Global Fishing Watch::gfwr

Open data and tools for a sustainable ocean

May 16th, 2023
CorrelAid

Tyler Clavelle, Senior Data Scientist

1. Intro to Global Fishing Watch
2. Overview of our core dataset
3. Intro to `gfwr`

Founded in 2015 as a collaboration between Google, Oceana, and SkyTruth

Established as an independent 501c3 international nonprofit in 2017

Largely grant funded by philanthropic foundations active in ocean conservation, with a small amount of government funding.

Global Fishing Watch

Vision

Healthy, productive and resilient oceans where transparent and effective governance of marine resources supports biodiversity and sustainable development.

Global Fishing Watch

Our team

75+ GFW Staff

20+ Countries

Mission

We advance ocean governance through increased transparency of human activity at sea.

By creating and publicly sharing map visualizations, data and analysis tools, we enable scientific research and drive a transformation in how we manage our ocean.

Global Fishing Watch

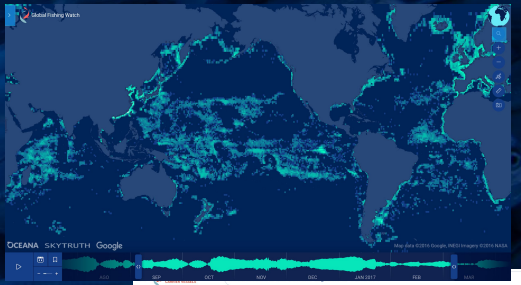
Our Solution :: Open Data

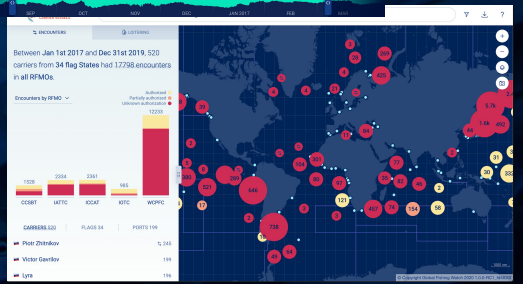
We share our data through different products:

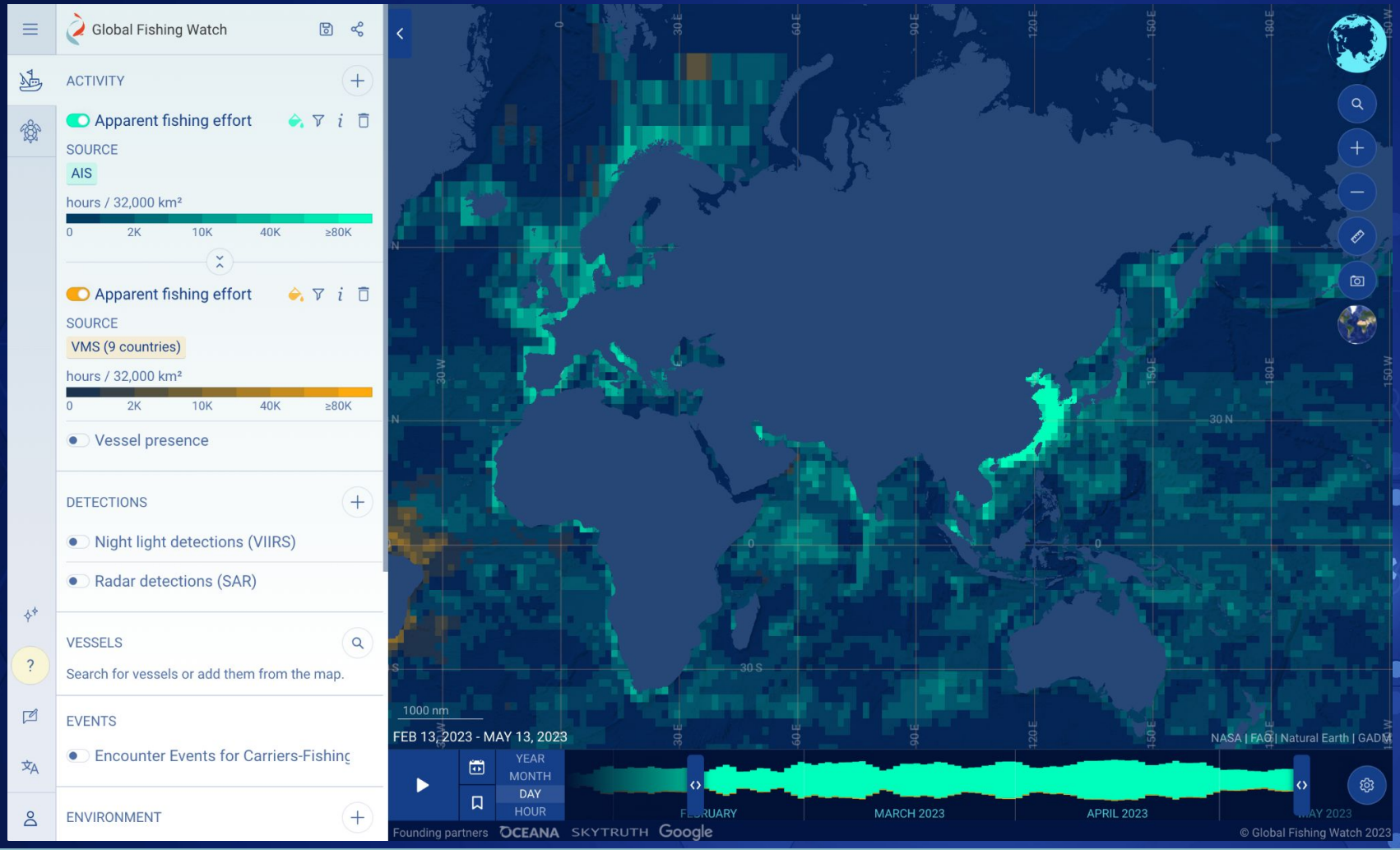
- **API Portal & gfwR**
- **Data Download Portal**
- **Map and Marine Manager Portal**
- **Carrier Vessel Portal**
- **Research and publications**

GFW Process

Idea or prototype → Research Projects → Products






Global Fishing Watch
ACTIVITY
Apparent fishing effort
SOURCE
AIS
hours / 32,000 km²
0
2K
10K
40K
≥80K
Apparent fishing effort
SOURCE
VMS (9 countries)
hours / 32,000 km²
0
2K
10K
40K
≥80K
Vessel presence
DETECTIONS
Night light detections (VIIRS)
Radar detections (SAR)
VESSELS
Search for vessels or add them from the map.
EVENTS
Encounter Events for Carriers-Fishing
ENVIRONMENT
1000 nm
FEB 13, 2023 - MAY 13, 2023
NASA | FAO | Natural Earth | GADM
YEAR
MONTH
DAY
HOUR
FEBRUARY
MARCH 2023
APRIL 2023
MAY 2023
Founding partners
OCEANA
SKYTRUTH
Google
© Global Fishing Watch 2023

Our tools

Historic data

Several years of data across all datasets: Fishing effort, encounters between vessels, night-light vessel detections, vessel events and identity.

Near real-time data

Data published with a 72-hour delay and access to historical vessel tracks and fishing activity back to January 1, 2012.

Easy to use

Designed so both experts and non-experts can see and assess fishing activity, and track individual vessels over time.

Accessible and free

Freely available to anyone with an internet connection; easy to access and use, developed to run on mid-level computers and with a low data consumption.

Global Fishing Watch

API Portal

Our API Portal provides a new way for partners and stakeholders to access dynamic datasets that are used in all our tools and incorporate them into directly into their own systems, platforms and research– *free for non-commercial use*.

By openly publishing our APIs, we aim to transform global collaboration and catalyze solutions to address the ocean's most complex problems.

Global Fishing Watch

Search

Introduction

Harvesting big data from s...

Data available

Quick Start

Get your GFW Postman col...

Map Visualization - 4Wings ...

Vessels API

Events API

Introduction

Welcome to the Global Fishing Watch Application Programming Interfaces (API) Documentation site!

Global Fishing Watch has created a technology portal that provides application programming interfaces, or APIs, to revolutionize the way we research and monitor human impacts across the world's ocean.

Our APIs provide software instructions that enable automatic connection of our data and products to other systems so users can easily download and integrate our datasets, code, and models to power their own platforms. By providing streamlined access to information on current fishing vessel activity, identity and history, we are helping create solutions for a resilient, sustainable ocean.

Global Fishing Watch APIs are only available for non-commercial purposes. They are used by researchers, governments and technology companies.

Open Data to Solve

Our API users represent more than 200 organizations from 67 countries, including 70 universities.

They are exploring new research in biodiversity and ecosystem resilience, fisheries management and technological innovation.

< Ecosystem & Biodiversity Resilience >

- “Identify patterns in fishing with climate change”
- “Reconstruct shark bycatch”
- “Identify marine debris occurring by fishing”
- “Understand how fishing redistributes following the implementation of MPAs”

< Fisheries Management >

- “Identify illegal, unreported and unregulated (IUU)”
- “Identify labor exploitation issues”
- “Cross check fishing grounds with official data submissions to tuna regional fisheries management organizations”

< Innovation >

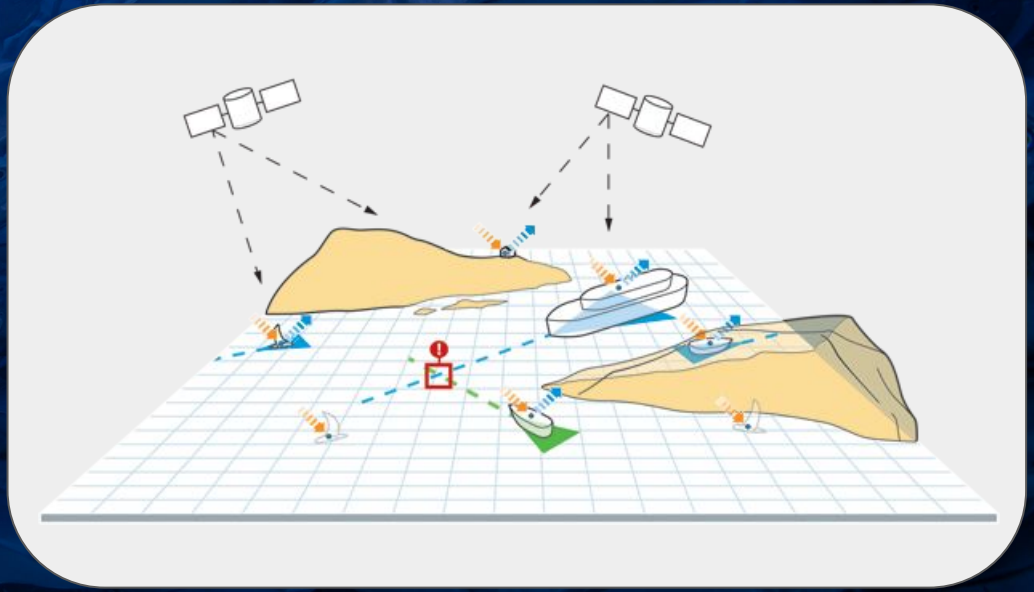
- “Support natural disaster planning”
- “Incorporate into university curriculum”
- “Teaching web development and data analytics bootcamp”

Automatic Identification System (AIS)

The **automatic identification system (AIS)** is a vessel broadcast technology originally designed for ship-to-ship communication as a collision avoidance system.

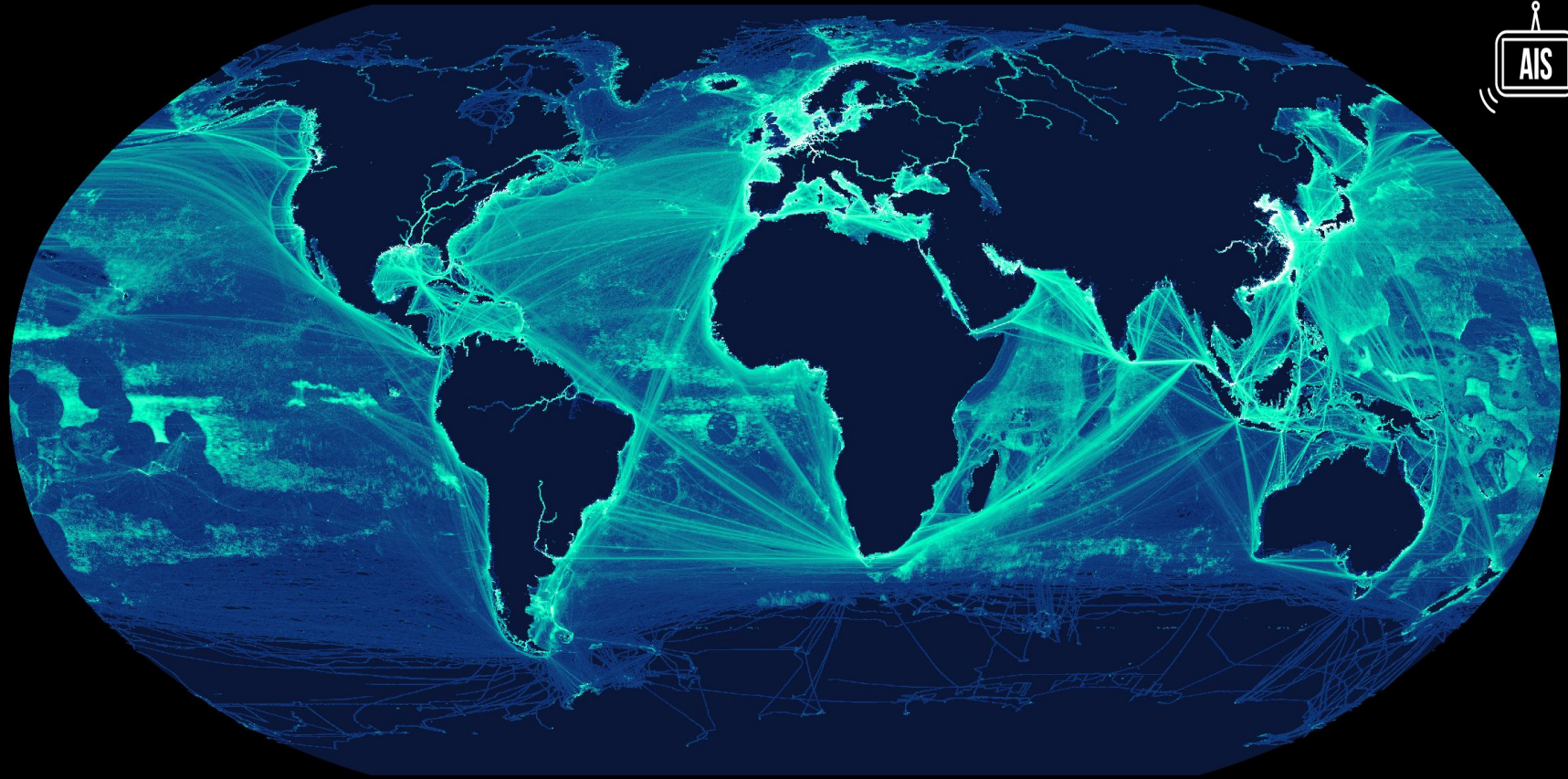
AIS messages transmit a lot of information about a vessel, such as identity, position, and speed.

AIS message are received by other ships and picked up by terrestrial and satellite receivers.



Global Fishing Watch

Vessels with AIS, 2018



Hours of vessel presence per 100 km^2

0.1 1 10 100 1,000 10,000

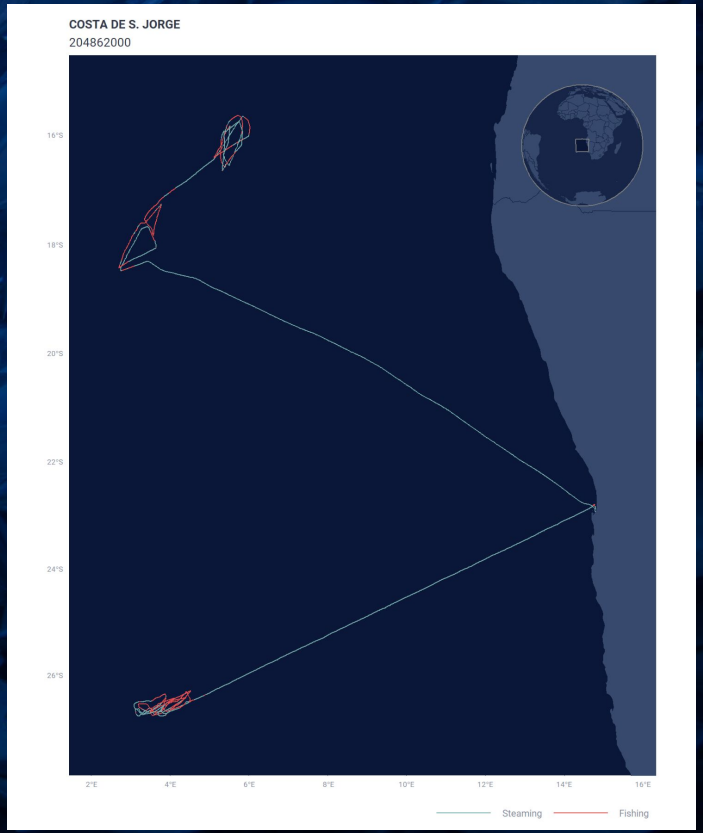
Global Fishing Watch

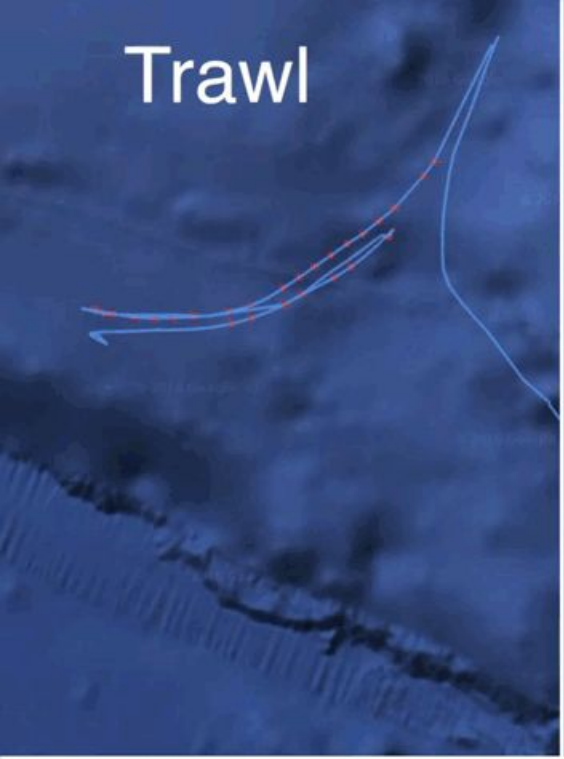
Fishing effort and vessel presence

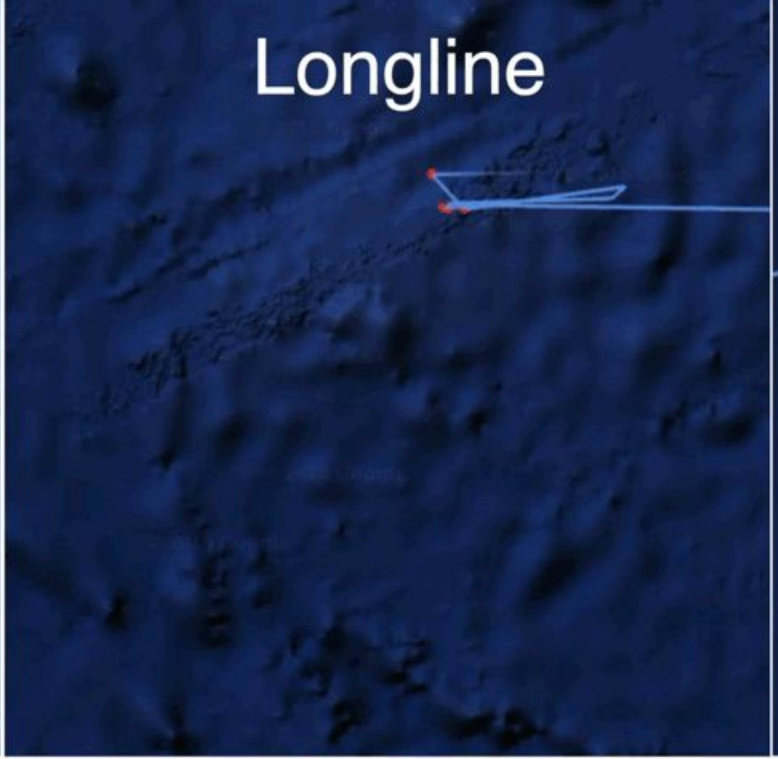
GFW identifies and classifies vessel presence and *apparent* fishing activity in the AIS data based on vessel movement by combining two convolutional neural network models.

A **vessel classification model** looks at all the AIS positions for a vessel and predicts the most likely vessel class (fishing or non-fishing) for that vessel. Information from vessel registries are also incorporated.

A (general) **fishing detection model** scores every AIS position as fishing, not fishing, or unknown.

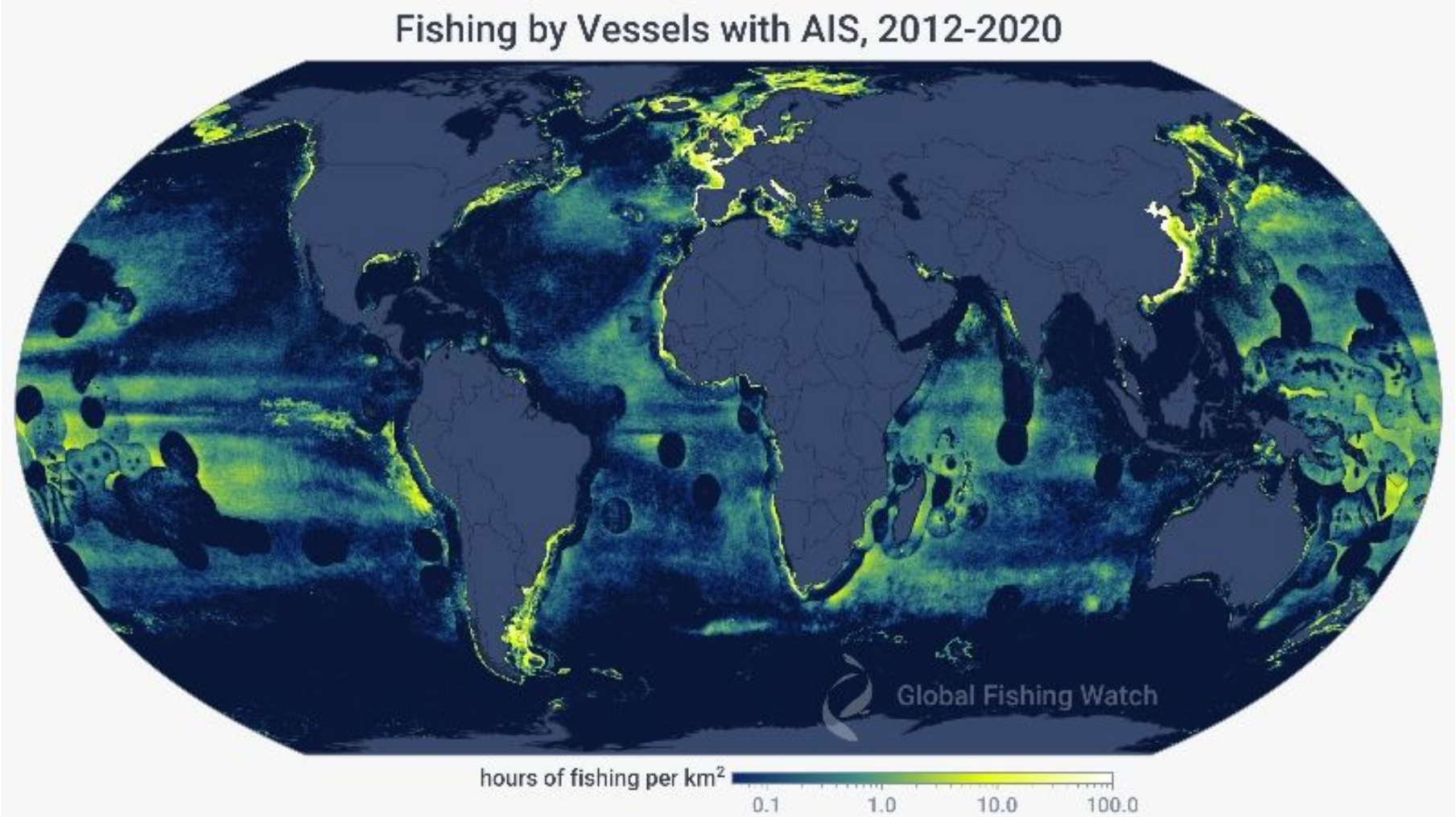



Trawl


Longline


Purse Seine


ZMGD


Fishing by Vessels with AIS, 2012-2020
Global Fishing Watch
hours of fishing per km²
0.1
1.0
10.0
100.0

R Package :: `gfwr`

The `gfwr` package provides convenient R functions to freely pull data from the GFW APIs.

Designed to allow researchers and other users with no API experience to work with data made accessible via the GFW APIs.

Will be periodically updated to incorporate new API endpoints and user feedback.

Open source and publicly available at:
https://github.com/GlobalFishingWatch/gfwr

R Package :: `gfwr`

The `gfwr` package currently works with the following GFW APIs:

Vessels

Vessel identity details based on AIS self-reported data and GFW models. Basic and advanced search modes.

Events

Vessel-level events for specific activities, such as fishing, port visits, and transshipment. Filter by date range and vessel.

Map Visualization (4Wings)

Gridded apparent fishing effort (e.g. rasters) based on AIS data. Multiple filtering and grouping options.

gfwr

Global Fishing Watch

Installation :: gfwr

You can install the development version of `gfwr` like so:

```
devtools::install_github("GlobalFishingWatch/gfwr")
```

Once everything is installed, you can load and use `gfwr` in your scripts with `library(gfwr)`

```
library(gfwr)
```

Authorization :: gfwr

The use of `gfwr` requires a GFW API token, which users can request from the GFW API Portal. Save this token to your `.Renviron` file (using `usethis::edit_r_environ()`) by adding a variable named `GFW_TOKEN` to the file (`GFW_TOKEN = "PASTE_YOUR_TOKEN_HERE"`). Save the `.Renviron` file and restart the R session to make the edit effective.

Then use the `gfw_auth` helper function to save the information to an object in your R workspace every time you need to extract the token and pass it to subsequent `gfwr` functions.

So you can do:

```
key <- gfw_auth()
```

gfwr
Global Fishing Watch

Vessels API :: gfwr

```
get_vessel_info(query = 224224000,
                search_type = "basic",
                dataset = "all",
                key = key)
#> # A tibble: 1 × 17
#>    name callsign first…¹ flag  geart…² id    imo   lastT…³ mmsi  msgCo…⁴ posCo…⁵
#>   <int> <chr>    <chr>   <chr> <lgl>   <chr> <chr> <chr>   <chr>   <int>   <int>
#> 1     1 EBSJ     2015-1… ESP   NA      3c99… 8733… 2019-1… 2242… 1887249   73677
#> # … with 6 more variables: shipname <chr>, source <chr>, vesselType <chr>,
#> #   years <list>, dataset <chr>, score <dbl>, and abbreviated variable names
#> #   ¹firstTransmissionDate, ²geartype, ³lastTransmissionDate, ⁴msgCount,
#> #   ⁵posCount
```

To combine different fields and do fuzzy matching to search the `carrier vessel` dataset:

```
get_vessel_info(query = "shipname LIKE '%GABU REEFE%' OR imo = '8300949'",
                search_type = "advanced", dataset = "carrier_vessel", key = key)
```

Events API :: gfwr

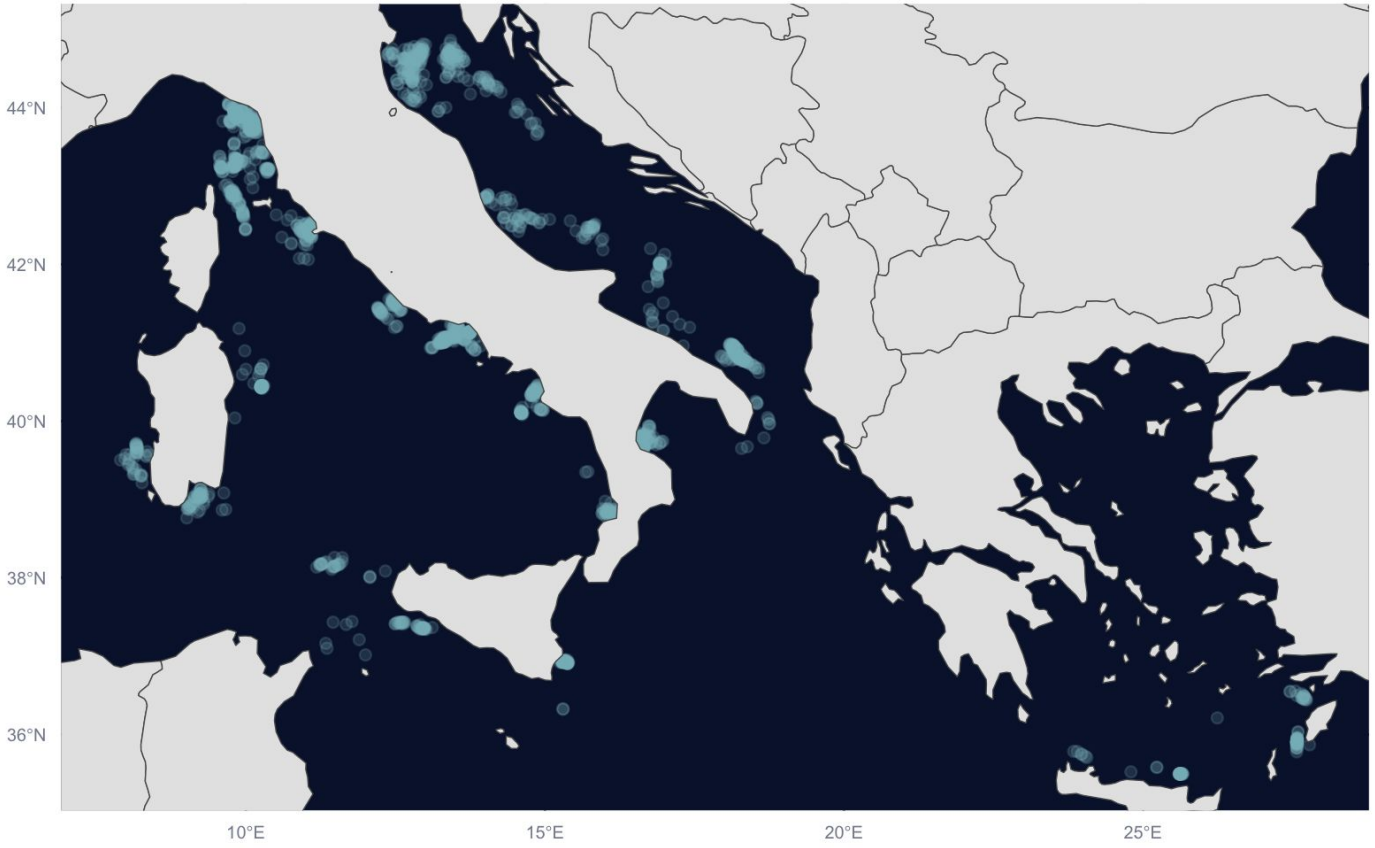
```
get_event(event_type='fishing',
          vessel = usa_trawler_ids,
          start_date = "2020-01-01",
          end_date = "2020-02-01",
          key = key
          )
#> [1] "Downloading 2 events from GFW"
#> # A tibble: 2 × 11
#>   id      type  start               end                   lat   lon regions
#>   <chr>   <chr> <dttm>              <dttm>              <dbl> <dbl> <list>
#> 1 777fb1… fish… 2020-01-09 23:15:22 2020-01-09 23:52:24  28.1 -94.0 <named list>
#> 2 f19d61… fish… 2020-01-10 12:59:23 2020-01-10 16:55:15  28.1 -93.9 <named list>
#> # … with 4 more variables: boundingBox <list>, distances <list>, vessel <list>,
#> #   event_info <list>
```

gfwr

Global Fishing Watch

Fishing events by Italian trawlers

2020-01-01 to 2020-06-01

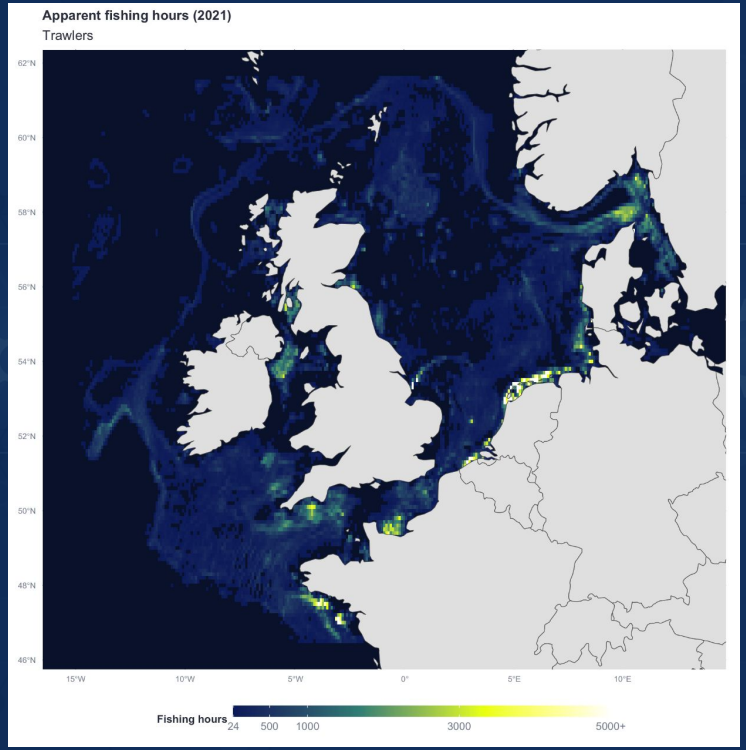


Map Visualization (4Wings) :: `gfwr`

The `get_raster()` function gets a dataframe of gridded apparent fishing effort and can be used to make heatmaps like the GFW Map.

Can select data for specific regions using three preset region types (EEZ, MPA, RFMO).

User can also provide their own geojson to download data for a custom region.



gfwr
Global Fishing Watch

Map Visualization (4Wings) :: `gfwr`

Multiple arguments are provided for customizing the output of `get_raster()`:

- `spatial_resolution`: `"low"` (0.1 degree) or `"high"` (0.01 degree)
- `temporal_resolution`: `"daily"`, `"monthly"`, or `"monthly"`
- `group_by`: `"vessel_id"`, `"flag"`, `"gearType"`, or `"flagAndGearType"`
- `date_range`: comma separated string of two dates. Must be one year or less (e.g. `"2021-01-01,2021-12-31"`)
- `region`: code geographically filter the data. This can be a one of three preset region types or a custom geojson
- `region_source` indicating the type of region (`"eez"`, `"mpa"`, `"rfmo"`, or `"user_json"`)

Example 1:: `gfwr`

Fishing activity within a region

The first example will use one of the preset region options to download and plot fishing activity

To get activity for a specific region, first use the `get_region_id()` function to find the id code of our region of interest.

```
# Use get_region_id function to get EEZ code for Italy
code_eez <- get_region_id(region_name = 'ITA', region_source = 'eez', key = gfw_auth())
```

```
#> # A tibble: 1 × 3
#>      id iso3  label
#>   <dbl> <chr> <chr>
#> 1  5682 ITA   Italy
```

Example 1 :: gfwr

Now we can download fishing activity using `get_raster()`, specifying choices for the remaining options for filtering and grouping the data:

```
# Download data for the Italian EEZ
eez_fish_df <- get_raster(spatial_resolution = 'low',
           temporal_resolution = 'yearly',
           group_by = 'flag',
           date_range = '2021-01-01,2021-10-01',
           region = code_eez$id,
           region_source = 'eez')
#> Rows: 5216 Columns: 6
#> ── Column specification ────────────────────────────────────────────────────
#> Delimiter: ","
#> chr (1): flag
#> dbl (5): Lat, Lon, Time Range, Vessel IDs, Apparent Fishing hours
#>
#> ℹ Use `spec()` to retrieve the full column specification for this data.
#> ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

gfwr
Global Fishing Watch

Example 1 :: `gfwr`

In this example, the `get_raster()` function returns a dataframe with 5,216 rows and six fields. The fields in the response will vary depending on the `group_by` choice.

```
#> # A tibble: 5,216 × 6
#>      Lat   Lon `Time Range` flag  `Vessel IDs` `Apparent Fishing hours`
#>    <dbl> <dbl>        <dbl> <chr>         <dbl>                   <dbl>
#>  1  37.4  13.1         2021 ITA             87                  5071.
#>  2  37.4  13.2         2021 ITA             74                  3266.
#>  3  37.2  13.5         2021 ITA             37                   905.
#>  4  40.9  17.5         2021 ITA             19                   314.
#>  5  35.5  12.6         2021 ITA             41                   828.
#>  6  35.4  12.8         2021 ITA             23                  1331.
#>  7  37.1  13            2021 ITA             37                   629.
#>  8  38.1  11.9         2021 ITA             19                   377.
#>  9  40    15.7         2021 ITA              4                    37.6
#> 10  39.6  16.7         2021 ITA             11                   181.
#> # … with 5,206 more rows
```

gfwr
Global Fishing Watch

Example 1 :: gfwr

The data includes, for each 10th degree grid cell, the total fishing hours in the Italian EEZ by vessels of different flag states during our date range.

```
#> # A tibble: 5,216 × 6
#>      Lat   Lon `Time Range` flag  `Vessel IDs` `Apparent Fishing hours`
#>    <dbl> <dbl>        <dbl> <chr>        <dbl>                   <dbl>
#>  1  37.4  13.1         2021 ITA             87                  5071.
#>  2  37.4  13.2         2021 ITA             74                  3266.
#>  3  37.2  13.5         2021 ITA             37                   905.
#>  4  40.9  17.5         2021 ITA             19                   314.
#>  5  35.5  12.6         2021 ITA             41                   828.
#>  6  35.4  12.8         2021 ITA             23                  1331.
#>  7  37.1  13            2021 ITA             37                   629.
#>  8  38.1  11.9         2021 ITA             19                   377.
#>  9  40    15.7         2021 ITA              4                    37.6
#> 10  39.6  16.7         2021 ITA             11                   181.
#> # … with 5,206 more rows
```

gfwr

Global Fishing Watch

Example 1 :: gfwr

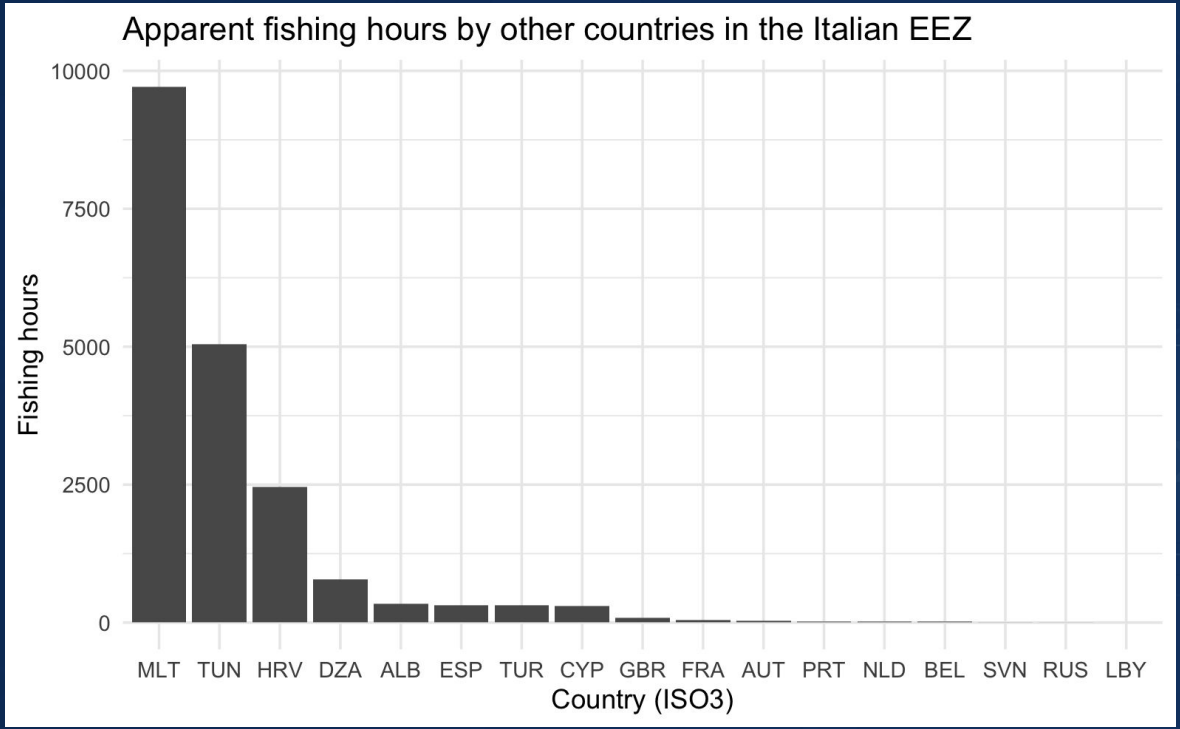
We can summarize the data and see what other flag states do the most fishing activity in the Italian EEZ:

```
eez_fish_df %>%
  filter(flag != 'ITA') %>%
  group_by(flag) %>%
  summarize(fishing_hours = sum(`Apparent Fishing hours`, na.rm = T)) %>%
  ggplot() +
  geom_col(aes(x = forcats::fct_reorder(flag, desc(fishing_hours)),
               y = fishing_hours)) +
  labs(title = 'Apparent fishing hours by other countries in the Italian EEZ',
       y = 'Fishing hours',
       x = 'Country (ISO3)') +
  theme_minimal()
```

gfwr

Global Fishing Watch

Example 1 :: gfwr



gfwr
Global Fishing Watch

Example 1 :: gfwr

Because the data includes fishing by all flag states, to make a map of all activity, we first need to summarize activity by grid cell:

```
eez_fish_all_df <- eez_fish_df %>%
  group_by(Lat, Lon) %>%
  summarize(fishing_hours = sum(`Apparent Fishing hours`, na.rm = T))
#> `summarise()` has grouped output by 'Lat'. You can override using the `.groups`
#> argument.
```

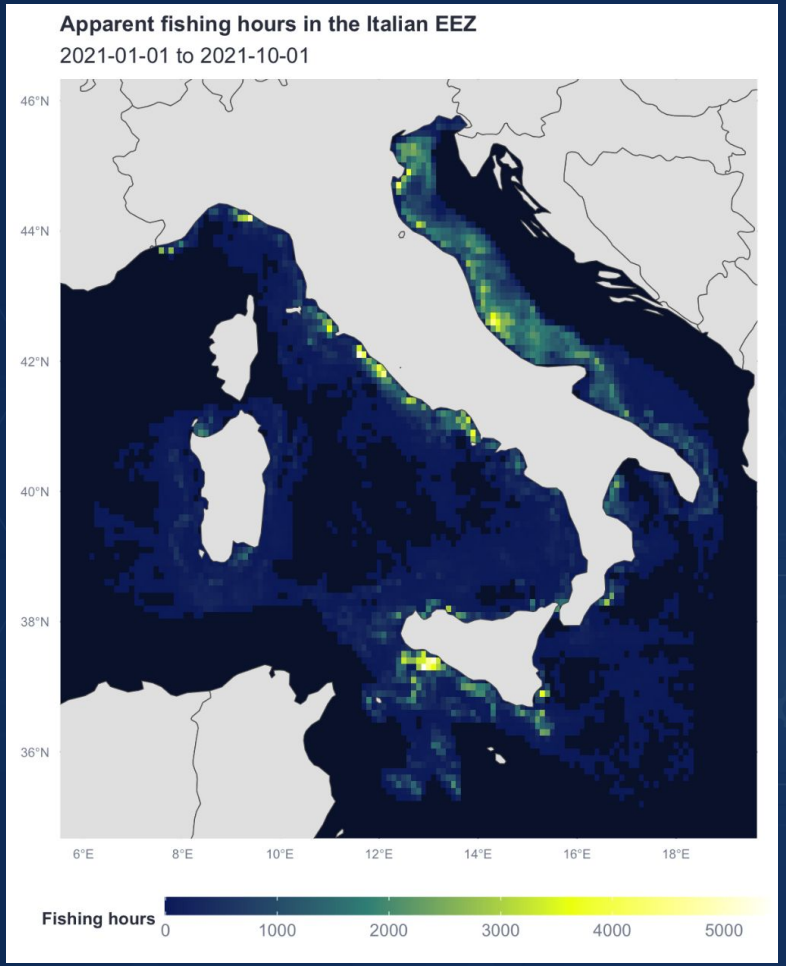
Example 1 :: gfwr

Now we can plot a map of the data with `ggplot2`.

```
eez_fish_all_df %>%
  ggplot() +
  geom_raster(aes(x = Lon,
                  y = Lat,
                  fill = fishing_hours)) +
  geom_sf(data = ne_countries(returnclass = 'sf', scale = 'medium')) +
  coord_sf(xlim = c(min(eez_fish_all_df$Lon),max(eez_fish_all_df$Lon)),
           ylim = c(min(eez_fish_all_df$Lat),max(eez_fish_all_df$Lat))) +
  scale_fill_gradientn(colors = map_effort_dark, na.value = NA) +
  labs(title = 'Apparent fishing hours in the Italian EEZ',
       subtitle = '2021-01-01 to 2021-10-01',
       fill = 'Fishing hours') +
  map_theme
```

gfwr

Global Fishing Watch



gfwr

Global Fishing Watch

Example 2 :: `gfwr`

Fishing activity within a custom region

The second example will download and plot fishing activity using geojson, which allows for downloading activity in a custom region defined by the user

Note: The `get_raster()` function does not have a built in method to download global scale data but you can use a geojson to accomplish this

Tip: The website https://boundingbox.klokantech.com/ is a handy resource for creating custom geojson

The geojson should be stored as a string with the following format:

```
'{"geojson":{"type":"Polygon","coordinates":[insert coordinates]}}'
```

Example 2 :: gfwr

Define the geojson object and pass it to the `region` argument. Indicate that the `region_source = 'user_json'`

```
region_json = '{"geojson":{"type":"Polygon","coordinates":[[[-15.1296846569,46.500565613],[13.25

geojson_df <- get_raster(spatial_resolution = 'low',
            temporal_resolution = 'yearly',
            group_by = 'gearType',
            date_range = '2021-01-01,2021-12-31',
            region = region_json,
            region_source = 'user_json')
#> Rows: 49246 Columns: 6
#> ── Column specification ────────────────────────────────────────────────────────
#> Delimiter: ","
#> chr (1): geartype
#> dbl (5): Lat, Lon, Time Range, Vessel IDs, Apparent Fishing hours
#>
#> ℹ Use `spec()` to retrieve the full column specification for this data.
#> ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

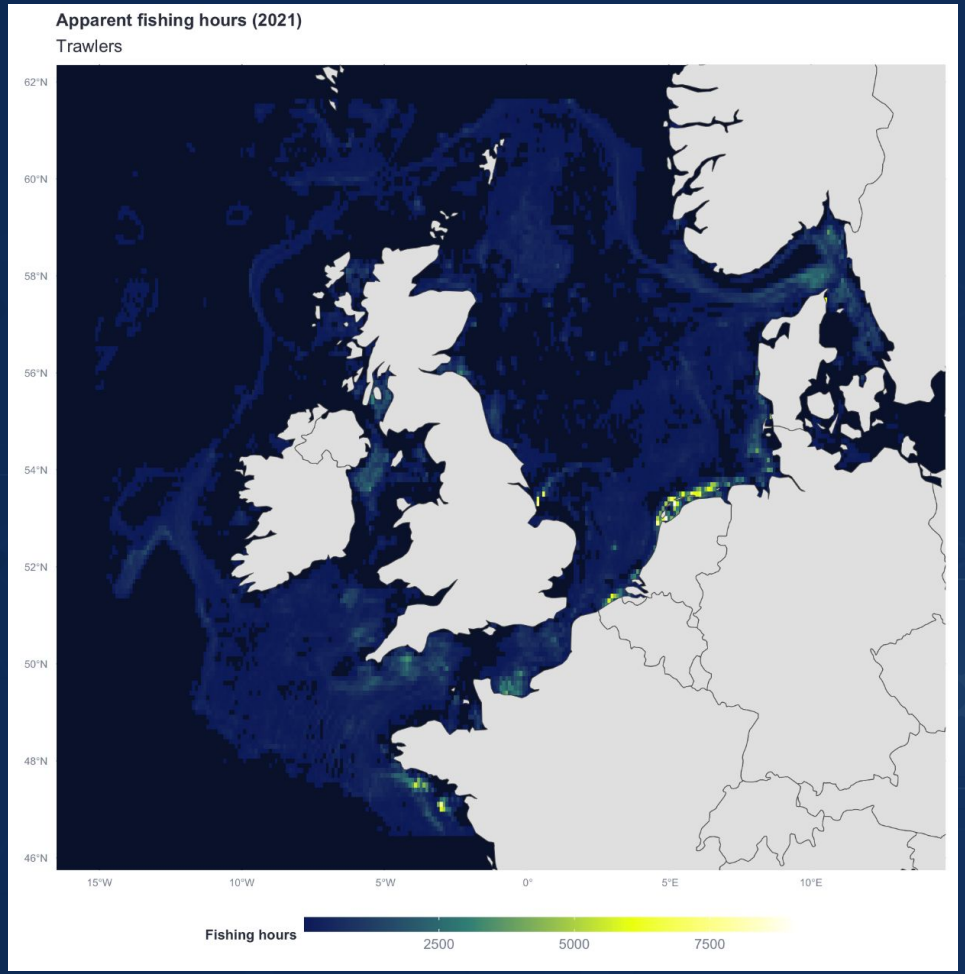
gfwr
Global Fishing Watch

Example 2 :: gfwr

Now let's plot all 2021 fishing activity by trawlers in our custom region, including only cells with at least 24 hours of apparent fishing activity.

```
geojson_df %>%
  filter(geartype == 'trawlers') %>%
  filter(`Apparent Fishing hours` > 24) %>%
  ggplot() +
  geom_raster(aes(x = Lon,
                  y = Lat,
                  fill = `Apparent Fishing hours`)) +
  geom_sf(data = ne_countries(returnclass = 'sf', scale = 'medium')) +
  coord_sf(xlim = c(min(geojson_df$Lon),max(geojson_df$Lon)),
           ylim = c(min(geojson_df$Lat),max(geojson_df$Lat))) +
  scale_fill_gradientn(colors = map_effort_dark, na.value = NA) +
  labs(title = 'Apparent fishing hours (2021)',
     subtitle = 'Trawlers',
     fill = 'Fishing hours') +
  map_theme
```

gfwr
Global Fishing Watch



gfwr

Global Fishing Watch

Final Thoughts :: `gfwr`

Global Fishing Watch is dedicated to creating and publicly sharing open data and tools to advance ocean governance

The `gfwr` package provides easy access to GFW data made available via the GFW API Portal.

`gfwr` currently includes functions to access GFW's Vessels, Events, and Map Visualization (4Wings) APIs and will be updated to access new GFW APIs as they become available

`gfwr` is open source and we welcome contributions! Clone the repository at https://github.com/GlobalFishingWatch/gfwr

Thank you

Global Fishing Watch is an international nonprofit organization dedicated to advancing ocean governance through increased transparency of human activity at sea. By creating and publicly sharing map visualizations, data and analysis tools, we aim to enable scientific research and transform the way our ocean is managed. We believe human activity at sea should be public knowledge in order to safeguard the global ocean for the common good of all.

Discover more at globalfishingwatch.org

Global Fishing Watch

Our datasets

Fishing effort and vessel presence
Vessel-specific fishing activity identified from vessel tracking data by GFW algorithms. Also includes non-fishing vessel activity.

Vessel identity
Comprehensive vessel information (MMSI, flag, gear type, length, authorizations, etc.) for all vessels listed on public vessel registries

Port visits and voyages
Global database of anchorages visited by vessels broadcasting AIS and the voyages they make between ports

Transshipment
Detection of potential transshipment events

Non-transmitting vessels
Detection of vessels not broadcasting AIS or VMS using satellite imagery

Global Fishing Watch

```r
# Map theme with dark background
map_theme <- ggplot2::theme_minimal() +
  ggplot2::theme(
    panel.border = element_blank(),
    panel.background = element_rect(fill = "#0a1738", color = NA),
    legend.position = "bottom", legend.box = "vertical",
    legend.key.height = unit(3, "mm"),
    legend.key.width = unit(20, "mm"),
    legend.title.align = 0.5,
    legend.text = element_text(color = "#848b9b", size = 8),
    legend.title = element_text(face = "bold", color = "#363c4c", size = 8),
    plot.title = element_text(face = "bold", color = "#363c4c", size = 10),
    plot.subtitle = element_text(color = "#363c4c", size = 10),
    axis.title = element_blank(),
    axis.text = element_text(color = "#848b9b", size = 6),
    panel.grid.major = element_line(color = "#0a1738"),
    panel.grid.minor = element_line(color = "#0a1738"))

# Palette for fishing activity
map_effort_dark <- c("#0c276c", "#3b9088", "#eeff00", "#ffffff")
```

gfwr

Global Fishing Watch

Which APIs are available?

Only AIS-derived data (for now) and and based on our peer reviewed research

Fishing effort Discover fishing patterns for nearly 70,000 vessels broadcasting on AIS, displayed across space and time at global scale, in a map visualization.

Fishing events Find vessel position data showing possible fishing events in detail, anywhere on the ocean.

Vessel search and identity Search fishing, carrier and support vessels through multiple identifiers and cross-check identity information derived from AIS self-reported data.

Encounters Find data on vessel encounters and possible transshipment events on a global scale for improved knowledge of at-sea catch transfers.

Loitering Locate information on vessel loitering to increase awareness of potential at-sea encounters and transshipment events.

Port visits Find data on port visits and patterns of vessel behavior across space and time.

gfwR Access data from its application programming interfaces, or APIs, directly through R—a specialized programming language for data processing, statistical analysis and data visualizations.

Global Fishing Watch

New APIs coming soon…

- **Statistics API:** get fishing activity and/or event statistics for the world or a region
- **Intentional AIS disabling events API**: get events where fishing vessels intentionally disabling their vessel tracking system (read paper)
- **More vessel types**: get non-fishing vessel and vessel details in our Vessel API
- **Vessel identity**: a new version of Vessel API including identity and authorization data from regional and national registries (read paper)
- **Vessel Insights API:** get vessel indicators like AIS coverage, fishing events in no take MPA and IUU list status, likely disabling events, name or identity changes etc.
- **Ports and Voyages API** : get vessel trip and activity details between port visits.

Global Fishing Watch