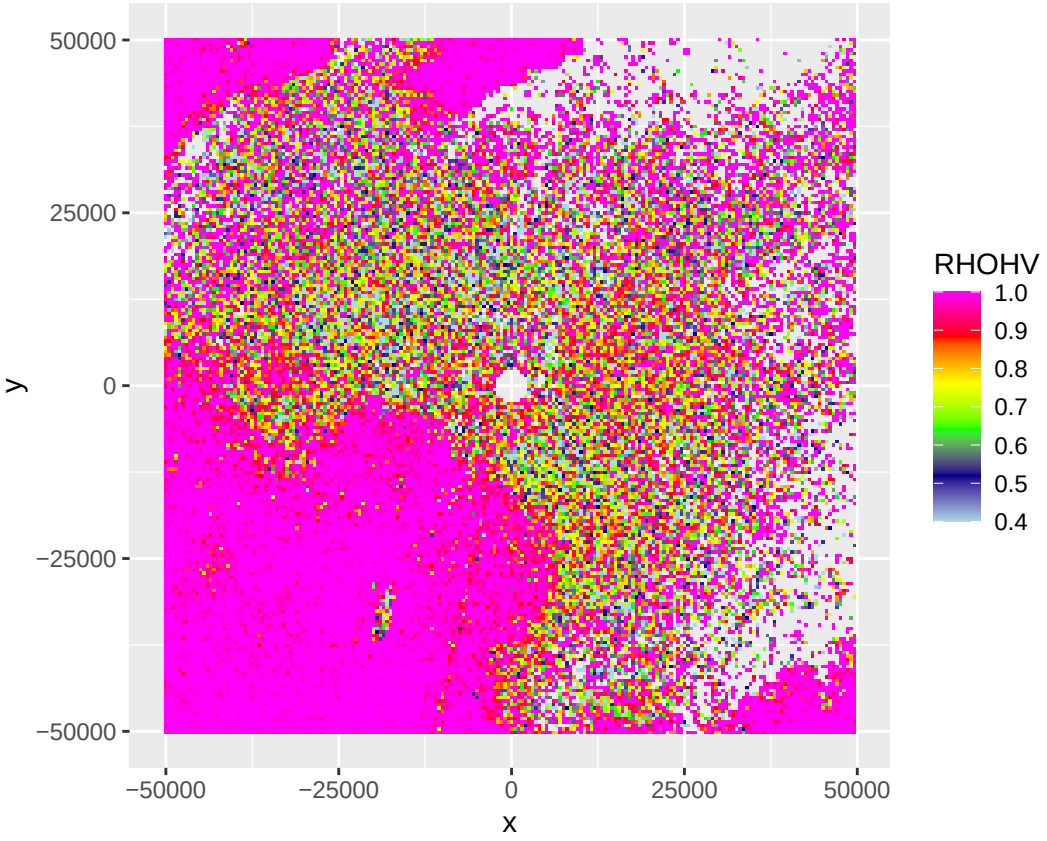
```r
# extract a single scan (biologists are usually primarily interested in the lowest elevation angle)
my_scan <- get_scan(my_pvol, 0.5)

# project this scan as planned position indicator
my_ppi<-get_scan(my_pvol, 0.5) %>%
    project_as_ppi()

read_pvolfile(my_pvolfiles[1]) %>%
    get_scan(0.5) %>%
    project_as_ppi() %>%
    plot(param = "RHOHV")
```

```
## Filename = 2023/10/04/KINX/KINX20231004_230113_V06, callid = KINX
```

```
## Removed 1 SAILS sweep.
## Call RSL_keep_sails() before RSL_anyformat_to_radar() to keep SAILS sweeps.
## Reading RSL polar volume with nominal time 20231004-230113, source: RAD:KINX,PLC:TULSA,state:OK,radar
## Warning: dimensions of scan parameter 1 at elev 3.999025 do not match scan dimensions, resampling ..
## Warning: dimensions of scan parameter 2 at elev 3.999025 do not match scan dimensions, resampling ..
## Warning: dimensions of scan parameter 5 at elev 3.999025 do not match scan dimensions, resampling ..
## Warning: dimensions of scan parameter 9 at elev 3.999025 do not match scan dimensions, resampling ..
## Warning: dimensions of scan parameter 10 at elev 3.999025 do not match scan dimensions, resampling .
```



Examine the data structure

- the data is stored in an array
- each sweep is a single time point (collected by the radar usually every 5 minutes)
- each sweep contains data from multiple altitudinal scans
- the spatial resolution of the data is an azimuthal grid of 0.5 degrees by 200 m, for 100-300 km (depending on the target)
- Each sweep contains 6 data products, as well as some associated information (radar site, lat lon, date time).

```
my_pvol$radar
```

```
## [1] "KINX"
```

```
# KHGX

my_pvol$datetime
```

```
## [1] "2023-10-04 23:01:13 UTC"
```

```
# May 4 2017

length(my_pvol$scans)
```

```
## [1] 14
```

```
# 14 scans, ie 14 elevation angles

my_pvol$scans[1]
```

```
## [[1]]
##                   Polar scan (class scan)
##
##      parameters:  DBZH RHOHV WRADH PHIDP ZDR VRADH
## elevation angle:  0.483395 deg
##            dims:  1201 bins x 720 rays
```

```
# this array has dimensions 1201x720. It's 0.5 degree azimuth 'rays' across a full rotation of 360 degr

# looking at individual arrays for each parameter
str(my_scan$params$DBZH)
```

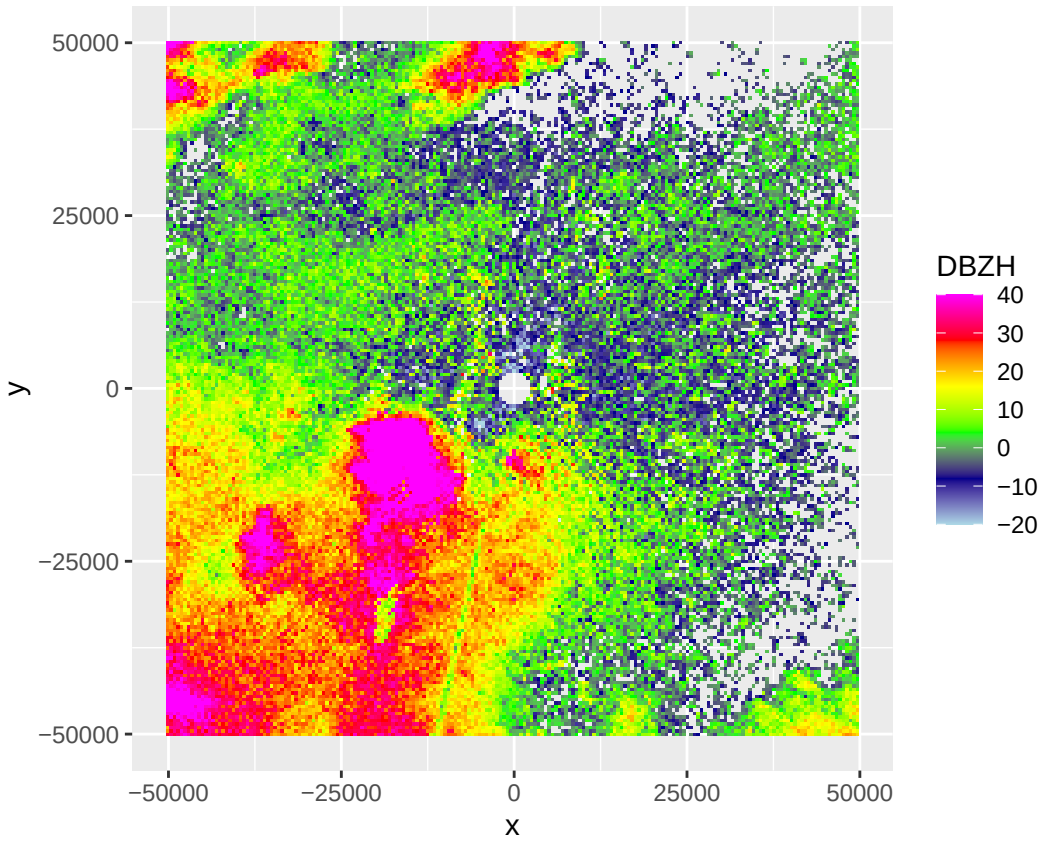
```
##  'param' num [1:1201, 1:720] NA NA NA NA NA NA NA NA NA -23 ...
##  - attr(*, "radar")= chr "KINX"
##  - attr(*, "datetime")= POSIXct[1:1], format: "2023-10-04 23:01:13"
##  - attr(*, "geo")=List of 7
##   ..$ lat    : num 36.2
##   ..$ lon    : num -95.6
##   ..$ height : num 204
##   ..$ elangle: num 0.483
##   ..$ rscale : num 250
##   ..$ ascale : num 0.5
##   ..$ rstart : num 0
##  - attr(*, "param")= chr "DBZH"
##  - attr(*, "conversion")=List of 5
##   ..$ gain    : num 1
##   ..$ offset  : num 0
##   ..$ nodata  : num -1000
##   ..$ undetect: num -999
##   ..$ dtype   : chr "H5T_IEEE_F64LE"
```

Preprocessing

We can filter out precipitation using a basic method with a cutoff for specific data products. We can also use a more sophisticated algorithm such as mistnet (Lin et al 2019, Methods in Ecology & Evolution). Here is an example classifying weather and removing it.

```
setwd("/Users/elsketielens/postdoc/HiRAD/WSR/raw_scans/NEXRAD/")

# Here's a raw data array (polar volume) with biological and non-biological scatter scatter
plot(my_ppi,  param = "DBZH", zlim = c(-20, 40))
```



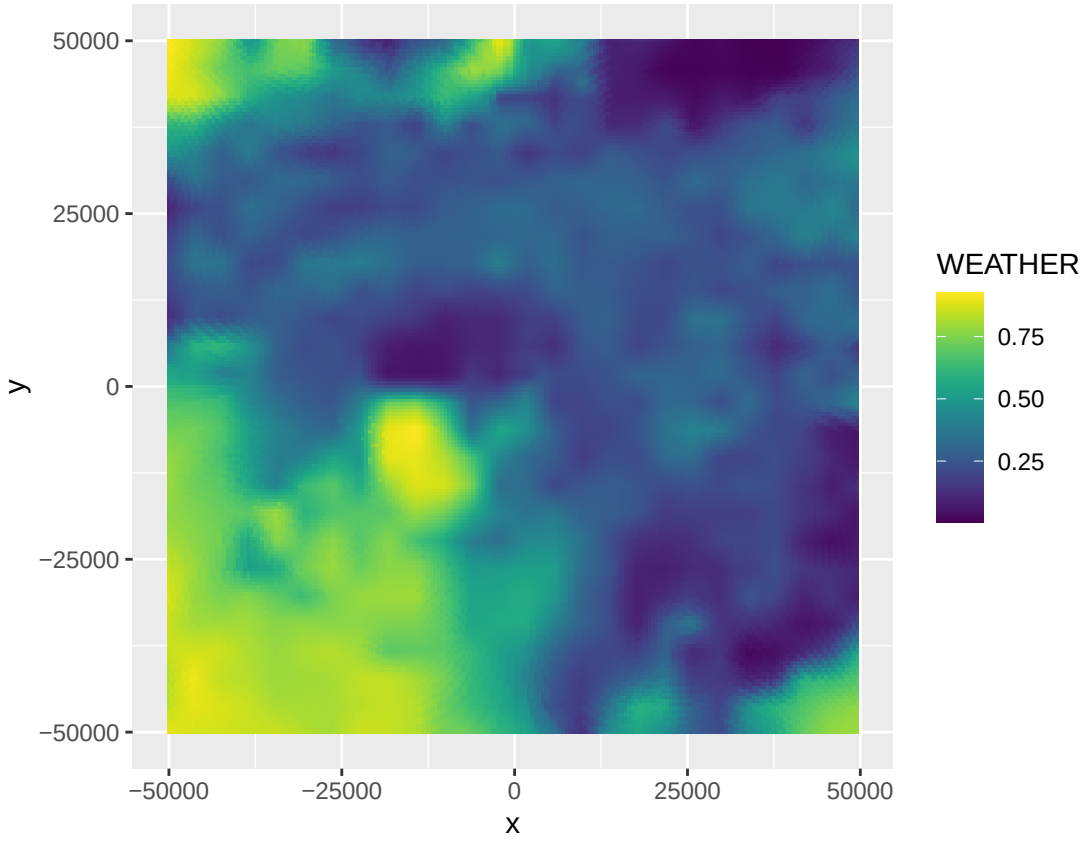

```
# We can use the mistnet algorithm to classify and remove weather
# apply the MistNet model to the polar volume file and load it as a polar volume (pvol):
my_pvol <- apply_mistnet(my_pvolfiles[1])
```

```
## Filename = 2023/10/04/KINX/KINX20231004_230113_V06, callid = KINX
## Removed 1 SAILS sweep.
## Call RSL_keep_sails() before RSL_anyformat_to_radar() to keep SAILS sweeps.
## Reading RSL polar volume with nominal time 20231004-230113, source: RAD:KINX,PLC:TULSA,state:OK,radar
## Warning: dimensions of scan parameter 1 at elev 3.999025 do not match scan dimensions, resampling ..
## Warning: dimensions of scan parameter 2 at elev 3.999025 do not match scan dimensions, resampling ..
## Warning: dimensions of scan parameter 5 at elev 3.999025 do not match scan dimensions, resampling ..
## Warning: dimensions of scan parameter 9 at elev 3.999025 do not match scan dimensions, resampling ..
## Warning: dimensions of scan parameter 10 at elev 3.999025 do not match scan dimensions, resampling .
## Running vol2birdSetUp
## Warning: radial velocities will be dealiased...
## Warning: disabling single-polarization precipitation filter for S-band data, continuing in DUAL polar
## Warning: using MistNet, disabling other segmentation methods
## Running segmentScansUsingMistnet.
## Warning: Requested elevation scan at 1.500000 degrees but selected scan at 1.318360 degrees
## Warning: Requested elevation scan at 3.500000 degrees but selected scan at 3.120115 degrees
## Warning: Requested elevation scan at 4.500000 degrees but selected scan at 3.999025 degrees
## Warning: Ignoring scan(s) not used as MistNet input: 2 4 8 9 10 11 12 13 14 ...
## Running MistNet...done
```

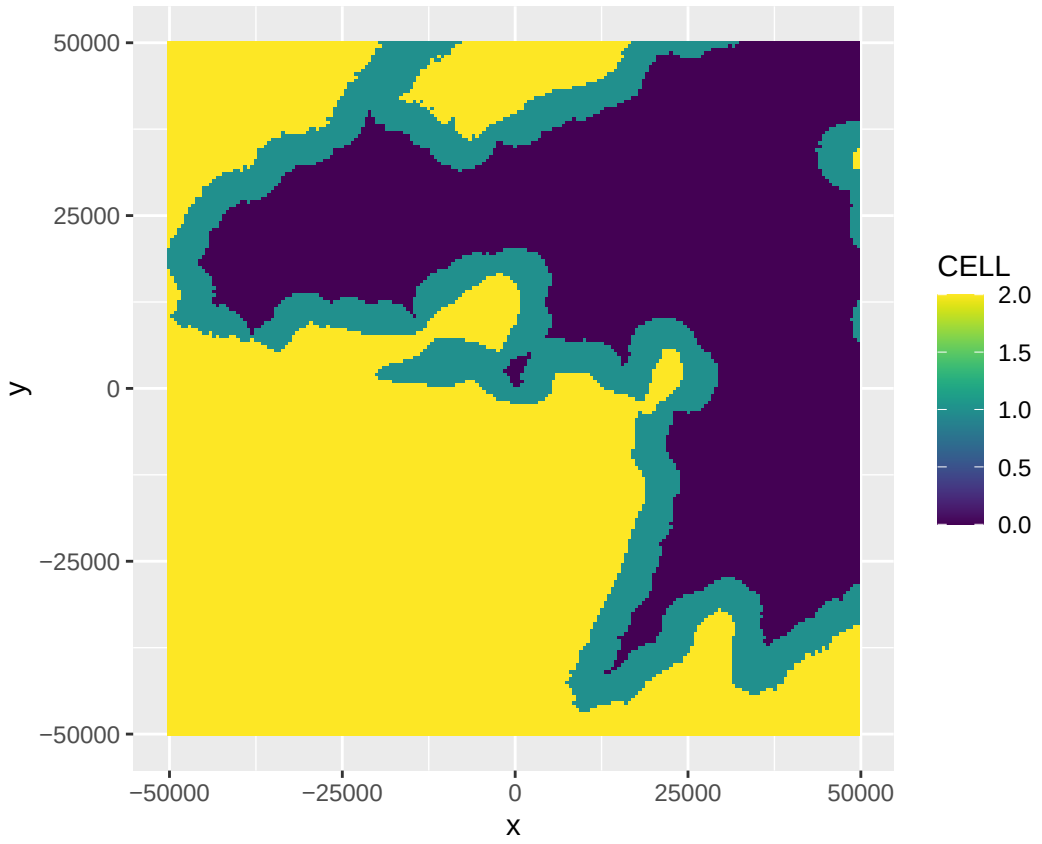
```
my_scan <- get_scan(my_pvol, 0.5)
# plot some summary info about the scan to the console:
my_scan
```

```
##                   Polar scan (class scan)
##
##      parameters:  DBZH VRADH RHOHV WRADH WEATHER BACKGROUND CELL PHIDP BIOLOGY ZDR
## elevation angle:  0.483395 deg
##            dims:  1201 bins x 720 rays
```

```r
# project the scan as ppi:
my_ppi <- project_as_ppi(my_scan)
# plot the probability for the WEATHER class
plot(my_ppi, param = 'WEATHER')
```



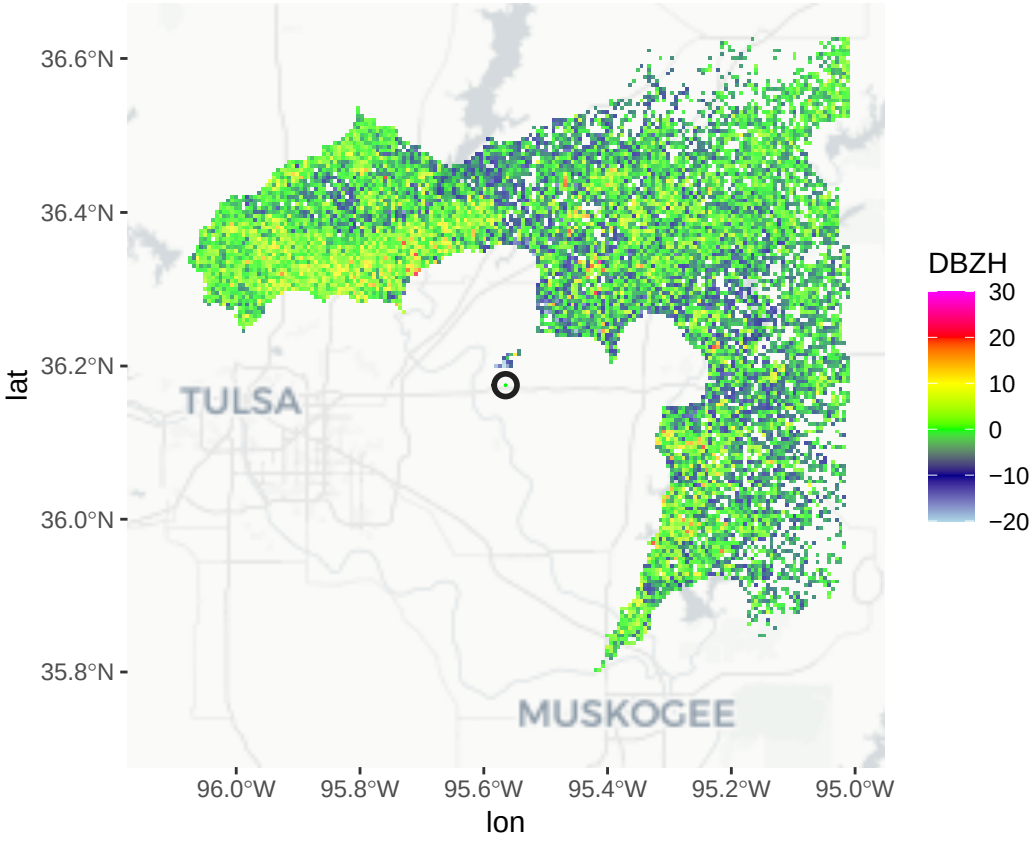

```r
plot(my_ppi, param = 'CELL')
```




```
# let's remove the identified precipitation area (and additional border) from the ppi, and plot it:
my_ppi_clean <- calculate_param(my_ppi, DBZH = ifelse(CELL >= 1, NA, DBZH))
map(my_ppi_clean, map="cartolight", param = 'DBZH')
```

```
## Warning in CPL_crs_from_input(x): GDAL Message 1: +init=epsg:XXXX syntax is
## deprecated. It might return a CRS with a non-EPSG compliant axis order.
```

```
## Zoom: 8
```



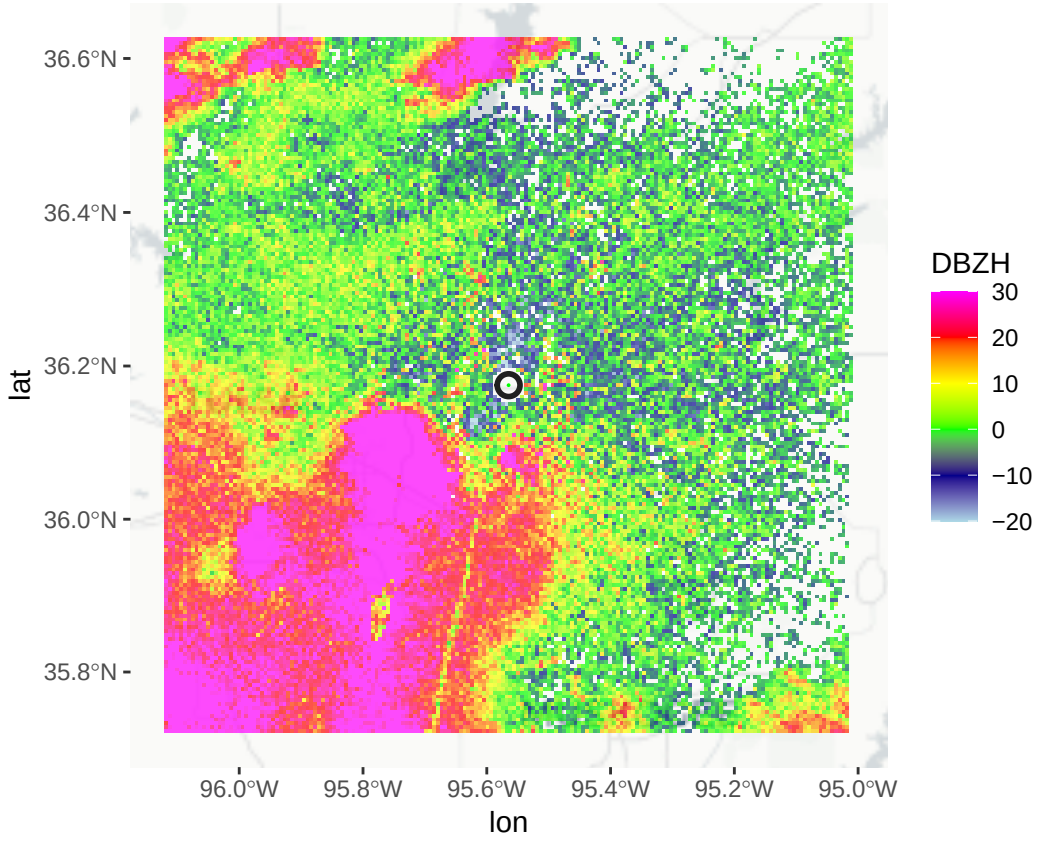

```
# also generate arrays with only clean data for other radar products
my_ppi_clean <- calculate_param(my_ppi, RHOHV = ifelse(CELL >= 1, NA, RHOHV))
my_ppi_clean <- calculate_param(my_ppi, VRADH = ifelse(CELL >= 1, NA, VRADH))
```

Visualize data products

Then we can plot various radar products that give information on attributes of the targets.

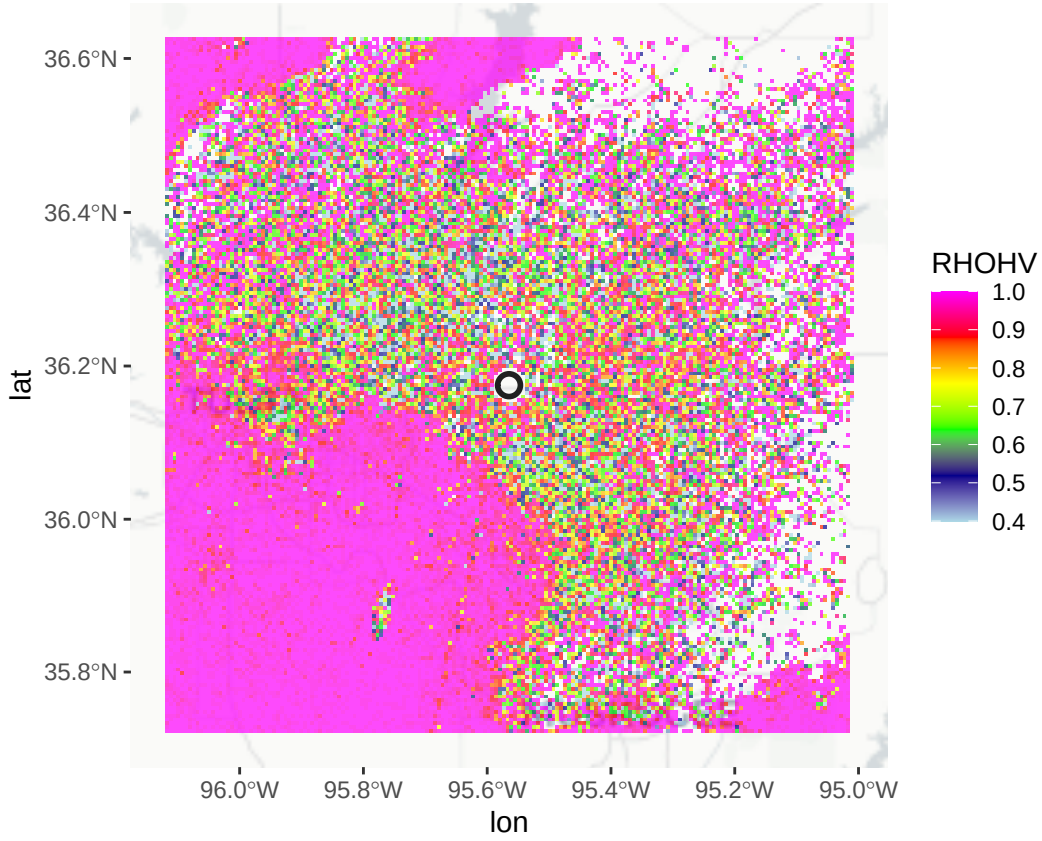
```
# Reflectivity
map(my_ppi_clean, map="cartolight", param = 'DBZH')
```

```
## Zoom: 8
```



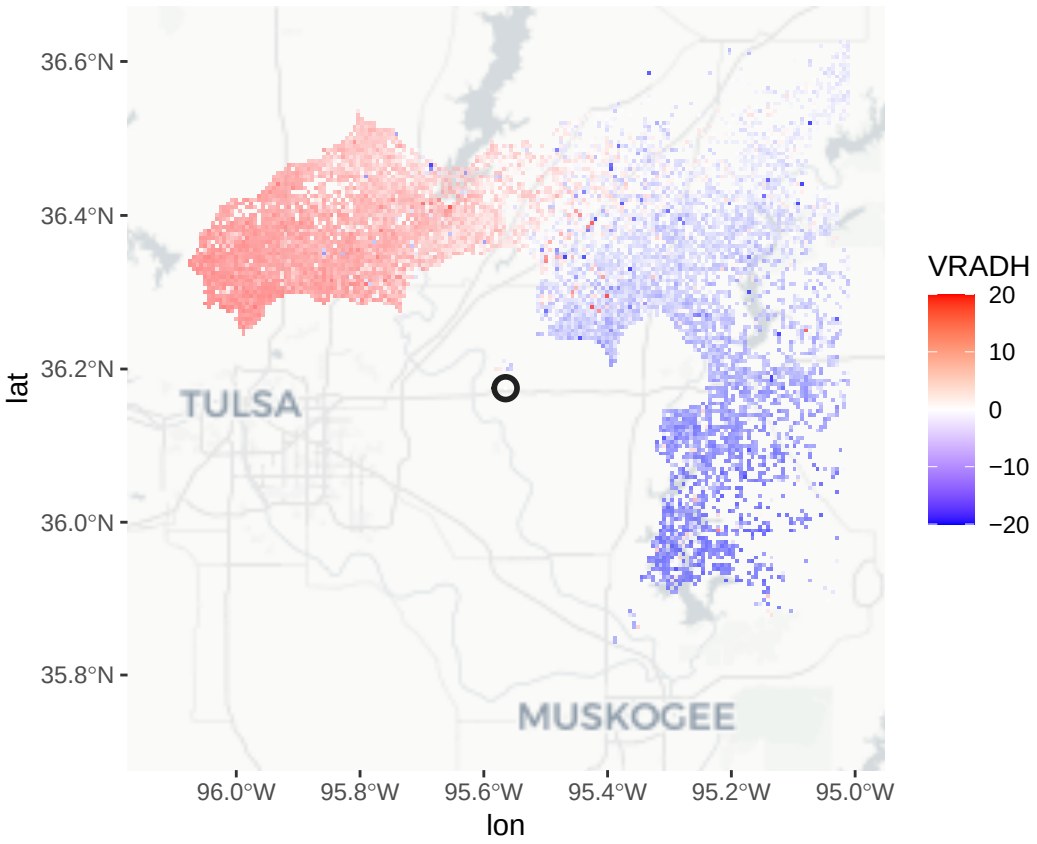

```
# Correlation coefficient
map(my_ppi_clean, map="cartolight", param = 'RHOHV')
```

```
## Zoom: 8
```




```
# Radial velocity
map(my_ppi_clean, map="cartolight", param = 'VRADH')
```

```
## Zoom: 8
```



day with lots of bird migration regular day insect activity insect migrating in front

For our animal classification algorithm, we propose to use data from multiple years across 3 countries (NL, BE, DE).