

Marine Sensitivity

Project Documentation

Ben Best

2024-09-16

Table of contents

Preface	4
1 Introduction	5
I Science	6
2 Science	7
3 Stressors	8
3.1 Offshore Wind Energy	8
3.2 Oil & Gas	9
4 Receptors	10
4.1 Corals	10
4.2 Invertebrates	10
4.3 Fish	10
4.4 Marine Mammals	10
4.5 Seabirds	10
4.6 Sea Turtles	10
5 Exposure	11
II Software	12
6 Software	13
6.0.1 Interactive Applications	13
6.0.2 Overcoming Challenges with Large Spatial Data	13
6.0.3 Github Repositories	14
6.0.4 Software Components	16
7 Server	17
7.1 Setup	17
7.2 Services	17

8 Database	20
8.1 Table and Column Naming Conventions	20
8.2 Species Distribution Models	20
9 Workflows	21
10 APIs	22
11 Libraries	24
12 Apps	25
13 Docs	28
14 Summary	29
References	30
Appendices	31
Glossary	31
III Applications	32
Areas of Interest	33
Bird Hotspots	34
Regional Map	35
Distributions, Vector	36
Distributions, Raster	37
Vulnerability Mapper	38

Preface

This is a Quarto book.

1 Introduction

This Marine Sensitivity (MS) project of [BOEM](#) seeks to assess the sensitivity of marine species to offshore energy development, whether oil & gas or wind. By combining the best available species distributions with known species sensitivities we can map out areas of the ocean that are most vulnerable to human activities. This information can be used to inform decisions about where to place energy infrastructure and/or implement mitigations to minimize impacts on the marine environment.

This is a process, not a product. Information is imperfect, especially given the large expanse of US waters. Distributions and abundance of species change, modified increasingly by climate change and human activities. Knowledge on species sensitivities continues to expand with more research. And finally the methods for both modeling and distributing all this information continue to improve. We aim to provide a transparent and reproducible process that can be regularly updated as new data and methods become available.

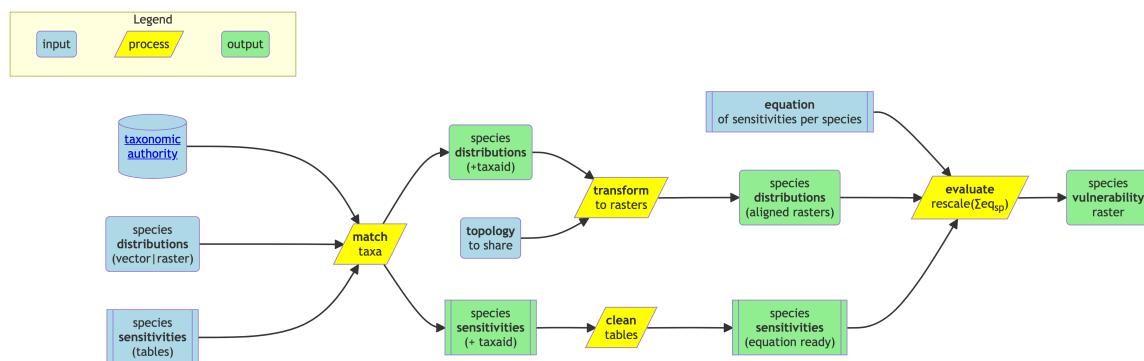


Figure 1.1: Flowchart of process for incorporating marine species sensitivities with distributions and generating a holistic vulnerability map.

Part I

Science

2 Science

The term vulnerability (V) is a function of exposure (E), sensitivity (S) and adaptive capacity (A).

$$V = f(E, S, A) \quad (2.1)$$

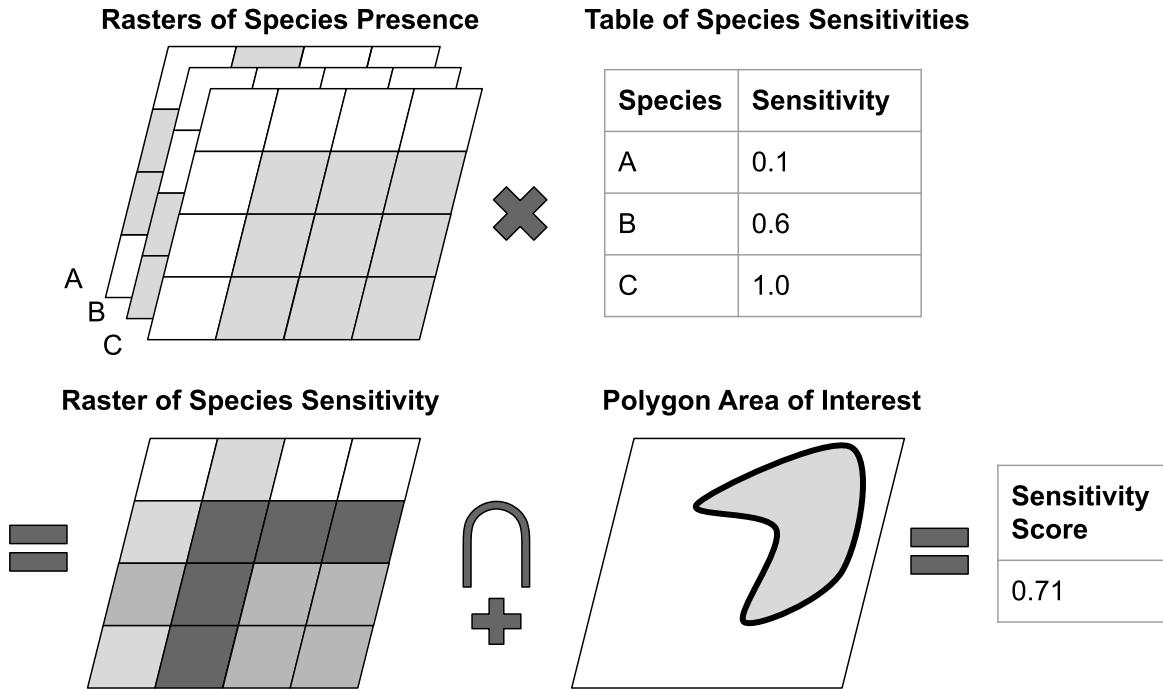


Figure 2.1: Overview of process.

$$cell_S = \sum_{spp} p * w \quad (2.2)$$

The raster of sensitivity (S) contains cells representing a sum across species (spp) of presence (p) multiplied by the sensitivity weight (w) (Equation 2.2).

3 Stressors

3.1 Offshore Wind Energy

Evaluation of stressors from the offshore wind industry needs to be evaluated based on human activities given the phase of development, whether pre-construction, construction, operation or decommissioning (Figure 3.1).

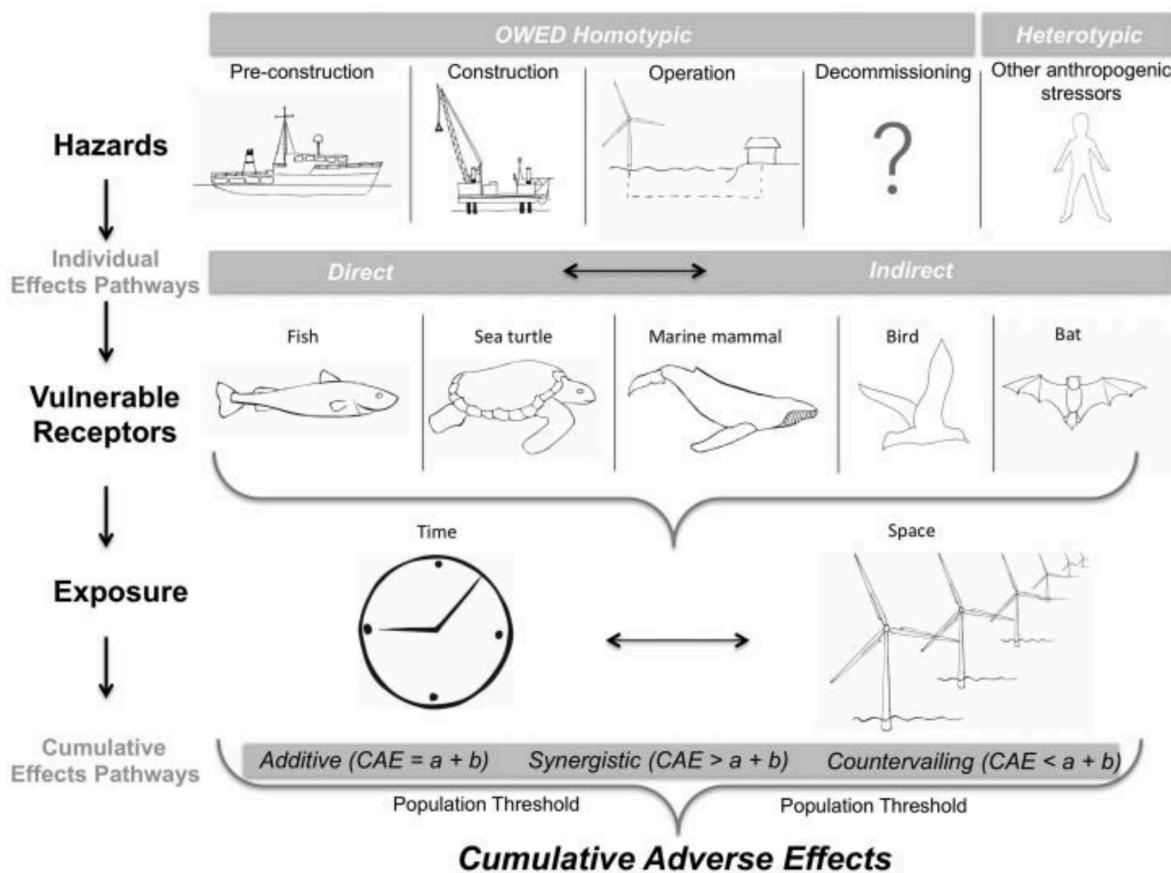


Figure 3.1: Cumulative adverse effects of offshore wind energy development on wildlife (Goodale and Milman 2016).

3.2 Oil & Gas

4 Receptors

Receptors are the species and habitats that are potentially impacted by the human activity.

4.1 Corals

4.2 Invertebrates

4.3 Fish

4.4 Marine Mammals

4.5 Seabirds

4.6 Sea Turtles

5 Exposure

Cumulative exposure (Figure 3.1) is important for understanding impacts to a population.

Part II

Software

6 Software

6.0.1 Interactive Applications

We have developed a series of interactive applications to explore the data and results of the MS project. These applications allow users to visualize the data, explore the results, and interact with the data in a more intuitive way. The applications are built using the [shiny](#) package in R, which allows us to easily create a user interface with complex reactivity for an interactive web application easily accessed through a web browser. The applications are designed to be user-friendly and intuitive, with interactive maps, charts, and tables that allow users to explore the data in a more dynamic way.

6.0.2 Overcoming Challenges with Large Spatial Data

The MS project incorporates many large spatial datasets that are problematic to render in a typical interactive application. For instance, the most common interactive mapping R package [leaflet](#) has a 4MB limitation for displaying rasters (see “Large Raster Warning” in [Raster Images • leaflet](#)). Vectors (i.e., points, lines and polygons) get smoothed when containing many vertices, but contiguity gets lost between polygons and rendering degrades to non-visible depending on the internet speed of the user’s connection.

To work around these limitations, we have implemented “cloud native” web services and formats (see also [Cloud-Optimized Geospatial Formats Guide](#)). Our implementations effectively reduce the size of any given spatial object based on the zoom level of the user’s browser. For rasters, we use cloud-optimized GeoTIFFs (COGs) and for vectors, we use Mapbox Vector Tiles (MVT). These formats are designed to be fast and efficient for web mapping applications, and they allow us to display large spatial datasets in an interactive web application without sacrificing performance or usability. Let’s take a closer look at implementation of each.

6.0.2.1 Raster: Cloud-Optimized GeoTIFFs (COGs) and Titiler

Historically, to read a raster, such as a GeoTIFF, from the web, the client software would have to read the entire file before rendering. Cloud Optimized GeoTIFFs ([COGs](#)) take advantage of [HTTP GET range requests](#) to read only the part of the file needed for rendering. So a COG stores quadtree simplifications of the original raster at multiple zoom levels and metadata for accessing their byte ranges in the file in the metadata header. This allows the client software

to request only the parts of the file needed for rendering, which can greatly reduce the amount of data transferred and speed up rendering. This is for accessing the raw data in pixel values, e.g., for a raster of species distribution then the abundance of a species in each cell. We would want to also apply a color ramp to visualize the data. The open-source ([TiTiler](#)) software is a lightweight web service that serves up these color ramped tiles on the fly. So COGs can be stored on a simple file server (like Amazon S3 or Azure Blob Storage) and served up as interactive web maps with TiTiler as an intermediary between the COG files and the client accessing the interactive Shiny mapping app (Figure 6.1).

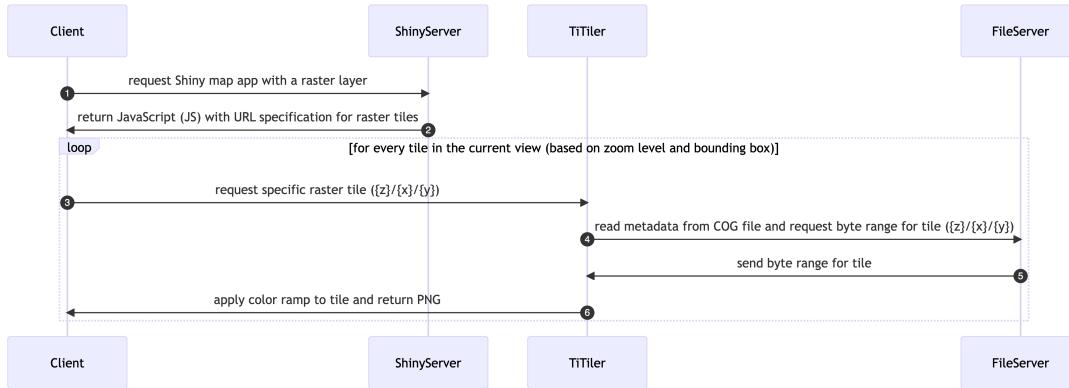


Figure 6.1: Sequence diagram implementing large raster interactive display using Cloud-Optimized GeoTIFFs (COGs) and Titiler in a Shiny mapping app.

6.0.2.2 Vector: Mapbox Vector Tiles (MVTs) and pg_tileserv

Although “cloud native” vector formats exist for simple file storage (see [Cloud-Optimized Geospatial Formats Guide](#)), none of these allow for flexible filtering and manipulation. Instead, we use PostgreSQL with the spatial extension ([PostGIS](#)) to store the vector data and serve it as Mapbox Vector Tiles (MVTs) using the [pg_tileserv](#) web service written in the language Go, which is very fast. This means that we don’t have to pre-render the MVTs (such as you might do with [tippecanoe](#)), but can instead serve the raw vector data directly from the database and let [pg_tileserv](#) handle the rendering on the fly. Filters (in the form of [CQL](#)) can be applied to the request. Symbology is rendered client-side via JavaScript, which allows for interactive hover and click events on vector objects (e.g., BOEM aliquot). Some speed-up is enabled by implementing a [Varnish](#) cache service in between. We can even write our own database functions for customized rendering, such as H3 hexagonal summaries. This allows us to serve vector data as web maps with minimal configuration and setup, and it provides a fast and efficient way to display large vector datasets in an interactive web application (Figure 6.2).

6.0.3 Github Repositories

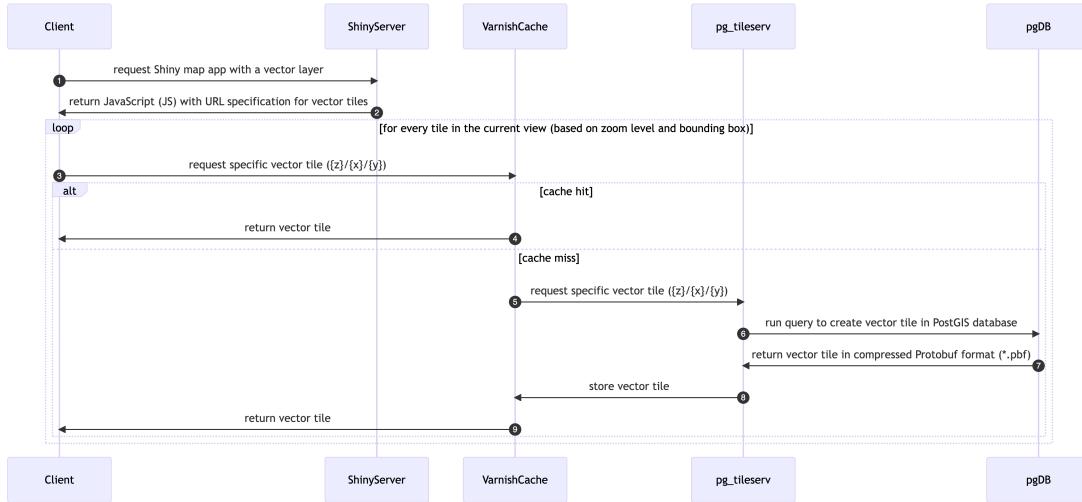


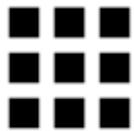
Figure 6.2: Sequence diagram implementing large vector interactive display using Mapbox Vector Tiles (MVTs) and pg_tileserv in a Shiny mapping app.

repo	description
api	application programming interface (API) using R Plumber package
apps	Shiny applications
docs	documentation for BOEM's offshore environmental sensitivity index products
manuscripts	Manuscripts with review of sensitivities by industry and receptors (species, habitats, human uses)
MarineSensitivity.github.io	default website
msens	R library of functions for mapping marine sensitivities, sponsored by BOEM
objectives	repository for issues spanning multiple repositories and doing big picture roadmapping
server	server setup for R Shiny apps, RStudio IDE, R Plumber API, PostGIS database, pg_tileserv
workflows	scripts for testing data analytics and visualization as well as production workflows

6.0.4 Software Components



APIs
application programming
interfaces (APIs)



Apps
interactive applications using
Shiny



Database
PostgreSQL database
extended spatially with
PostGIS



Docs
technical documentation



Libraries
documented functions as an R
package



Server
server software configuration
using Docker



Workflows
scripts for exploring plus
production workflows

7 Server

The server is for serving up any web services outside those of Github (e.g., [website](#), [docs](#) and R package [msens](#)) using [Docker](#) (see the [docker-compose.yml](#); with reverse proxying from subdomains to ports by [Caddy](#)).

7.1 Setup

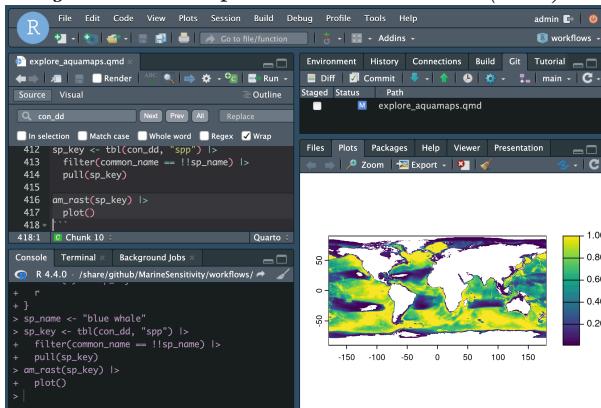
For instructions on launching an Amazon instance and installing the server software, see [Server Setup](#) · [MarineSensitivity/server Wiki](#).

7.2 Services

The server is running the following services:

- **RStudio**

integrated development environment (IDE) to code and debug directly on the server

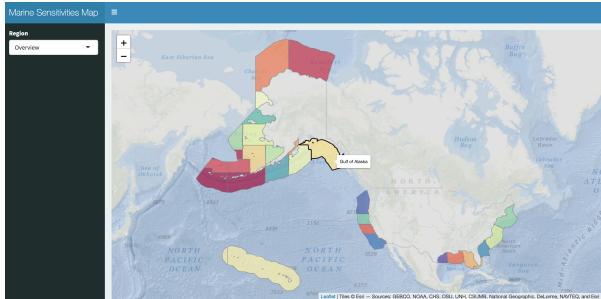


[More info...](#)

- **Shiny**

interactive applications

e.g., shiny.marinesensitivity.org/map



[More info..](#)

- **PGadmin**

PostGreSQL database administration interface

A screenshot of the PgAdmin PostgreSQL database administration interface. The left sidebar shows the 'Object Explorer' with a tree view of database objects. The main area displays a query result for the table 'public.ply_rgns_s05'. The results show 27 rows of data, including columns like shft_key, ctr_idn, rgn_key, rgn_name, shft_name, ctr_lat, ctr_lon, area_km2, and geometry. To the right of the results, there is a 'Geometry Viewer' window showing a map of the same marine sensitivity zones as the top figure, overlaid on a world map. The viewer includes controls for zooming and panning.

[More info..](#)

- **api**

custom API: using R plumber

A screenshot of the MarineSensitivities Custom API documentation generated by Swagger. At the top, there is a navigation bar with links for 'Swagger', 'https://api.marinesensitivities.org/openapi.json', and 'Explore'. Below this is a header for 'MarineSensitivities Custom API 1.0.0 OAS3'. A 'Servers' dropdown is set to 'https://api.marinesensitivities.org/'. The main content area shows the 'default' endpoint with several API operations listed:

- GET /echo**: Echo back the input
- GET /plot**: Plot a histogram
- POST /sum**: Return the sum of two numbers
- GET /**: redirect to the swagger interface

[More info..](#)

- **swagger**

generic database API: using PostGREST

Datasets: **HTTP**

Introspection

- geography_columns**
 - GET /geography_columns**
 - POST /geography_columns**
 - DELETE /geography_columns**
 - PATCH /geography_columns**
- geometry_columns**
 - GET /geometry_columns**
 - POST /geometry_columns**

[More info..](#)

- **tile**

spatial database API: using pg_tileserv for serving vector tiles

pg_tileserv

Service Metadata

- [index.json](#) for layer list

Table Layers

- aquamaps.cells ([preview](#) | [json](#))
- public.ply_rgns ([preview](#) | [json](#))
- public.ply_rgns_s05 ([preview](#) | [json](#))
- public.ply_shfis ([preview](#) | [json](#))
- public.ply_shfis_s05 ([preview](#) | [json](#))
- public.sdm_geometries ([preview](#) | [json](#))
- raw.boem_ak_blk_clip ([preview](#) | [json](#))
- raw.boem_ak_prot_clip ([preview](#) | [json](#))
- raw.boem_ad_aq ([preview](#) | [json](#))
- raw.boem_at_blk_clip ([preview](#) | [json](#))
- raw.boem_at_prot_clip ([preview](#) | [json](#))
- raw.boem_gom_blk_clip ([preview](#) | [json](#))
- raw.boem_gom_prot_clip ([preview](#) | [json](#))
- raw.boem_pc_aq ([preview](#) | [json](#))
- raw.boem_pc_blk_clip ([preview](#) | [json](#))
- raw.boem_pc_prot_clip ([preview](#) | [json](#))
- raw.boem_usa_mhk_plan ([preview](#) | [json](#))
- raw.boem_usa_wind_lease ([preview](#) | [json](#))
- raw.boem_usa_wind_plan ([preview](#) | [json](#))
- raw.mr_eaz ([preview](#) | [json](#))

Function Layers

- [public.sdm_spatial](#) ([preview](#) | [json](#))

Serves the Species Distribution Model given parameters: dataset_key, species_key, popn, time_interval, variable.

[More info..](#)

8 Database

8.1 Table and Column Naming Conventions

- Table names are plural and use all lower case.
- Unique identifiers are suffixed with:
 - *_id for unique integer keys;
 - *_key for unique string keys;
 - *_seq for auto-incrementing sequence integer keys.
- Column names are singular and use snake_case.
- Foreign keys are named with the singular form of the table they reference, followed by _id.
- Primary keys are named id.

8.2 Species Distribution Models

See entity relationship diagram (ERD) for the species distribution models (SDM) database tables in this workflow:

- [Create SDM Tables](#)

And example of ingesting SDM outputs into the database in this workflow:

- [Ingest GoMex cetacean & sea turtle SDMs](#)

9 Workflows

Workflows are scripts for testing data analytics and visualization as well as production workflows for ingesting data. See:

- marinesensitivity.org/workflows
rendered html pages from the scripts (as Quarto notebooks)
- github.com/MarineSensitivity/workflows
source code in the Github repository

10 APIs

There three APIs, each used for different purposes:

1. api

custom API: using R plumber
source: [MarineSensitivity/api](#)

The screenshot shows the MarineSensitivity Custom API documentation generated by Swagger. At the top, there's a navigation bar with the title "MarineSensitivity Custom API 1.0.0 OAS3", the URL "https://api.marinesensitivities.org/openapi.json", and a "Explore" button. Below the navigation, there's a "Servers" dropdown set to "https://api.marinesensitivities.org/". The main content area is titled "default" and contains four API endpoints:

- GET /echo**: Echo back the input.
- GET /plot**: Plot a histogram.
- POST /sum**: Return the sum of two numbers.
- GET /**: redirect to the swagger interface.

2. swagger

generic database API: using PostGREST
source: Postgres database, non-spatial

The screenshot shows the standard public schema documentation generated by PostGREST. At the top, there's a navigation bar with the title "standard public schema 1.0.0 OAS3", the URL "https://rest.MarineSensitivity.org/", and a "Explore" button. Below the navigation, there's a "Schemes" dropdown set to "HTTP". The main content area is titled "Introspection" and lists several endpoints:

- GET /**: OpenAPI description (No document).
- geography_columns**
 - GET /geography_columns**
- geometry_columns**
 - GET /geometry_columns**
 - POST /geometry_columns**
 - DELETE /geometry_columns**
 - PATCH /geometry_columns**

3. tile

spatial database API: using pg_tileserv for serving vector tiles

source: Postgres database, spatial

pg_tileserv

Service Metadata

- [index.json](#) for layer list

Table Layers

- aquamaps.cells ([preview](#) | [json](#))
- public.ply_rgns ([preview](#) | [json](#))
- public.ply_rgns_s05 ([preview](#) | [json](#))
- public.ply_shfts ([preview](#) | [json](#))
- public.ply_shfts_s05 ([preview](#) | [json](#))
- public.sdm_geometries ([preview](#) | [json](#))
- raw.boem_ak_blk_clip ([preview](#) | [json](#))
- raw.boem_ak_prot_clip ([preview](#) | [json](#))
- raw.boem_atl_atlq ([preview](#) | [json](#))
- raw.boem_atl_blk_clip ([preview](#) | [json](#))
- raw.boem_gom_blk_clip ([preview](#) | [json](#))
- raw.boem_gom_prot_clip ([preview](#) | [json](#))
- raw.boem_pc_atlq ([preview](#) | [json](#))
- raw.boem_pc_blk_clip ([preview](#) | [json](#))
- raw.boem_pc_prot_clip ([preview](#) | [json](#))
- raw.boem_usa_mnh_plan ([preview](#) | [json](#))
- raw.boem_usa_wind_lease ([preview](#) | [json](#))
- raw.boem_usa_wind_plan ([preview](#) | [json](#))
- raw.mr_eez ([preview](#) | [json](#))

Function Layers

- [public.sdm_spatial](#) ([preview](#) | [json](#))

Serves the Species Distribution Model given parameters: dataset_key, species_key, popn, time_interval, variable.

11 Libraries

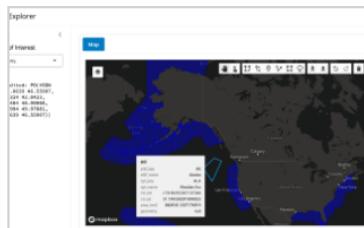
By creating an R package, we can document functions and make them easily available to other users.

- [msens](#)

R library of functions for mapping marine sensitivities, sponsored by BOEM

Functions can *read* data from the one [APIs](#) (which communicate with the [Database](#)), *analyze* the data, *visualize* the results and store some smaller *data*.

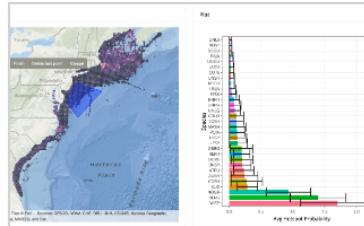
12 Apps



Areas of Interest

[AREAS](#) [VECTOR](#)

Map high resolution Areas of Interest (using vector tiles) for visualization (and later summarization).

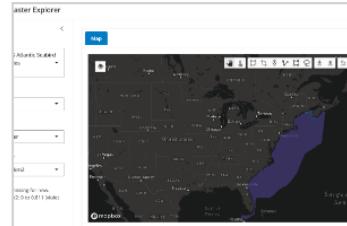


Bird Hotspots

[DISTRIBUTIONS](#)

[HOTSPOTS](#)

Bird hotspots application showing hotspot probability for species present given drawn Area of Interest.

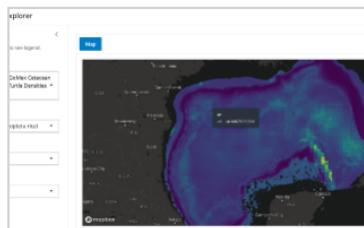


Distributions, Raster

[DISTRIBUTIONS](#)

[RASTER](#)

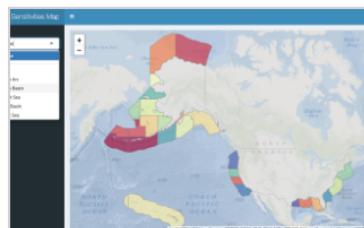
Show species distributions with high resolution rasters as cloud-optimized GeoTIFFs (COGs).



Distributions, Vector

[DISTRIBUTIONS](#)

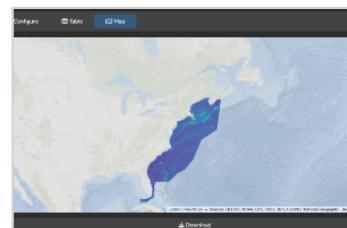
[VECTOR](#)



Regional Map

[AREAS](#)

Basic interactive map of BOEM



Vulnerability Mapper

[DISTRIBUTIONS](#)

[VULNERABILITIES](#)

See also details of individual applications in the Appendix.

13 Docs

Technical documentation is principally in this book:

- marinesensitivity.org/docs
the main documentation site
- github.com/MarineSensitivity/docs
source code in the Github repository

But there are also some other self-documenting resources:

- marinesensitivity.org/msens
documented R functions

14 Summary

...

References

- Goodale, M. Wing, and Anita Milman. 2016. “Cumulative Adverse Effects of Offshore Wind Energy Development on Wildlife.” *Journal of Environmental Planning and Management* 0 (0): 1–21. <https://doi.org/10.1080/09640568.2014.973483>.
- Ross, Pauline M., Elliot Scanes, Maria Byrne, Tracy D. Ainsworth, Jennifer M. Donelson, Shawna A. Foo, Pat Hutchings, Vengatesen Thiagarajan, and Laura M. Parker. 2023. “Surviving the Anthropocene: The Resilience of Marine Animals to Climate Change.” In. CRC Press.

Glossary

acclimatisation the adjustment of an organism to environmental conditions in the field or environment rather than the laboratory without an adjustment in their genetics. Acclimation has been used to describe phenotypically plastic responses in natural conditions. Source: Ross et al. (2023).

adaptation the evolutionary mechanism where natural selection of traits is genetically passed on, typically over many generations, to create an organism suited to the environment. Source: [rossRoss et al. (2023)

adaptive capacity the capacity of the ecosystem or organism to improve and reorganise in response to stress such as climate change through phenotypic plasticity (acclimation, acclimatisation) or adaptation, distributional shifts, and rapid evolution of traits suited to new conditions. Source: Ross et al. (2023).

epigenetics the modification of phenotype plasticity of an organism through altered gene expression without an alteration to the DNA sequence. ‘Epi’ means above the DNA and includes DNA methylation, modification of histones, and non-coding RNA. Source: Ross et al. (2023).

exposure the magnitude of the change in the environment

fecundity the maximum physiological potential reproductive output of an organism to produce offspring (reproductive output). This differs from fertility, which is the number of offspring born. Source: Ross et al. (2023)

MBON Marine Biodiversity Observation Network; see [MarineBON.org](#)

resilience the capacity of an ecosystem, society, or organism to absorb disturbance and reorganise while undergoing change so as to retain essentially the same function, structure, identity, and feedbacks. Resilience reflects the degree to which a complex adaptive system is determined by its capacity to reorganise and adapt in order to avoid being disturbed again. Source: Ross et al. (2023).

sensitivity the magnitude of response to the change

stressor the stimulus that causes stress to an organism

vulnerability combination of exposure and sensitivity

Part III

Applications

Areas of Interest

Map high resolution Areas of Interest (using vector tiles) for visualization (and later summarization).

- [🌐 website](#)
- [💬 code](#)

Area Explorer

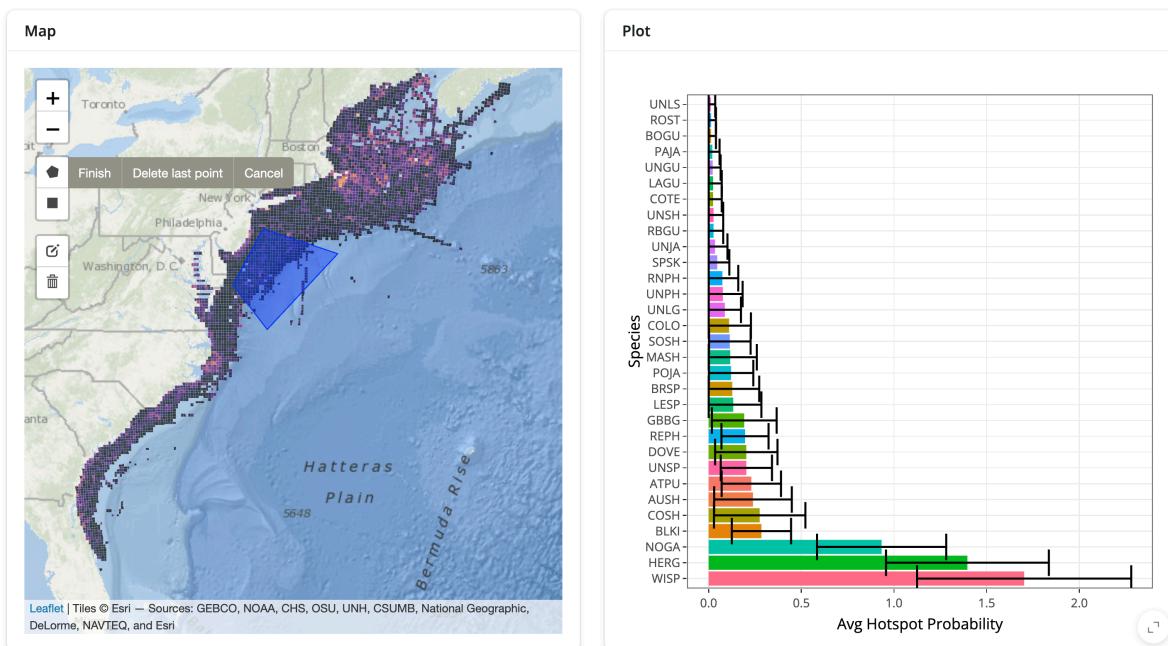
The screenshot shows the 'Area Explorer' application interface. On the left, there's a sidebar titled 'Areas of Interest' with a dropdown menu set to 'Regions'. Below it, a text area displays the coordinates of a selected polygon: 'last edited: POLYGON((-141.0639 46.55907, -136.6324 42.0423, -132.0404 48.99068, -135.5994 49.97881, -141.0639 46.55907))'. The main right panel is a map of North America with a blue polygon highlighting the Aleutian Arc region off the coast of Alaska. A tooltip window over the polygon provides the following data:

aoi	shlf_key	shlf_name	rgn_key	rgn_name	AK	Alaska	ALA
				Aleutian Arc			
	ctr_lon	-178.56252307137368					
	ctr_lat	51.149538291898025					
	area_km2	860518.1237176519					
	geometry				null		

Bird Hotspots

Bird hotspots application showing hotspot probability for species present given drawn Area of Interest.

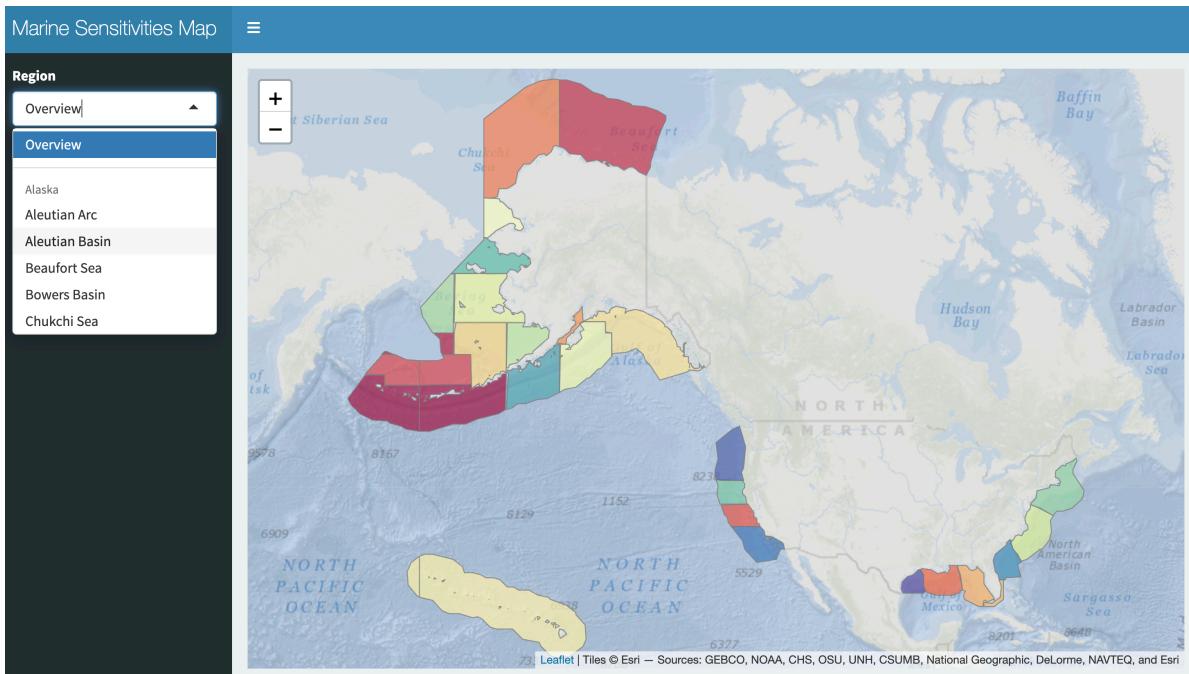
- [🌐 website](#)
- [📄 code](#)



Regional Map

Basic interactive map of BOEM regions.

- [🌐 website](#)
- [🔗 code](#)

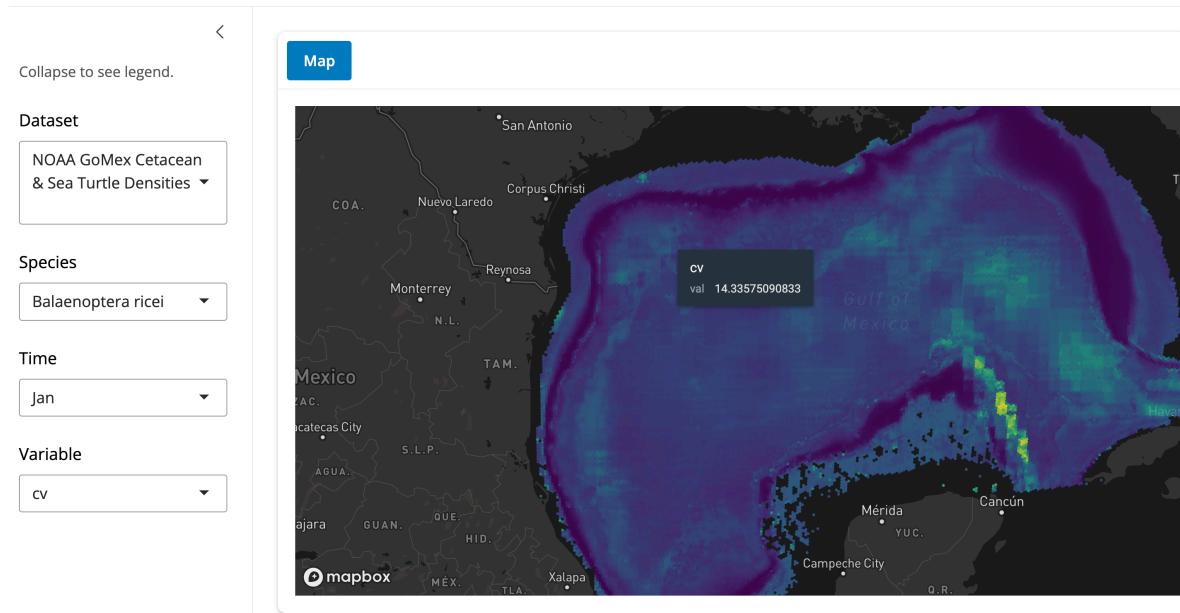


Distributions, Vector

Show species distributions with high resolution vectors (as vector tiles).

- [website](#)
- [code](#)

SDM Explorer

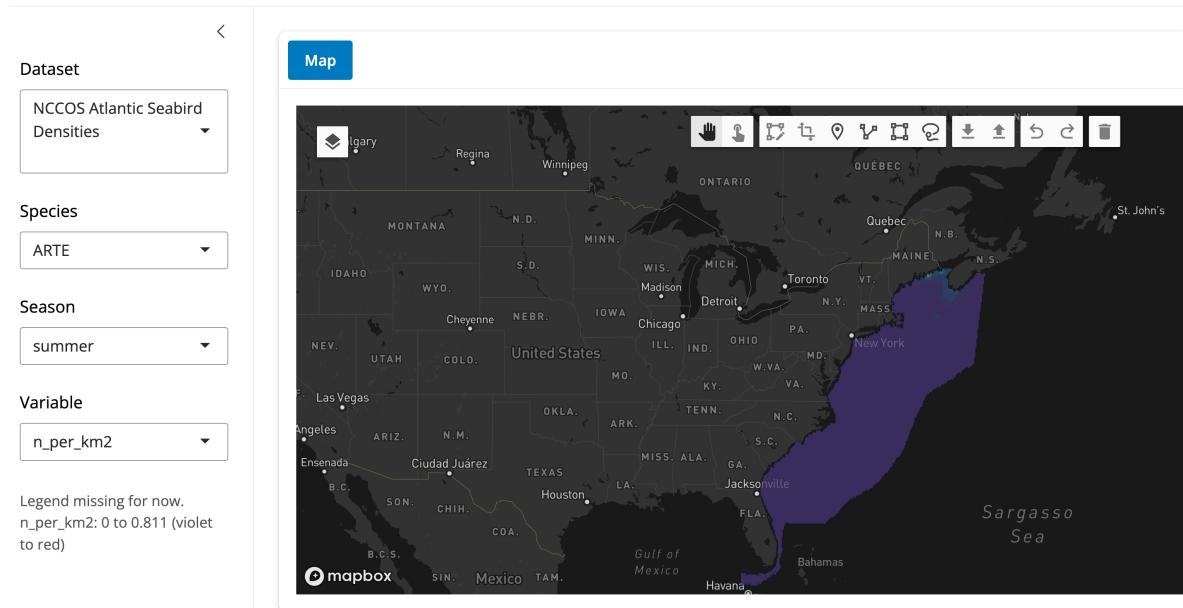


Distributions, Raster

Show species distributions with high resolution rasters as cloud-optimized GeoTIFFs (COGs).

- [🌐 website](#)
- [📄 code](#)

SDM Raster Explorer



Vulnerability Mapper

Combine species distribution models (raster) and vulnerability metrics (tables) to identify areas of high conservation concern.

- [!\[\]\(6a8b31c5847cdbbc60180bc0d7c3f345_img.jpg\) website](#)
- [!\[\]\(472ec9e3fcfea6a0fb3a3dccb4169d19_img.jpg\) code](#)

