GeoPressureR **CHEATSHEET** [Part 1]

Workflow

STEP 1

Prepare tag object

Store the raw logger data and aggregate all the information needed to ultimately model the bird's trajectory.

`tag_create(id)` crop_start, crop_end

`tag_label(<tag>)`

- `tag_label_write(<tag>)`
- `tag_label_read(<tag>)`
- `tag_label_stap(<tag>)`

`tag_set_map(<tag>,extent)` scale, known

STEP 2

Build likelihood maps

Determine the position of the bird based on pressure data by matching pressure timeseries of each stationary period with ERA-5 data.

`geopressure_map_(<tag>)`

`geopressure_map_mismatch(<tag>)` max_sample, margin, thr_mask,

`geopressure_map_likelihood(<tag>)` sd, log_linear_pooling_weight

STEP 3 [optional]

Build likelihood maps with light data

If available, include light data in your analysis to increase computational efficiency when creating your graph.

`twilight_create(<tag>)` twl_thr, twl_offset

`twilight_label_write(<tag>)`

`twilight_label_read(<tag>)`

`geolight_map(<tag>)` twl_calib_adjust, twl_llp

STEP 4

Create graph

Create a trellis graph representing the possible trajectory of the bird with a Hidden Markov Model.

`graph_create(<graph>)` thr_likelihood, thr_gs

`tag_download_wind(<tag>)` (only wind)

`graph_add_wind(<graph>)` (only wind)

`graph_set_movement(<graph>)`

If wind: power2prob, bird, low_speed_fix
Otherwise: shape, scale,low_speed_fix

STEP 5

Produce trajectory outputs

Combine the observation model of pressure with the movement model of flight to build various trajectory outputs.

`graph_marginal(<graph>)`

`graph_most_likely(<graph>)`

`graph_simulation(<graph>)` nj

*all the parameters in brackets are mandatory

Variables & outputs

<tag>

pressure: date | value | label | stap_id

acceleration: date | value | label | stap_id

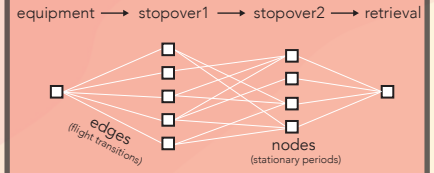
light: date | value | stap_id

stap: stap_id | start | end | known_lat | known_lon | include

map_pressure_mse + map_pressure_mask → map_pressure

twilight: twilight | rise | stap_id | label → map_light

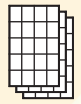
<graph>



A `<graph>` is essentially a list of the source `s` and target `t` nodes of all edges.

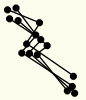
The index of nodes is based on the 3D matrix (lat-lon-stap) of size `sz`.

In addition, `gs` and `ws` store the groundspeed and windspeed of each edge.



Marginal maps
Probability map of positions accounting for all observations and transitions

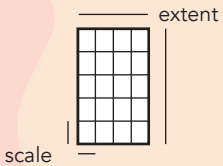
Most likely path
Set of positions for each stationary period which maximizes the joint probability of the trajectory



Simulated paths
Multiple independent paths, on which you can compute metrics and summary statistics

<map>

A GeoPressureR `<map>` object contains discretized spatio-temporal data according to scale and extent (space), and long stationary periods (time).

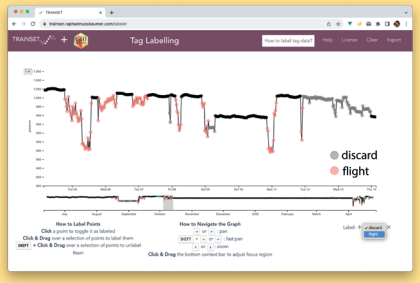


LABELLING TRACKS

Labelling your timeseries is an iterative process that involves

(1) identifying flights to define stationary periods and flight duration, and

(2) discarding vertical altitudinal movements of the bird and outliers.



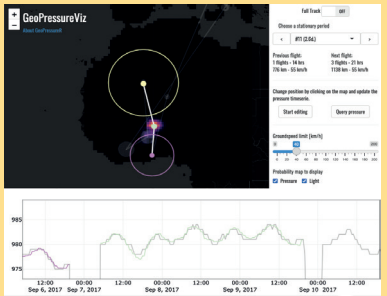
https://trainset.raphaelnussbaumer.com/

3 Labelling tools

1. GEOPRESSUREVIZ

Use this Shiny app to visualize the overall trajectory of the bird as well as each step-by-step move, or share the trajectory with collaborators.

`geopressureviz(id|file|<tag>)`



2. PRESSUREPATH

Create a dataframe to directly compare the actual pressure measured by the sensor to the ERA-5 reanalysis data along the best estimate of the path. You can also compute altitude throughout the bird's trajectory.

`pressurepath_create(<tag>)`

3. UPDATE

If you chang the lable on a few stationary periods, use theses functions to only re-compute these stationary periods.

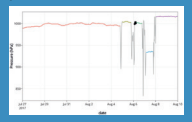
`tag_update(<tag>)`
`pressurepath_update(pressurepath)`

GeoPressureR **CHEATSHEET** [Part 2]

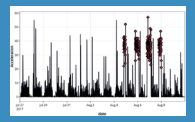
`plot(<tag>, type)`

Display the data `type` contained in `<tag>` as timeseries or a map.

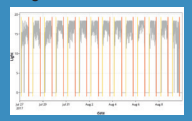
pressure



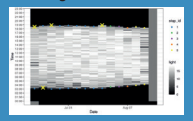
acceleration



light



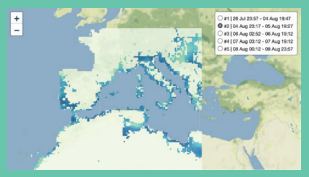
twilight



`map_pressure_mask`, `map_pressure_mse`, `map_pressure`

`plot(<map>, path)`

Display a `<map>` with, optionally, a `path` on top.

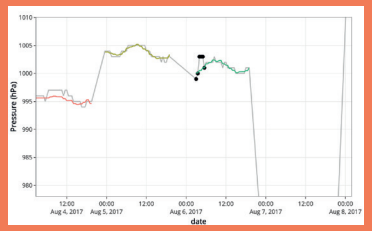


`plot_path(path)`

Plot a path data.frame.

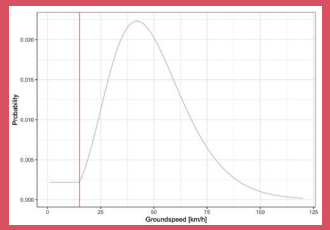
`plot_pressurepath(pressurepath)`

Display a pressurepath data.frame as a timeseries or a histogram.



`plot_graph_movement(<graph>)`

Display the movement model.



GEOPRESSURE TEMPLATE

A standardized project folder structure to store your data.

```
GeoPressureTemplate/
├── DESCRIPTION
├── README.md
├── GeoPressureTemplate.Rproj
├── LICENCES.md
├── config.yml
├── data/
│   ├── raw_tag/
│   │   ├── 18LX/
│   │   │   ├── 18LX_20180725.acceleration
│   │   │   ├── 18LX_20180725.glf
│   │   │   └── 18LX_20180725.pressure
│   │   └── CB619/
│   │       └── CB619.deg
│   ├── tag_label/
│   │   ├── 18LX-labeled.csv
│   │   └── CB619-labeled.csv
│   ├── twilight_label/
│   │   └── 18LX-labeled.csv
│   ├── wind/
│   │   └── 18LX/
│   │       ├── 18LX_1.nc
│   │       └── ...
│   └── interim/
│       └── 18LX.RData
├── analysis/
│   ├── 1-label.qmd
│   ├── 2-twilight.qmd
│   ├── 3-wind.qmd
│   └── 4-geopressure.R
└── output/
    ├── create_figures.R
    └── figures/
        └── marginal.png
```

/Rafnuss/GeoPressureTemplate

UTILITIES

General utility functions of the GeoPressureR package

`tag2path(<tag>)` Create a path from the positions with the highest likelihood value.

`path2edge(path, <graph>)` Retrieve the edges and flight information of a path in a graph.

`stap2flight(stap)` Compute flights from stationary periods.

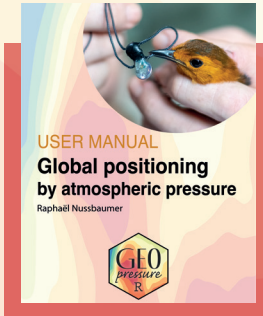
`stap2duration(stap|flight)` Compute the duration of stationary periods or flights.

`rast(<map>)` Construct a terra::SpatRaster from a map.

`geopressure_map_preprocess(<tag>)` Clean, smooth and downscale pressure data to match ERA-5 data.

`pressurepath2altitude(pressurepath)` Compute the timeseries of altitude from a pressurepath.

`geopressure_timeseries(lat, lon, pressure)` Download the pressure timeseries at a given location.



raphaelnussbaumer.com/GeoPressureManual

Learn how to use GeoPressureR with the the **GeoPressureManual**. Using the examples of a Swainson's Warbler and a Great Reed Warbler, this user guide will takes you through each step of the analysis in detail.

PARAMETERS

`<param>`

`<param>` contains all the essential function arguments used to create the likelihood maps and graph. It is nested within `<tag>` and `<graph>`.

This allows for reproducibility and examination of parameters post-creation.