`robspack`, a fast R package to read, process and plot NOAA/GML ObsPack

University of Colorado Boulder

CIRES

Sergio Ibarra-Espinosa[1,2*], Lei Hu[2*]

1. University of Colorado-Boulder; 2. NOAA Global Monitoring Laboratory

*Correspondence to: sergio.ibarra-espinosa@noaa.gov

NOAA — Global Monitoring Laboratory, Earth System Research Laboratories

Motivation and goals

- Methane is a greenhouse gas (GHG) that affects the Earth's radiative forcing and consequently, the climate system.
- The NOAA Global Monitoring Laboratory (GML) manages a global network of atmospheric observations compiled and delivered to the public as ObsPack CH_4 GLOBALVIEW+.
- We developed the R package **`robspack`**, which reads and processes NOAA/GML ObsPack

Ok, but why?

We are studying the impact of COVID19 on CH_4 posterior emissions. We use atmospheric inversions to estimate posterior methane emissions (e.g. Hu et al., 2017; 2019).

$$s = s_p + (HQ)^T * (HQH^T + R)^{-1} * (z - Hs_p)$$

- ***z*** vector of measurements
- ***H*** matrix that describes the relationship between measurements and the unknown fields (*footprints*)
- ***R*** is the covariance matrix of the model-data mismatch errors
- s_p is the prior estimate of ***s***
- ***Q*** is a prior error covariance matrix of emissions

`robspack` provides ***z*** and inputs for ***H*** (not only *flask*, also insitu!)

System Requirements

- **`robspack`** imports data.table (Dowle & Srinivasan, 2021), which is basically C + OpenMP (super fast! check https://h2oai.github.io/db-benchmark/)
- Also imports lubridate (Grolemund, 2011) , cptcity (Ibarra-Espinosa 2018), and other base packages (R Core Team 2022).
- **`robspack`** can be installed in any OS (there is no need of conda 🐍)

Application for tower-insitu

Installation and summary of the data

```
remotes::install_github("ibarraespinosa/robspack")
obs <- "obspack_ch4_1_GLOBALVIEWplus/data/txt"
index <- obs_summary(obs = obs)
```

```
Number of files of index: 362
             sector    N
1:     aircraft-pfp   40
2:  aircraft-insitu   11
3:            flask  101
4:   surface-insitu  124
5:          aircore    1
6:      surface-pfp   33
7:     tower-insitu   51
8: shipboard-insitu    1
9:    Total sectors  362
Detected 136 files with agl
Detected 226 files without agl
```

1) The object index is a table with path for each ObsPack file, including altitude AGL and sector.
2) Read and add metadata information as columns. The object df contains all observations, including laboratory and respective scales. Then we can filter by space and time.

```
df <- obs_read(index = index, categories = "tower-insitu")
df[altitude_final < 10 & longitude > -170 & longitude < -50 &
   latitude > 10 & latitude < 80 & year ==% 2020]
```

3) Time. convert from seconds to time POSIXct, identifying intervals and ending and starting times.

```
df2 <- obs_addtime(df)
```

4) Plot, where we can control groups to plot. For site "LEF", higher observations are found closer to surface on Fig 1, n = 104870

```
obs_plot(dt = df2[site_code== "LEF"], time = "timeUTC", yfactor = 1e9,
cex = 0.5,ylab = expression(CH[4]~ppb), colu = "altitude_final",
pal = "mpl_viridis", n = c(122, 30, 396))
```

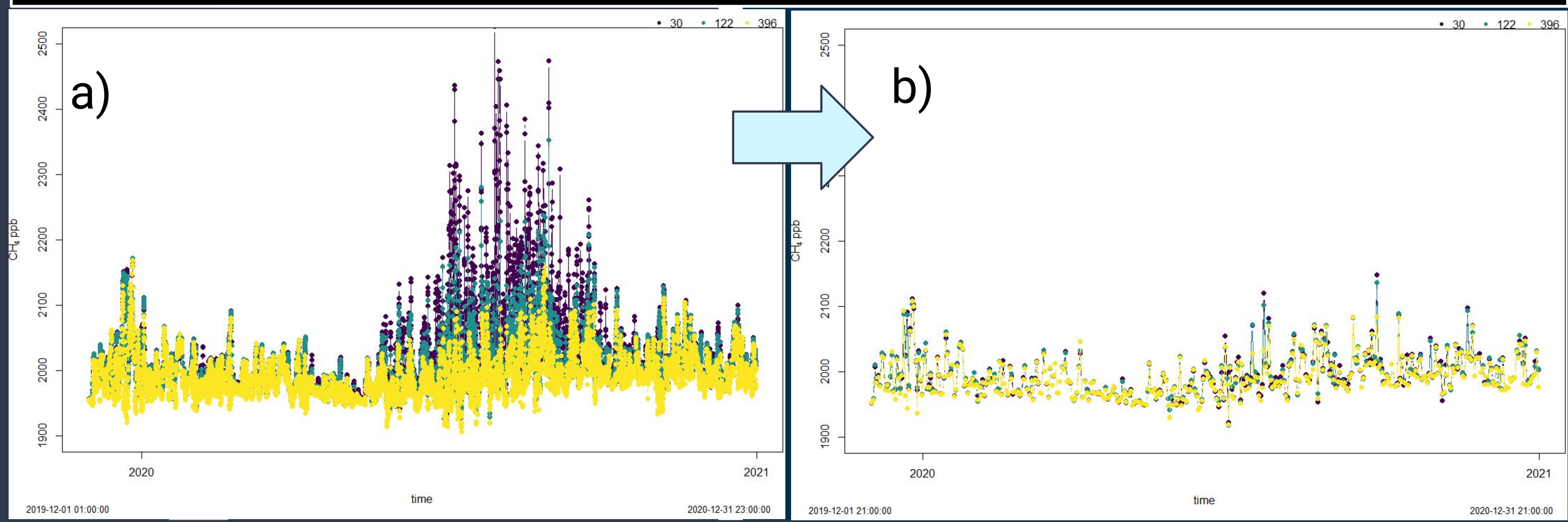

Figure 1. a) Observations of CH_4 for the site LEF by altitude and b) the same but filtered.

5) First we cut UTC time every 2 hours. We are interested in daily data 2-3pm local time (LT). LT is obtained via metadata, or calculated using the following formula. Then we select local hours 2-3pm.

$$LT = UTC + longitude/15*60*60$$

6) Now, as there are many observations with the same time with different values. Then, we need to aggregate the data by time, lab, scale, site code and altitude. Then, the new plot on 1b) with 4394 observation has same pattern as Fig 1a), with 104870.

```
df2 <- obs_addtime(df)
df2$timeUTC <- cut(x = df2$timeUTC + 3600,
                   breaks = "2 hour")
df3 <- obs_addltime(df2) ; df3 <- df3[lh %in% 14:15]
df4 <- obs_agg(dt = df3, gby = "mean", cols =
c("value","latitude", "longitude", "type_altitude",
"dif_time", "site_utc2lst"), byalt = TRUE)
df5[, max_altitude := max(altitude_final, na.rm = TRUE),
    by = site_code]
```

7) We need to identify the max altitude by site and select the highest

8) The data can be save as a master file, with all information, and the receptor, which the list of receptors. Saving the master is done including a YAML header.

```
obs_write(dt = df5,
      notes = "tower 2020",
      out = "tower_2020.csvy")
```

9) Saving the receptor data is similar, but the format is .txt and it is done by sector and type of altitude, AGL or ASL.

```
re<-df5[altitude_final == max_altitude,
c("site_code", "year", "month", "day",
"hour", "minute", "second", "latitude",
"longitude","altitude_final",
"type_altitude","year_end", "month_end",
"day_end", "hour_end", "minute_end",
"second_end")]
re[, af:= round(receptor$altitude_final)
re <- obs_format(re)
```

10) At this point HYSPLIT is run following Hu et al (2023) to generate footprints. The last step check the the NetCDf footprints and write the inputs to run the inverse model. More details in Ibarra-Espinosa and Hu (2023)

```
tf <- obs_invfiles(master = df5,
path = "path", Type = "continuous",
SubType = "ground",
Surface_Elev = master$max_altitude)
```

References

Hu, L., etm al: Declining, seasonal-varying emissions of sulfur hexafluoride from the United States, Atmos. Chem. Phys., 23, 1437–1448, https://doi.org/10.5194/acp-23-1437-2023, 2023.

Hu, L., et al. 2019. Enhanced North American carbon uptake associated with El Niño. Science advances, 5(6), p.eaaw0076.

Hu et al, 2017. 2017. Considerable contribution of the Montreal Protocol to declining greenhouse gas emissions from the United States. Geophysical Research Letters, 44(15), pp.8075-8083.

Ibarra-Espinosa, S. and Hu, L. (2023) robspack, an R package to process NOAA GML CH_4 Obspack GLOBALView+. To be submitted to JOSS https://ibarraespinosa.github.io/robspack/ .

Garrett Grolemund, Hadley Wickham (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. URL https://www.jstatsoft.org/v40/i03/.

Sergio Ibarra-Espinosa (2018) cptcity: incorporating the cpt-city archive into R. R package version 1.0.0. https://CRAN.R-project.org/package=cptcity

R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Acknowledgement

Funding: This project is funded by the NOAA Climate Program Office AC4 and COM programs (NA21OAR4310233 / NA21OAR4310234). This research was supported by NOAA cooperative agreement NA22OAR4320151. Also, thanks to John Miller NOAA GML, Kenneth Schuldt NOAA GML, Kirk Thoning NOAA GML