

# User Guide: Visual Filter App

## Introduction

The Visual Filter App is designed to help users to filter and process animal tracking data from the ATLAS system (Advanced Tracking and Localization of Animals in real-life Systems).

## System Requirements

**Software:** R version 4.0 or higher (version 4.2.2. is the most recommended), RStudio (recommended)

**Access:** a working VPN connection to the ATLAS database server, and credentials to connect to the database.

### R Packages:

- roxygen2
- crayon
- lubridate
- DBI
- RMySQL
- RSQLite
- Rcpp
- dplyr
- leaflet
- sf
- RColorBrewer
- htmltools
- leaflet.extras

# Usage

## Instructions for the first run

### 1. Download the App Folder

1.1. Download the folder `Visual_Filter_App`

### 2. Create a new R project

2.1. Open RStudio

2.2. Go to File -> New Project -> Existing Directory

2.3. Browse the `Visual_Filter_App` folder and click 'Create Project'.

2.4. Now you should see the contents of the app's folder under the Files tab in RStudio

### 3. Install the required packages:

3.1. open and run the script:

```
install_required_R_packages_visual_filter.R
```

### 4. Set the Working Directory

4.1. Open the script `visual_filter_shiny_app.R`

4.2. Update the variable `path_to_visual_filter_folder` to the path where `Visual_Filter_App` is located.

4.3. Make sure to include "/" at the end of the path

### 5. Set the running configuration:

open the file `config_visual_filter.R` and modify the following sections as follows:

#### 5.1. Data to retrieve

5.1.1. Choose your animal's name code, e.g.:

5.1.1.1. Choose two English letters which represent the animal's name

5.1.1.2. Example: `animal_name_code <- "BO"` for Barn Owl

5.1.2. Insert the tag number and time range:

5.1.2.1. `tag_number <- 972006000556`

5.1.2.2. `start_time <- '2021-07-04 17:00:00'`

5.1.2.3. `end_time <- '2021-07-06 05:00:00'`

#### 5.2. Baseline Filter Settings:

5.2.1. set the values that apply to the animal you are working on

5.2.2. Please tag outliers in data of one animal and one tag at a time

### 5.3. Start and End times for the `AssignDayNumbers` function

5.3.1. The Visual Filter divides the data into day segments and assigns them DAY numbers (e.g. 1, 2, etc.)

5.3.2. The function which does that is called `AssignDayNumbers`

5.3.3. If you want to see data **only within a specific time range** each day (e.g., 05:00–18:00), set:

5.3.3.1. `day_start_time <- "05:00:00"`

5.3.3.2. `day_end_time <- "17:00:00"`

5.3.4. Otherwise, leave as **full-day settings**:

5.3.4.1. `day_start_time <- "00:00:00"`

5.3.4.2. `day_end_time <- "00:00:00"`

### 5.4. File Paths

5.4.1. Create two folders:

5.4.1.1. One for the **raw ATLAS data (input)**

5.4.1.2. One for the **filtered data (output)**

5.4.2. Set the folder paths in the config file:

5.4.2.1. `raw_data_path <- "path/to/raw_data_folder/"`

5.4.2.2. `filtered_data_path <- "path/to/filtered_data_folder/"`

5.4.2.3. Make sure to include `'/'` at the end of the paths

### 5.5. CSV Settings

5.5.1. The **manual tagging** process requires raw data from the ATLAS server.

Keep this setting:

5.5.1.1. `upload_gps_data_from_csv <- FALSE`

5.5.2. To **upload data later from CSV files for other uses**, set:

5.5.2.1. `upload_gps_data_from_csv <- TRUE`

5.5.3. To **save filtered data as CSV** in addition to the default SQLite storage:

5.5.3.1. `save_filtered_data_as_csv <- TRUE`

### 5.6. Background Map Setting- choose between:

5.6.1. `'Esri.WorldImagery'`

5.6.2. `'CartoDB.Positron'`

### 5.7. Credentials of the Harod ATLAS database

5.7.1. In case the credentials change later, update them here

5.7.2. **Do not share these credentials** with unauthorized users or store them in Git or other unsecured cloud platforms.

## 5.8. ATLAS time information

5.8.1. No changes needed

## 6. Connect to the TAU VPN server (GlobalProtect)

6.1. Ensure you are connected to the TAU VPN before running the app.

## 7. Run the app

7.1. Open `visual_filter_shiny_app.R` and click on 'Run App' in RStudio.

## 8. Tag outliers and save the data

8.1. Instructions in the next section

## 9. Upload and plot the filtered data

9.1. Open `upload_and_plot_filtered_data.R`

9.2. Set the file name you want to upload, for example:

```
file_name <- "B0_0556_from_2021-07-04_17-00-03_to_2021-07-04_23-59-58_filtered.sqlite"
```

## 10. Validate the outliers tagging

## 11. Submit the .sqlite files of the tagged data

## Instructions for runs with modified parameters

1. Modify parameters in `config_visual_filter.R`

To retrieve data for a different **animal**, **tag number**, or **time range**:

### 1.1. Set the Data to Retrieve:

Open `config_visual_filter.R` and update the following variables:

```
1.1.1. animal_name_code <- "B0"
1.1.2. tag_number <- 972006000556
1.1.3. start_time <- '2021-07-04 17:00:00'
1.1.4. end_time <- '2021-07-06 05:00:00'
```

### 1.2. Adjust the baseline filter thresholds (if needed)

```
1.2.1. Speed in m/s: speed_threshold_baseline_filter <- 20
1.2.2. Standard Deviation: std_threshold_baseline_filter <- 15
```

1.2.3. Number of participating Base Stations:

```
nbs_threshold_baseline_filter <- 3
```

**1.3. Modify the Start and End times for the `AssignDayNumbers` function (if needed)**

1.3.1. 

```
day_start_time <- "00:00:00"
```

1.3.2. 

```
day_end_time <- "00:00:00"
```

(If you want to filter data within a specific time window each day, change these values accordingly.)

**2. Run `visual_filter_shiny_app.R`**

**3. Tag outliers and save the data**

3.1. Instructions in the next section

**4. Upload and plot the filtered data**

4.1. Open `upload_and_plot_filtered_data.R`

4.2. Set the file name you want to upload, for example:

```
file_name <- "B0_0556_from_2021-07-04_17-00-03_to_2021-07-04_23-59-58_filtered.sqlite"
```

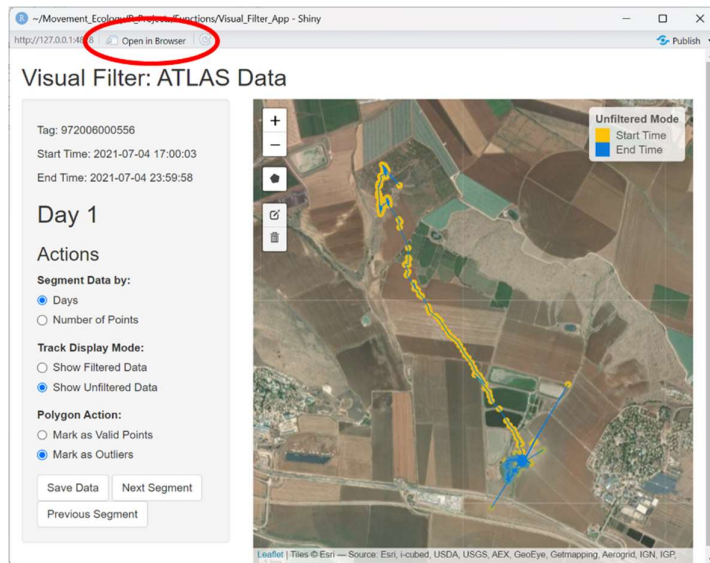
**5. Validate the outliers tagging**

**6. Submit the .sqlite files of the tagged data**

## App Features: Manual Tagging of Outliers

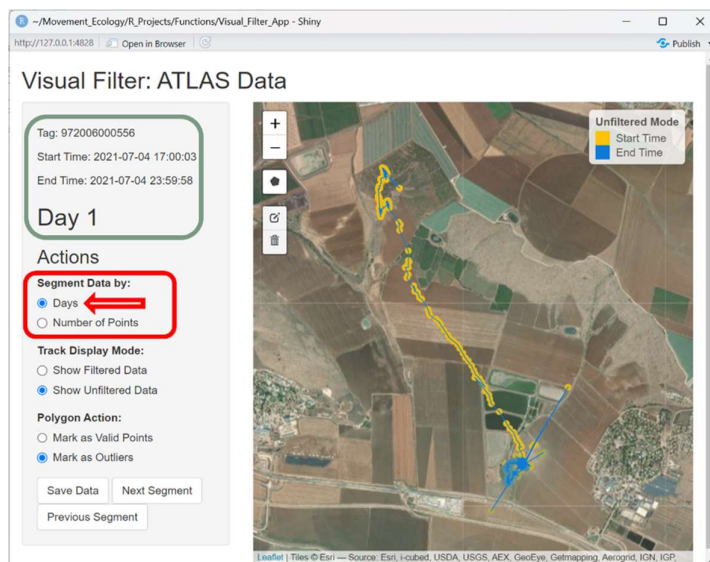
### Running the App in a Web Browser

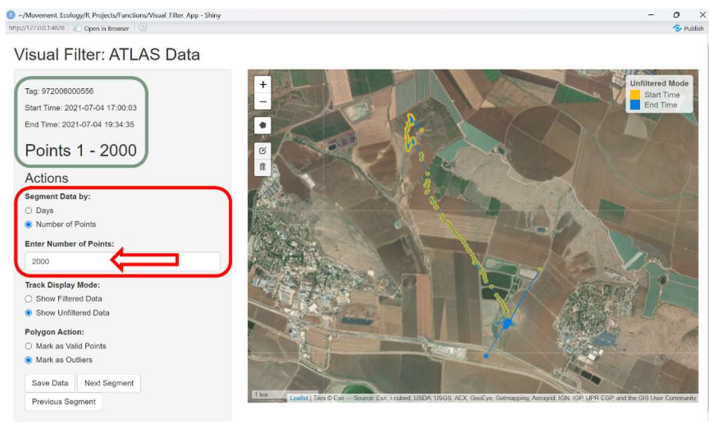
- Click "**Open in Browser**" to launch the app in your web browser:



## Data Display Options

- The data can be displayed in **daily segments** (as assigned by the `AssignDayNumbers` function) or in **custom-sized chunks** (set by the user):

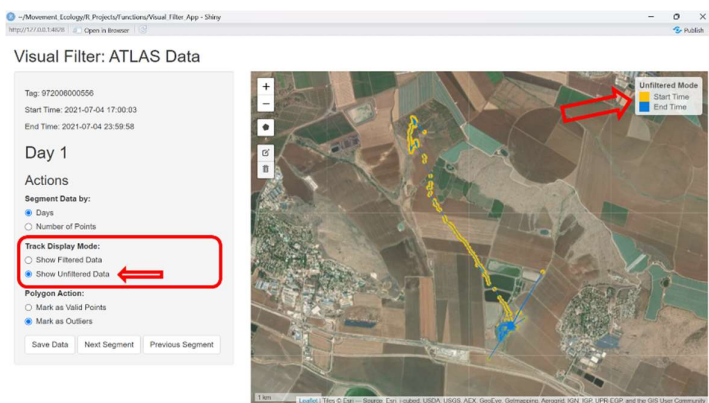




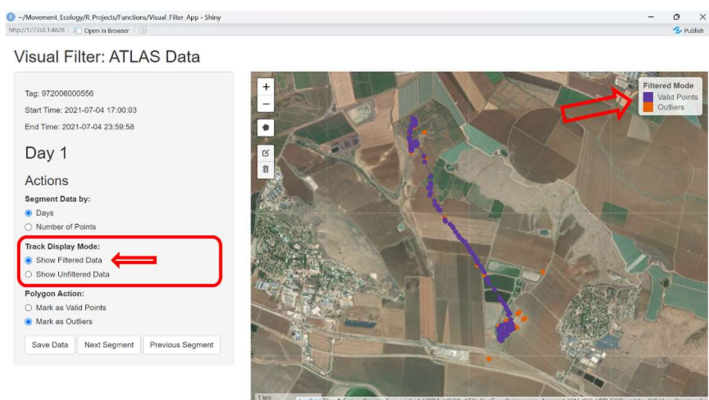
## Viewing the Track

- For each segment, you can choose between:

### 1. Unfiltered Track – Displays the entire track without filtering:



### 2. Filtered Track – Shows valid points and outliers based on the baseline filter thresholds set in the config file:

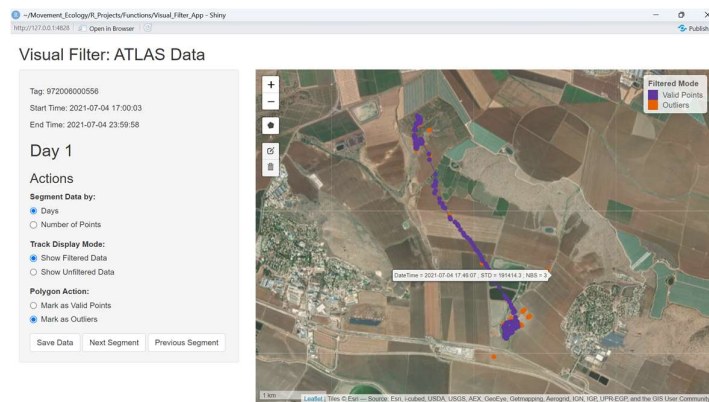


## Manual Tagging of Data

- To manually tag the data, ensure that you are in **"Filtered Data" Display Mode**.

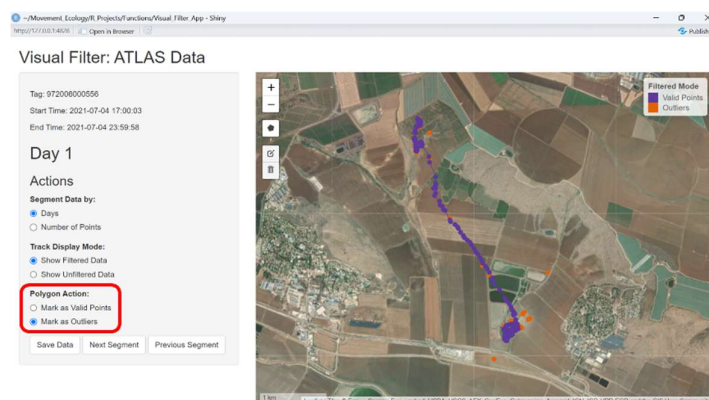
## Tagging Individual Points

- In **"Filtered Data" Display Mode**, you can manually **toggle** the classification of **single points** (valid vs. outlier) by clicking on them.
- By hovering over a point, a popup will appear, which includes the date, time, standard deviation (STD), and number of participating base stations (NBS):



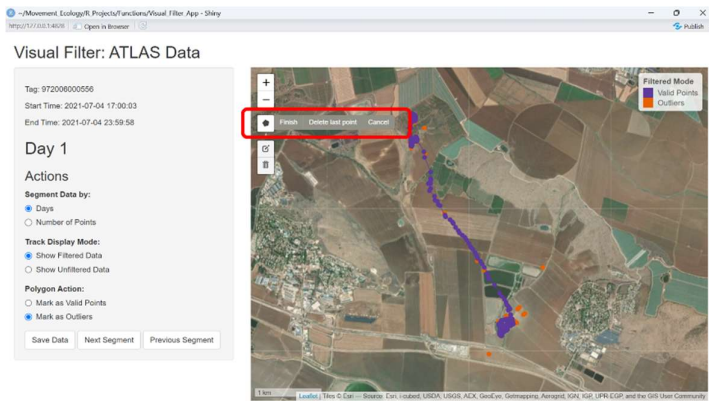
## Tagging by Polygon Selection

- You can mark multiple points using a **polygon selection tool**:
1. In the **"Polygon Action"** menu, choose whether to classify selected points as **Valid Points or Outliers**:

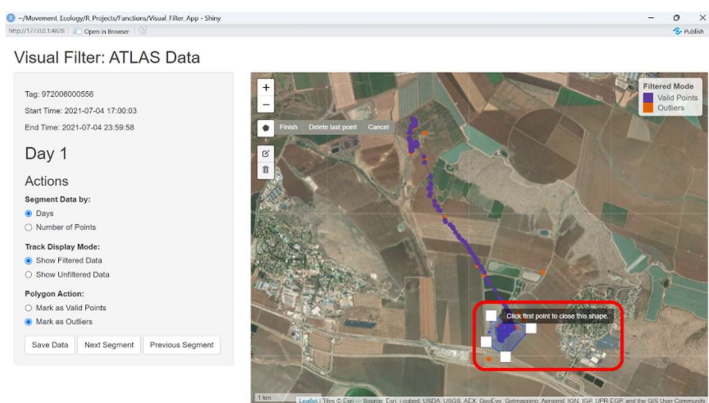


2. Click the **polygon button** on the map to activate polygon selection:



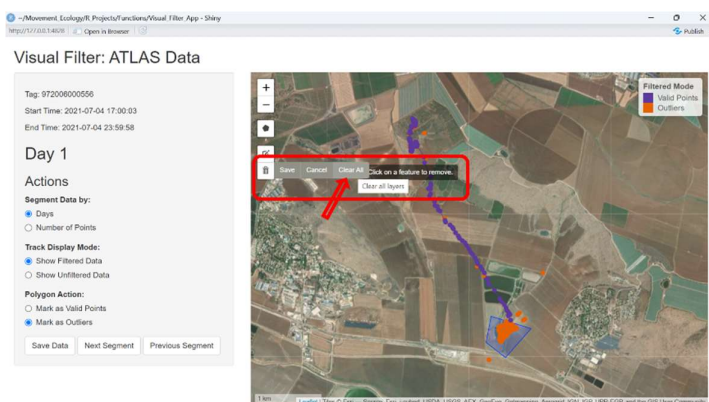


3. Click on the map to **place polygon vertices** and create the desired selection area:



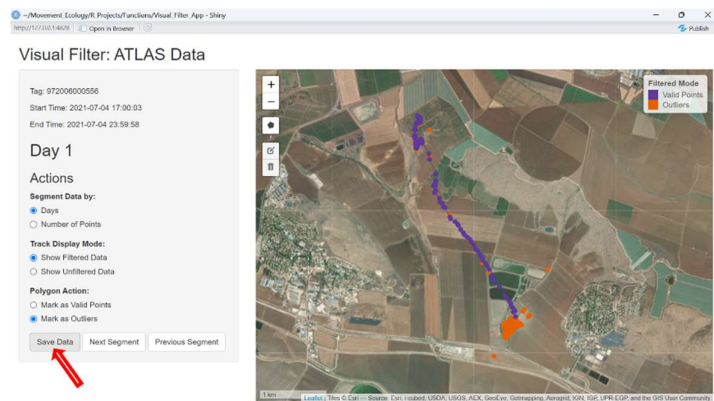
## Removing Marked Polygons

- To clear marked polygons, click "Delete Layers", then select "Clear All":

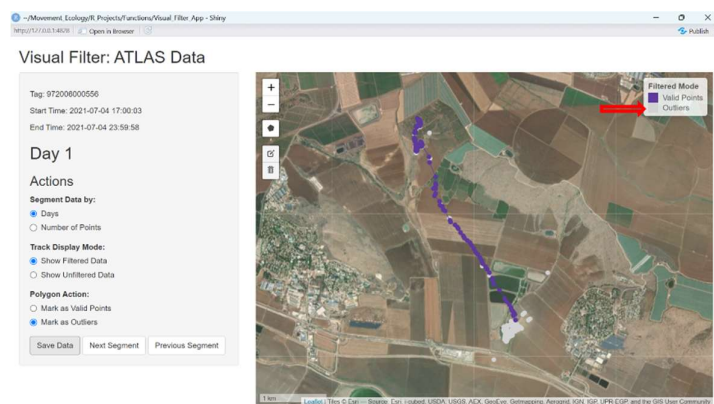


## Saving the tagged data

- To save the segment data when finished tagging, click "Save Data":

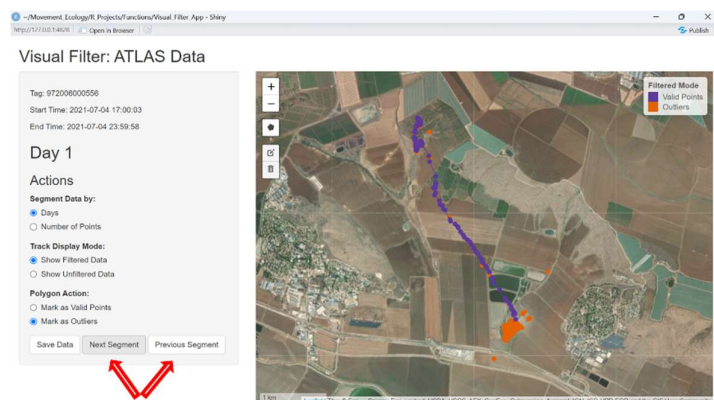


- The Outliers will change their colour. If needed, you can keep toggling points and save again:



## Navigating to the next or previous data segment

- To navigate to the next or previous segment, click **“Next Segment”** or **“Previous Segment”**:



- You will be asked if to save the data or not, even if you saved it already, in order to make sure the data is saved

## Troubleshooting

---

### Problem:

Error in .local(drv, ...) :

Failed to connect to database: Error: Can't connect to server on '132.67.132.47' (10060)

### Solution:

Make sure you are connected to the ATLAS database server

---

### Problem:

Warning: Error in <reactive:validate\_data\_for\_days>: No DAY numbers were found in the provided data.

### Solution:

Make sure that the tag number is correct and that there are data in the time range you chose.

---

### Problem:

Error: Could not connect to database:

unable to open database file

### Solution:

Make sure the paths you defined in the config file and the path of the working directory are correct

---

### Problem:

Warning: Error in wk\_handle.wk\_wkb: Loop 0 is not valid: Edge 2 crosses edge 5

### Solution:

Run the app again and draw a polygon with no self-intersecting edges

---

## Contact Information

For support, contact:

Dr. Neta Tsur

Email: [tsur.neta@gmail.com](mailto:tsur.neta@gmail.com)