Ocean Modeling Trends and Data Driven Modeling with Julia

JuliaEO25, Terceira, Azores

Gaël FORGET, 2025/01/09

Outline

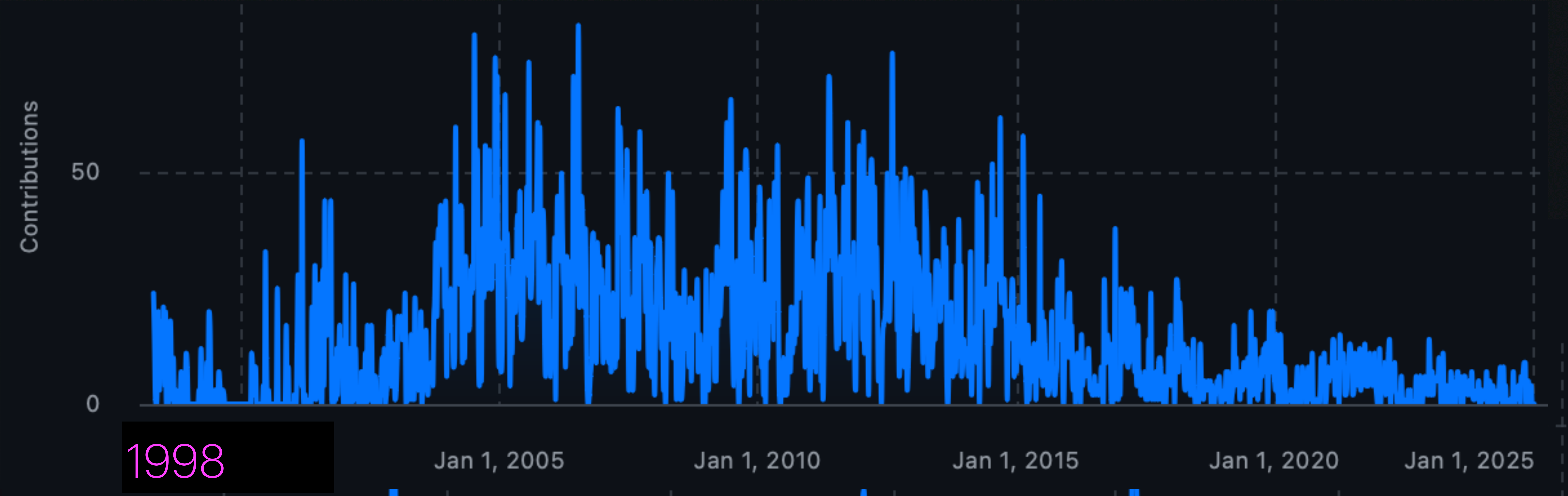
- General Circulation Modeling
- Data Driven Ocean Modeling
- Geospatial & Analysis Perspective
- Bridging Communities
- Today’s session

General Circulation Modeling at MIT

(See also Simone's presentation on ClimaOcean.jl)

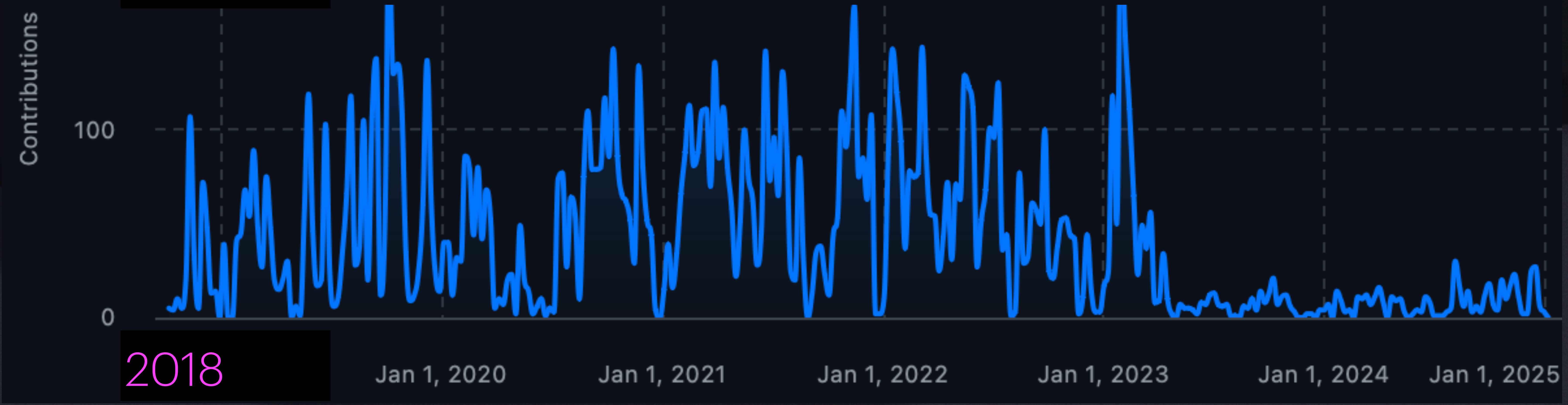
MITgcm

(19.9k commits)



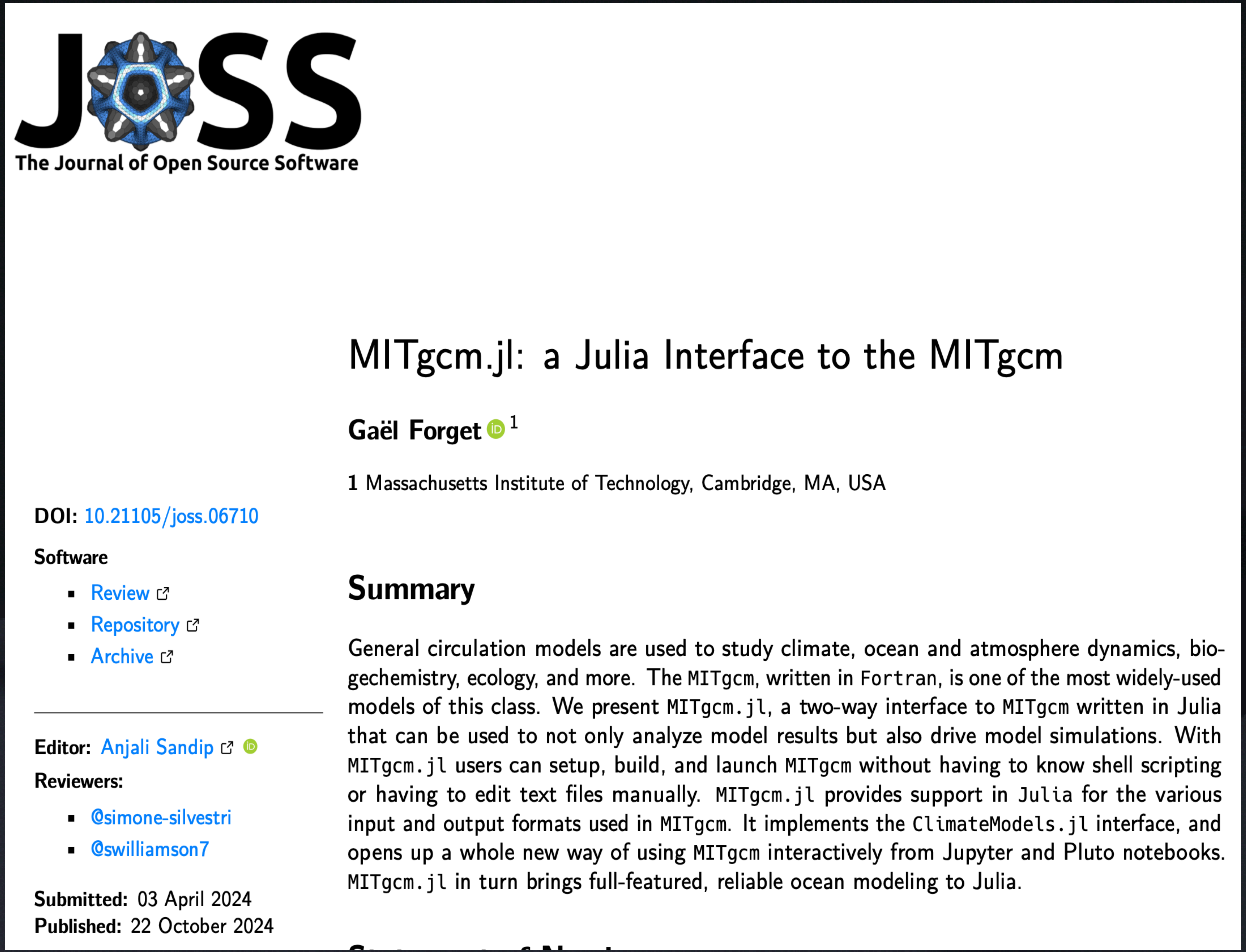
Oceananigans.jl

(12.6k commits)



MITgcm.jl and ClimateModels.jl

(See my later presentation)



JOSS
The Journal of Open Source Software

MITgcm.jl: a Julia Interface to the MITgcm

Gaël Forget[1]

1 Massachusetts Institute of Technology, Cambridge, MA, USA

DOI: 10.21105/joss.06710

Software

- Review
- Repository
- Archive

Editor: Anjali Sandip

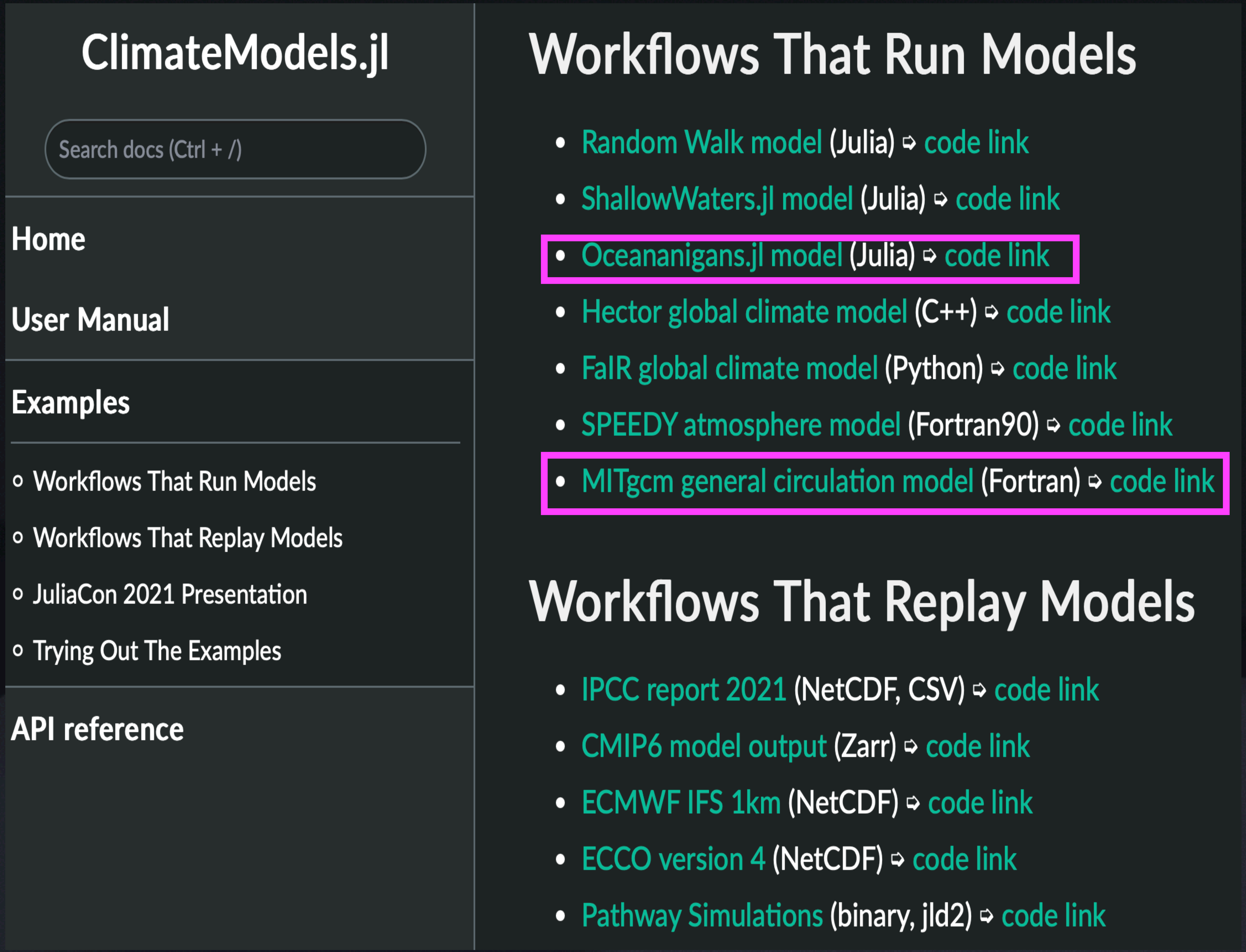
Reviewers:

- @simone-silvestri
- @swilliamson7

Submitted: 03 April 2024
Published: 22 October 2024

Summary

General circulation models are used to study climate, ocean and atmosphere dynamics, biogechemistry, ecology, and more. The MITgcm, written in Fortran, is one of the most widely-used models of this class. We present MITgcm.jl, a two-way interface to MITgcm written in Julia that can be used to not only analyze model results but also drive model simulations. With MITgcm.jl users can setup, build, and launch MITgcm without having to know shell scripting or having to edit text files manually. MITgcm.jl provides support in Julia for the various input and output formats used in MITgcm. It implements the ClimateModels.jl interface, and opens up a whole new way of using MITgcm interactively from Jupyter and Pluto notebooks. MITgcm.jl in turn brings full-featured, reliable ocean modeling to Julia.



Interactivity Provided by Julia

e.g. Pluto reactive notebook examples

Modify Parameters

Edit or **run** this notebook

First, select a model parameter group (or the default):

data

PARM01

```
▸OrderedDict(:rigidLid ⇒ false, :implicitFreeSurface ⇒ true,
```

Then, enter parameter name (without ":") **and new value:**

parameter name 👉 👈

new value 👉 👈

Rerun Model

Once ready, click `Update & Relaunch` **to:**

- update parameter file
- rerun the model
- update the plots

Update & Relaunch

Table of Contents

e.g. Julia REPL

```
using MITgcm
params=read_toml(:ECCO4)
MC=MITgcm_config(inputs=params)
run(MC)
```

Drifters.jl

(See my later presentation)

The Journal of Open Source Software

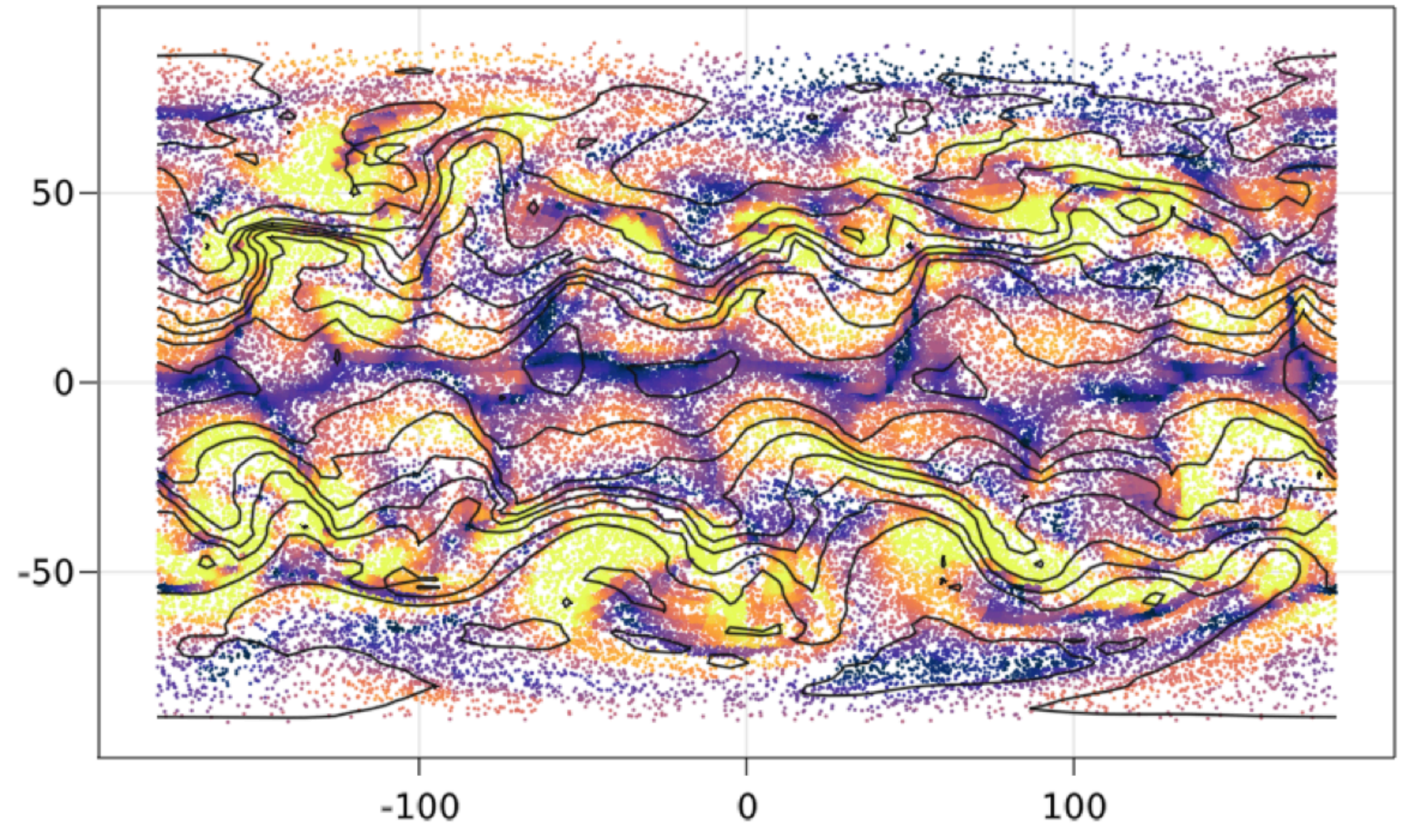
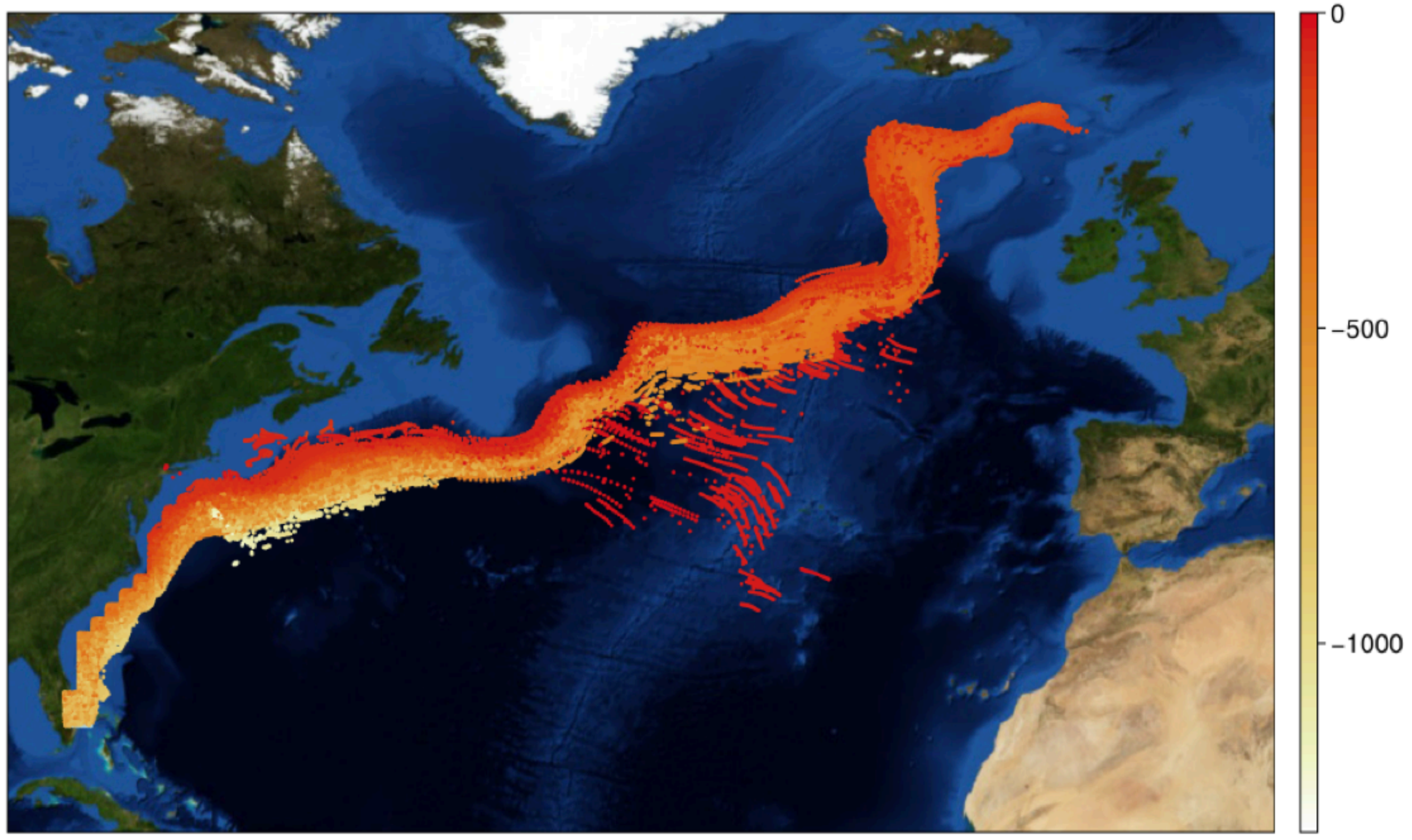


Figure 1: Atmosphere simulation example. Hundred thousand particles are displaced by time-varying 2D flow fields provided by MITgcm on a cube-sphere grid (Campin et al., 2020) using `MITgcmTools.jl`. Particle colors show velocity while contours show temperature.



3D particle tracking along the Gulf Stream, computed by Drifters.jl (Gaël Forget, 2021)

Ocean Biogeochemistry Models

- https://github.com/JuliaOcean/PlanktonIndividuals.jl
- https://github.com/JuliaOcean/AIBECS.jl
- https://github.com/OceanBioME/OceanBioME.jl
- …

Darwin3 Model (1D)

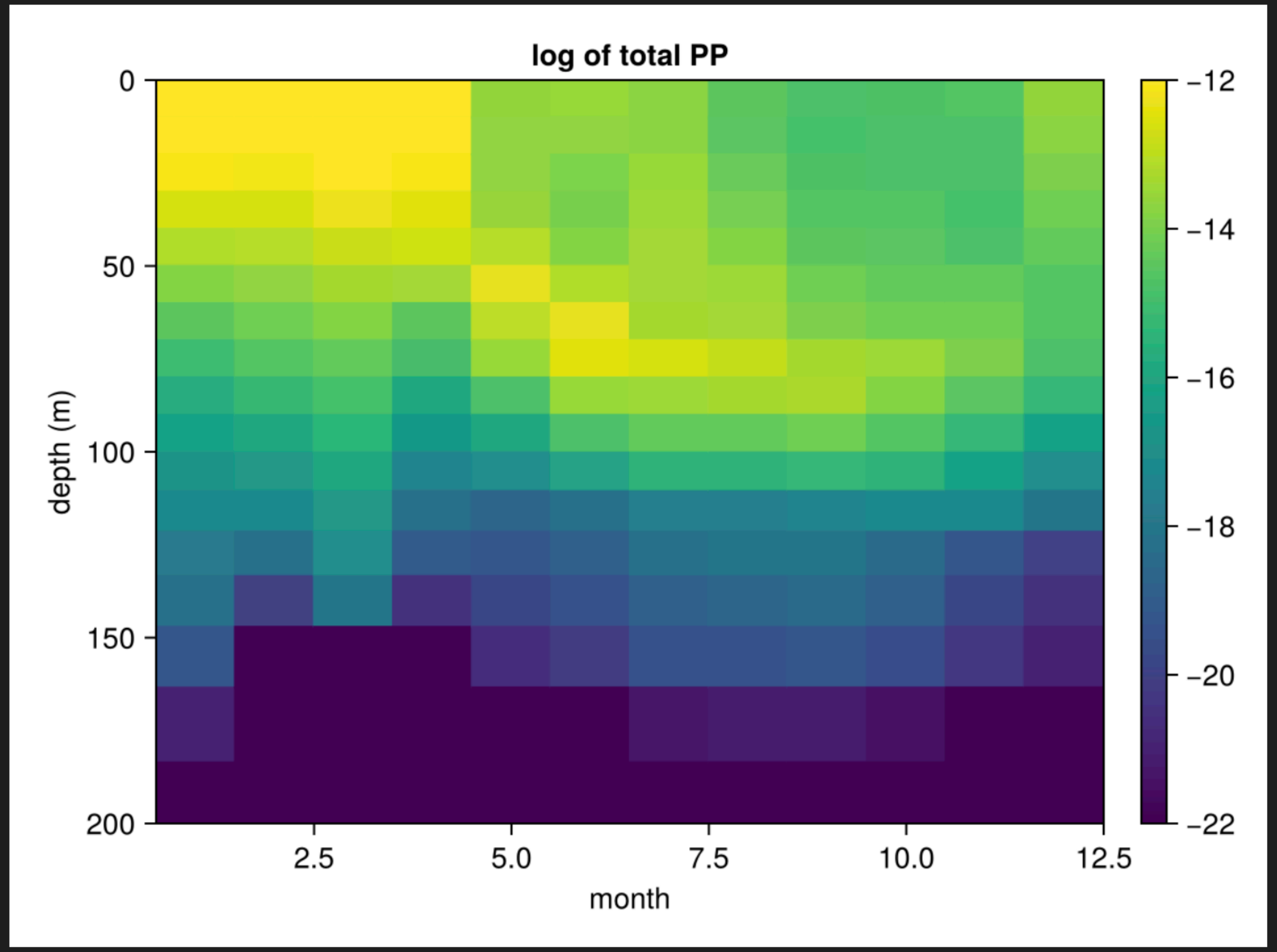
Contributors:

- Stephanie Dutkiewicz provided the examples
- Oliver Jahn and collaborators the Darwin3 source code
- Gaël Forget the MITgcm.jl interface and notebook prototype

Note

For more information, please refer to

- darwin3.readthedocs
- darwinpropject.mit.edu
- cbiomes.org
- MITgcm.readthedocs
- MITgcm.jl.



Outline

- General Circulation Modeling
- Data Driven Ocean Modeling
- The Geo & Analysis Perspective
- Bridging Communities
- Today's session

Pre-Julia data driven ocean modeling

ECCO4 , a view from 2015 (and still today)

- **Ocean reanalysis** : ECCO (JPL) / OCCA (MIT)
- **Ocean model** : MITgcm & its adjoint, at 1 degree
- **Period** : 1992-2017 (ECCO4), 1960-present (OCCA2)

$$J(\mathfrak{u}) = \sum_i \alpha_{\mathrm{i}} \times \left(d_i^T\, \mathbf{R}_i^{-1}\, d_i\right) + \sum_j \beta_j \times \left(\mathfrak{u}_j^T \mathfrak{u}_j\right)$$

$$d_i = \mathcal{P}(m_i - o_i),$$

- **Data constraints** : temperature and salinity profiles from the field, multi-decadal satellite data record (*SST, SSS, SLA, BP, sea-ice*), atmospheric reanalysis for surface boundary conditions (*MERRA2 or ERA5 currently*), plus river runoff (*Fekete et al*)
- **Optimized parameters** : atmospheric state variables, surface wind stress, sub-grid scale mixing rates, meso-scale transport rates, initial ocean state

ECCO4 (... ECCO.jl)

(See also Alex's presentation today, JuliaSim's on Wed, ...)

optimize over :

Table 7. Control parameters that have been adjusted as part of the state estimation.

Description	Frequency	Size
Initial condition for temperature	N/A	2.4×10^6
Initial condition for salinity	N/A	2.4×10^6
Diapycnal diffusivity	Time mean	2.4×10^6
Isopycnal diffusivity	Time mean	2.4×10^6
GM intensity	Time mean	2.4×10^6
Atmospheric temperature at 2 m	Bi-weekly	3.2×10^7
Specific humidity at 2 m	Bi-weekly	3.2×10^7
Precipitation	Bi-weekly	3.2×10^7
Downward longwave radiation	Bi-weekly	3.2×10^7
Downward shortwave radiation	Bi-weekly	3.2×10^7
Zonal wind stress	Bi-weekly	3.1×10^7
Meridional wind stress	Bi-weekly	3.1×10^7

MITgcm

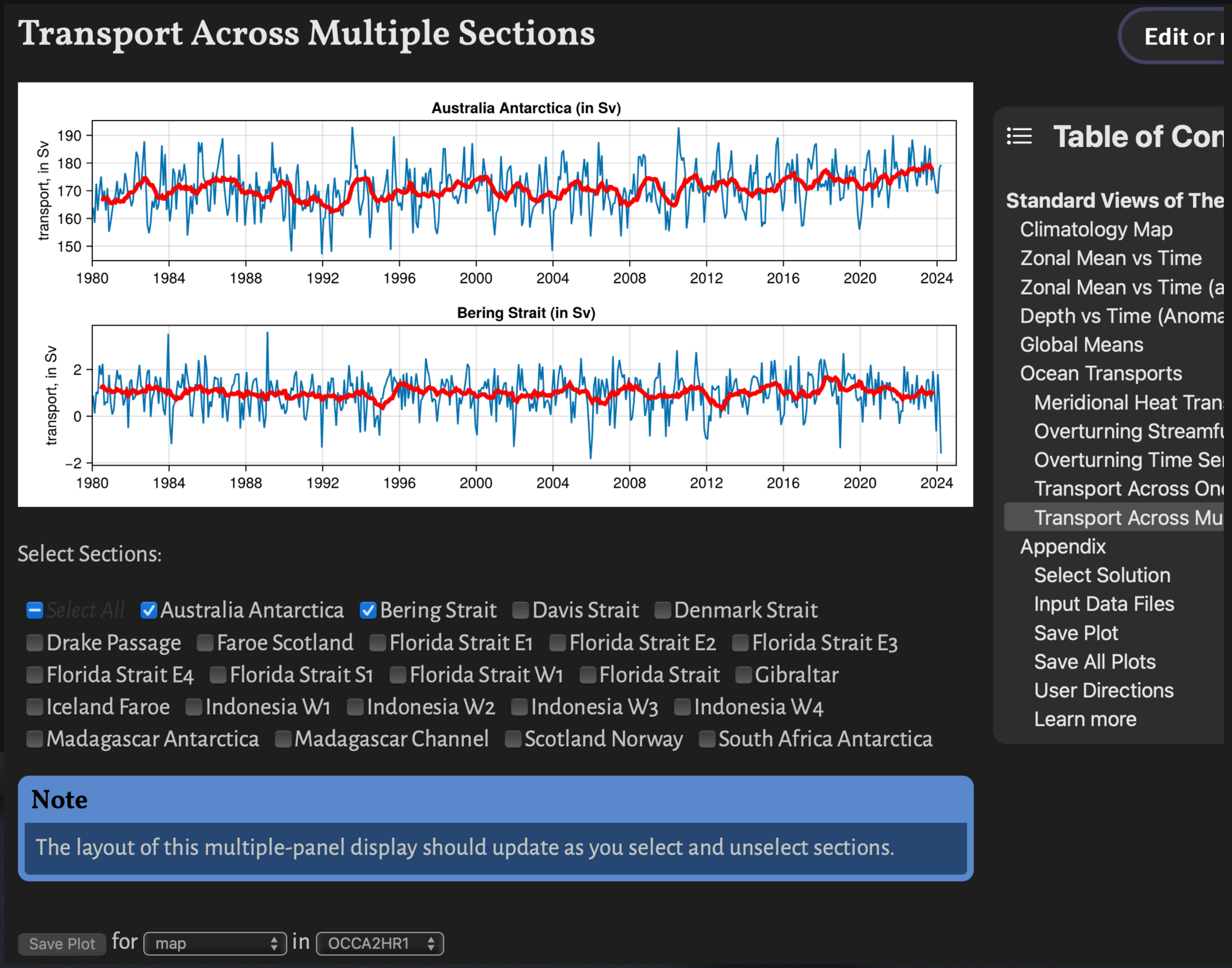
MITgcm adjoint

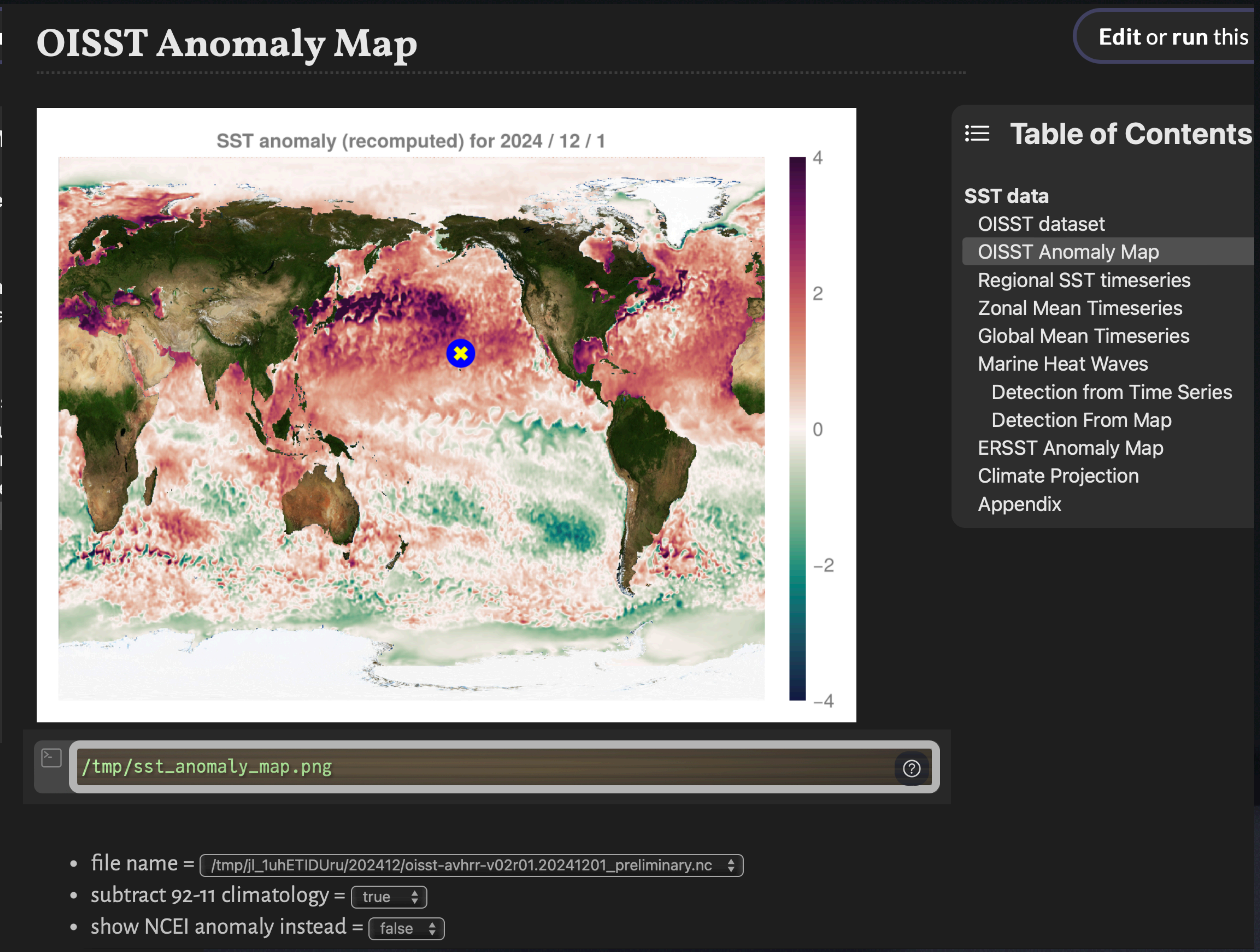
(backpropagation)

Data set	T profiles	S profiles	Origin
Argo	833 033	800 269	IFREMER
CTD	379 012	333 266	NODC, WOA09
XBT	597 009	0	NODC, WOA09
ITP	18 033	17 745	Toole et al. (2011)
SEaOS	103 117	87 806	Roquet et al. (2011)

Variable	Description	Period	Size
MDT	DNSC08 mean SSH minus EGM2008 geoid model	1993–2004	6.2×10^4
T, S	Blended monthly climatology		$2 \times 5.7 \times 10^8$
	OCCA	2004–2006	
	WOA 2005	Unclear	
	PHC 3.0	Unclear	
SLA	Daily bin average of along-track altimetry	1992–2011	7.7×10^7
SST	Monthly maps	1992–2011	1.5×10^7
ICF	Monthly maps	1992–2010	1.4×10^7

Gridded Datasets





Example notebooks from Climatology.jl

OceanRobots.jl

juliacon

Digital Twins for Ocean Robots

Gaël Forget[1]

[1]Earth, Atmospheric and Planetary Sciences, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, 02139, MA, USA

ABSTRACT

DTOR is a framework to access, analyze, and simulate the global fleet of ocean observing devices (or *ocean robots*) that monitor climate change. It brings these observations to Julia and lets us pair ocean robots with virtual counterparts (or *twins*). Digital twins provide a bridge from the data to predictive models and enable machine learning. In turn, observing system simulations in a digital environment can help evaluate observational strategies a priori, during deployment, or afterwards. In this paper we present the DTOR framework, its supported observing systems, capabilities to simulate ocean robots, and envisioned applications.

Keywords

Julia, Ocean, Climate, Robot, Observation, Observing, Platform, Sensor, Drifter, Buoy, Profiler, Model, Modeling, Machine Learning, Artificial Intelligence, Data Assimilation, Parameter Inference, Climatology, Geospatial, Statistics

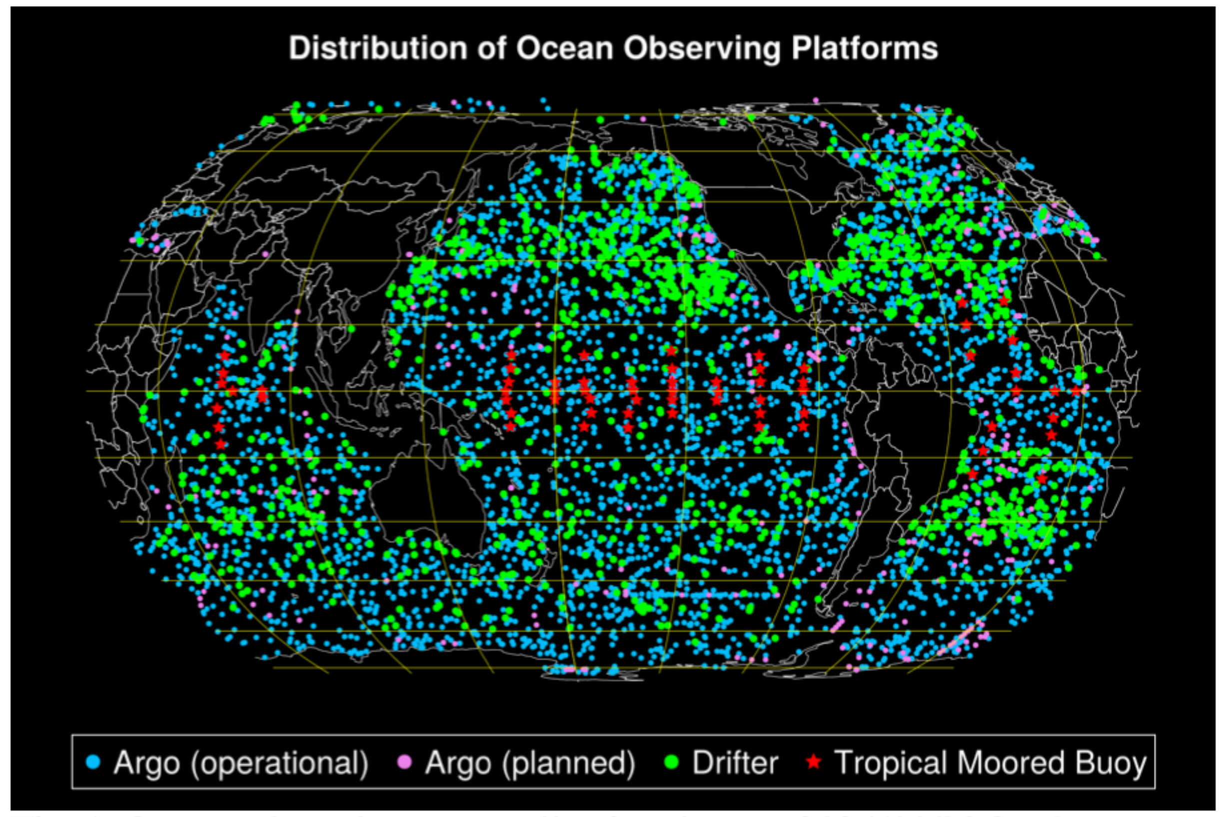


Fig. 1: Ocean robots that were collecting data on *2024/11/29* for three types of observations (blue, green, and red dots). Planned deployments of Argo profilers are also shown (purple dots). Data is from https://www.ocean-ops.org and queried via OCEANROBOTS.JL.

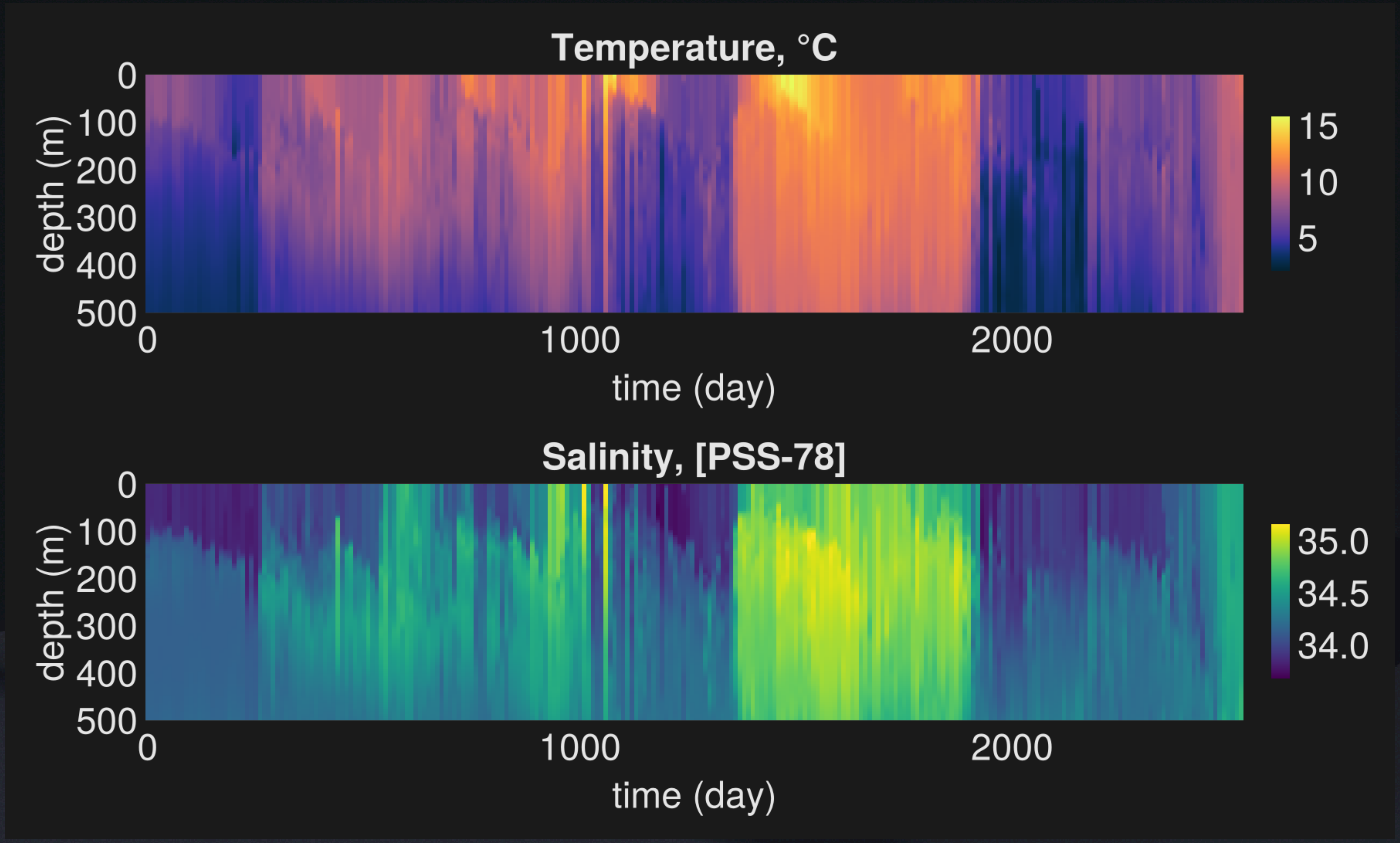
Table 1. : Ocean observing platforms targeted by OCEANROBOTS.JL, and associated data structures (or blank if not yet implemented).

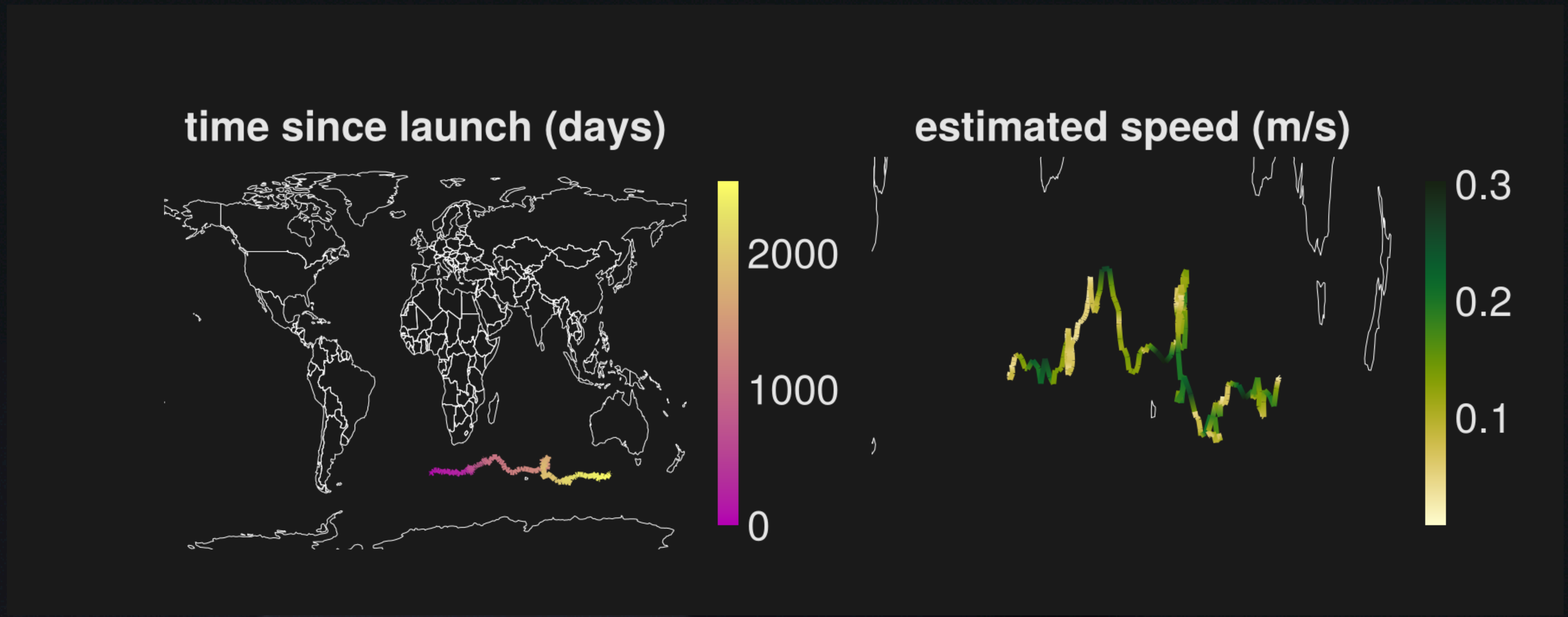
Platform Type	Data Structure
surface drifters	*SurfaceDrifter*, *CloudDrift*
drifting profilers	*ArgoFloat*
moored buoys	*OceanSite*, *NOAAbuoy*
sea gliders	*Gliders*
expendable bathythermographs	
sail drones	
research vessel data	CCHDO
ships of opportunity	
orbiting satellites	*SeaLevelAnomaly*
suborbital flights	
marine mammals	

OceanRobots.jl : List of supported observing networks + those planned

Example 1 : Argo Floats

Building a simple Julia interface to observations in DTOR



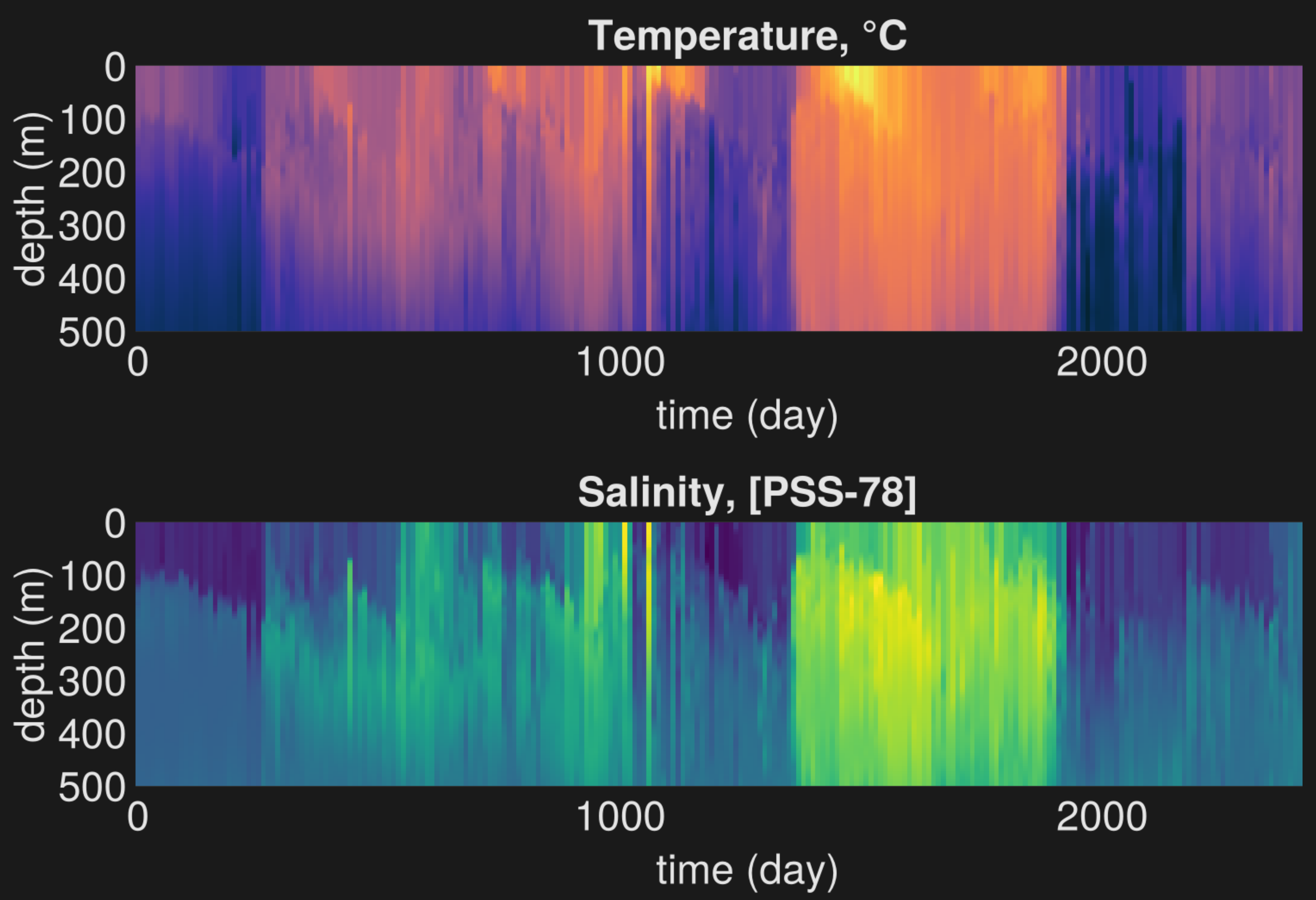


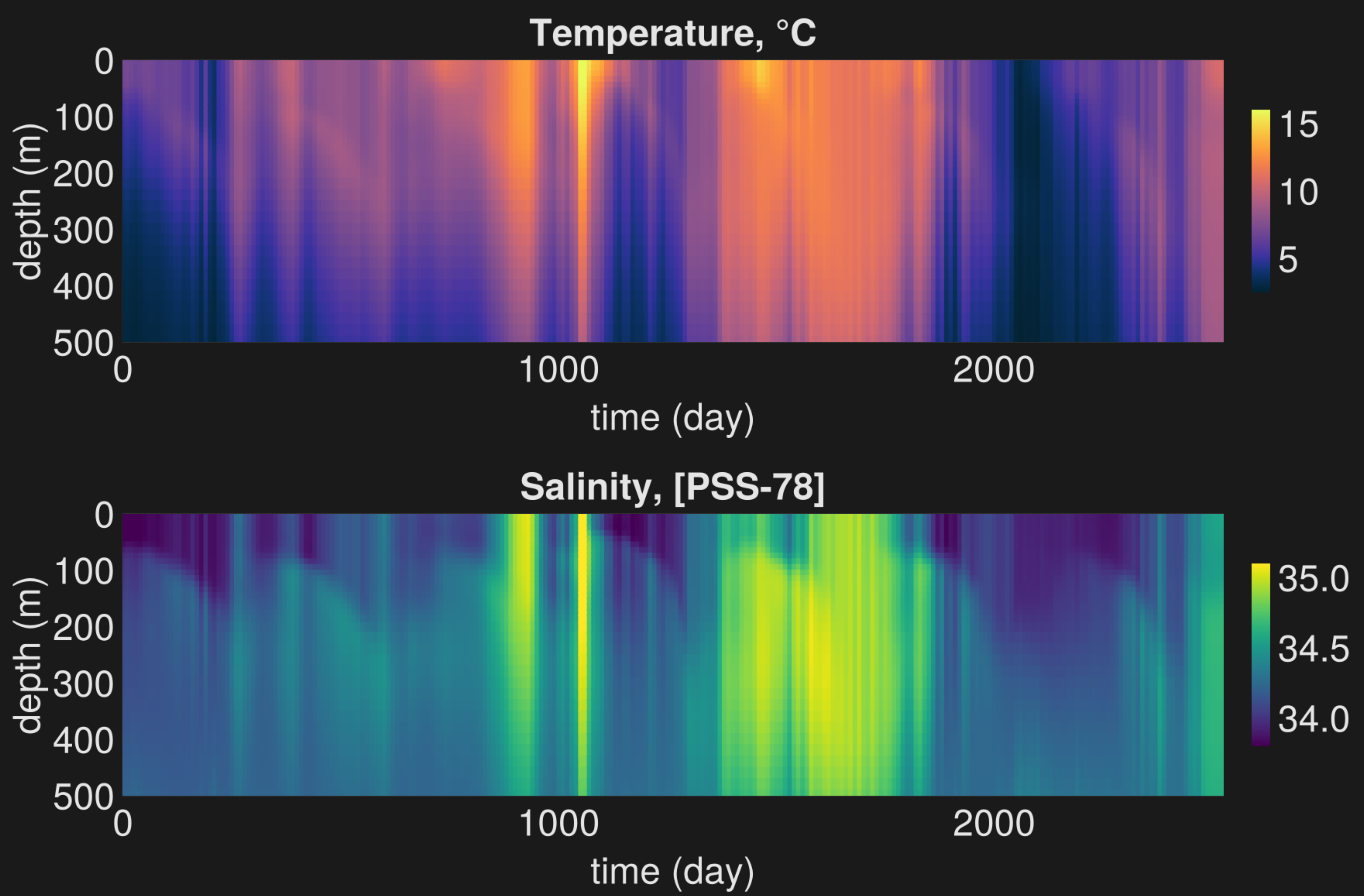
Code 1: Download and visualize one Argo float data as in Fig. 4.

```
using OceanRobots, ArgoData, CairoMakie
argo=read(ArgoFloat(),wmo=6900900)
fig=plot(argo,option=:standard)
```

Example 1 : Argo Floats

Pairing Physical and Virtual Twins



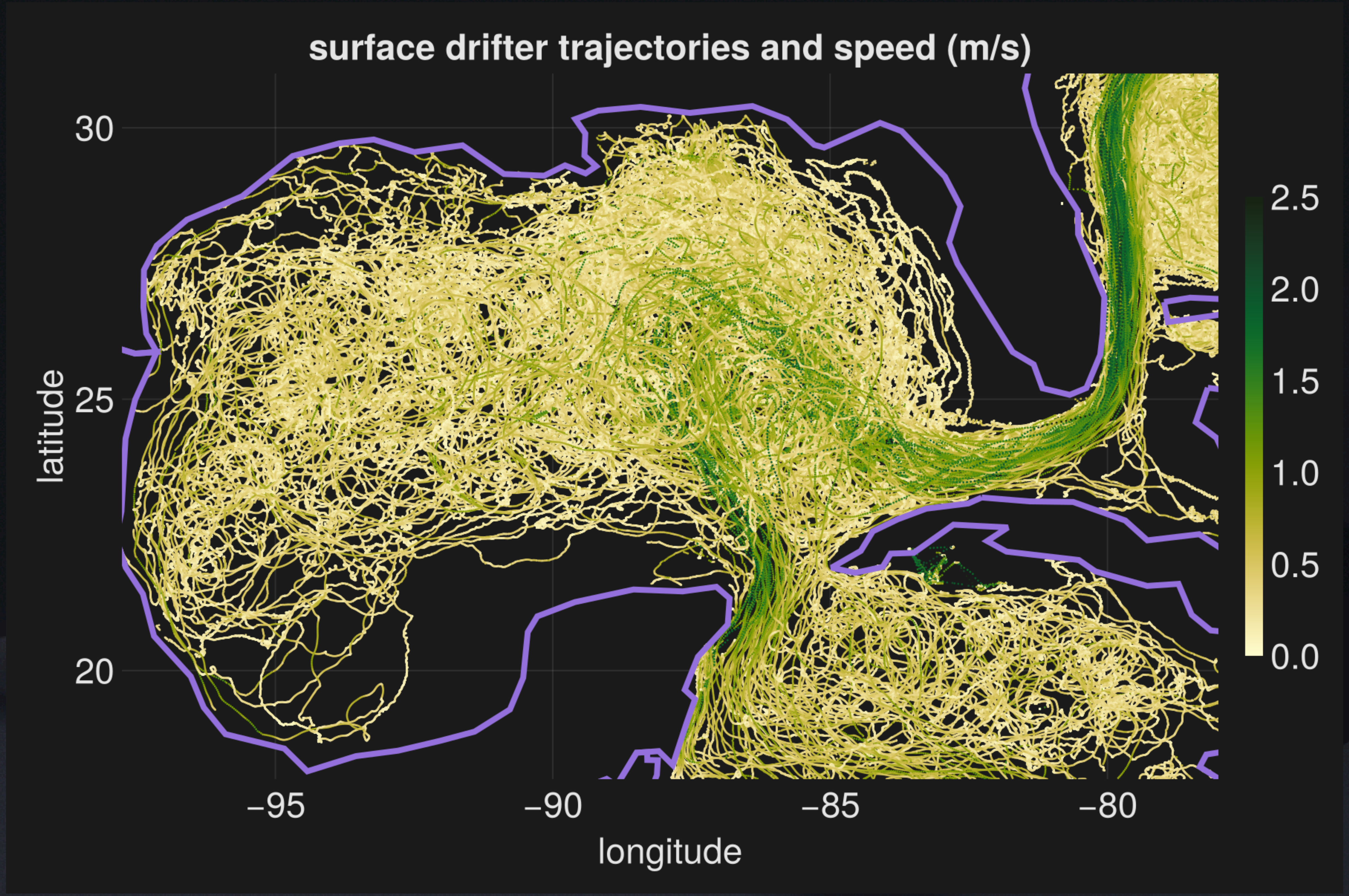


Argo (real world profiles)

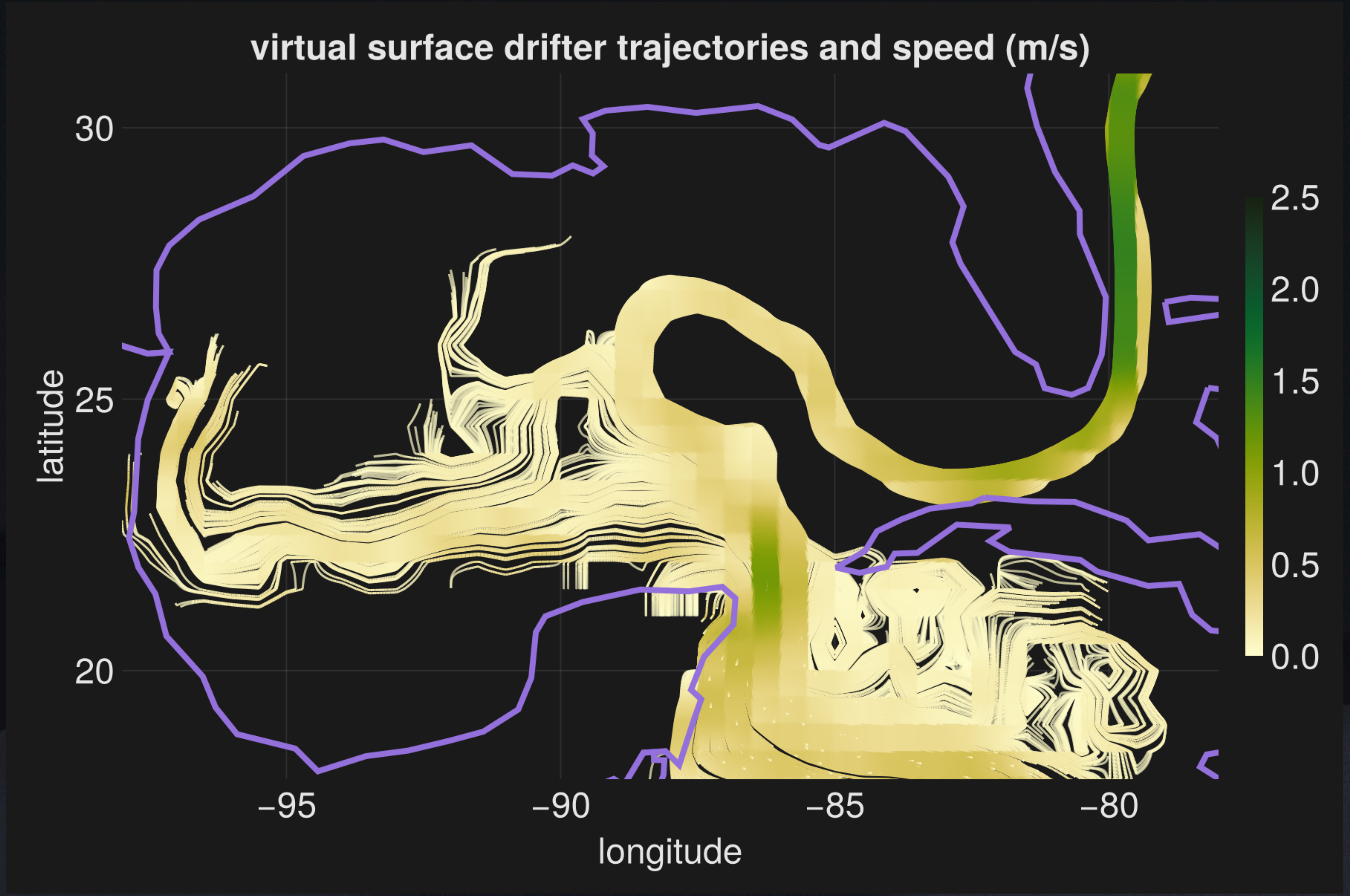
Climatology.jl (emulated profiles)

Example 2 : Drifters

Pairing Physical and Virtual Twins



Drifters (real world trajectories)



Drifters.jl (basic emulator)

Example 3 : Biogeochemical-Argo

Building Emulators & Assessing Performance

- Blue : model truth
- Black : neural emulators
- Software : Flux.jl, MITgcm.jl , Makie.jl

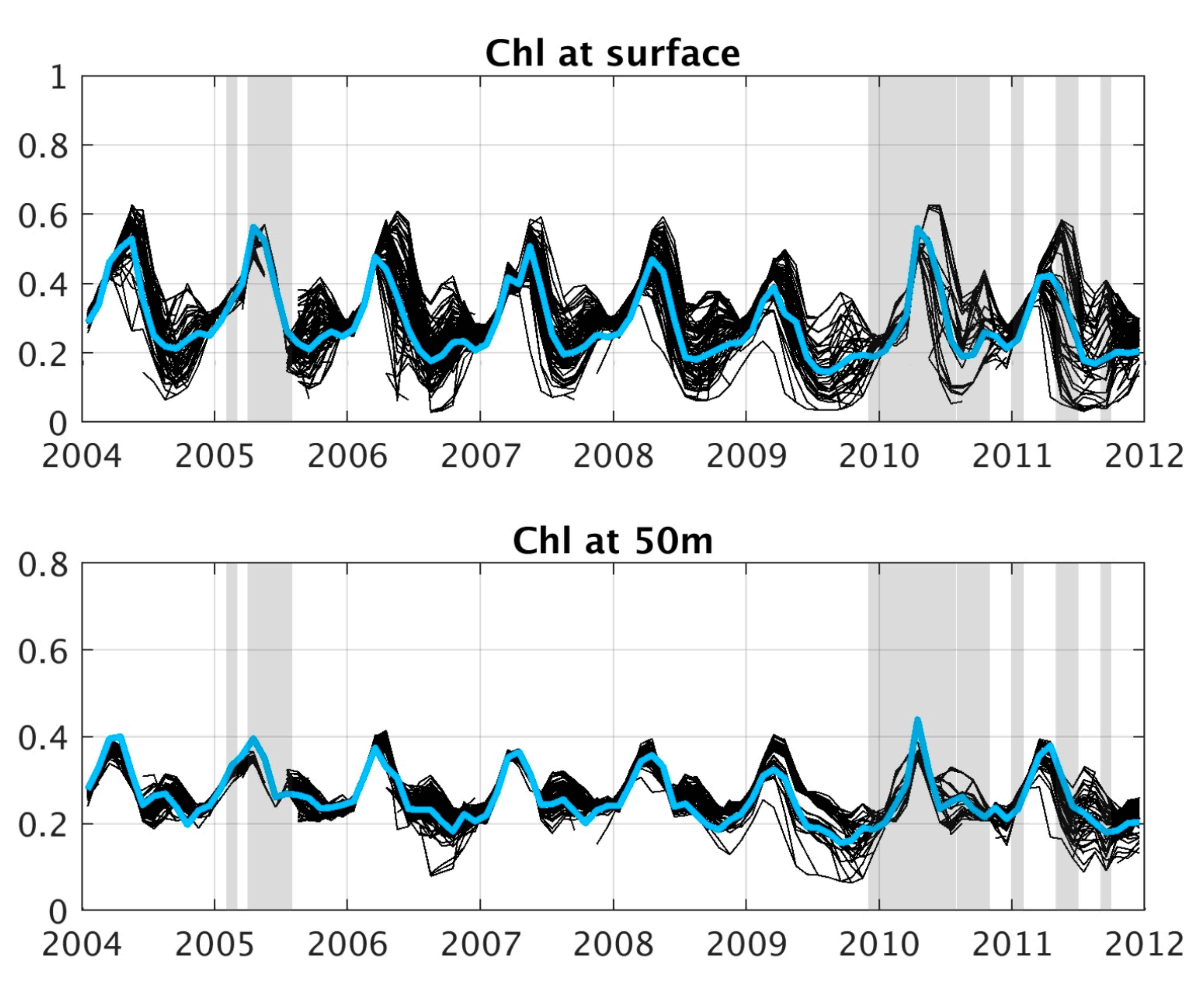


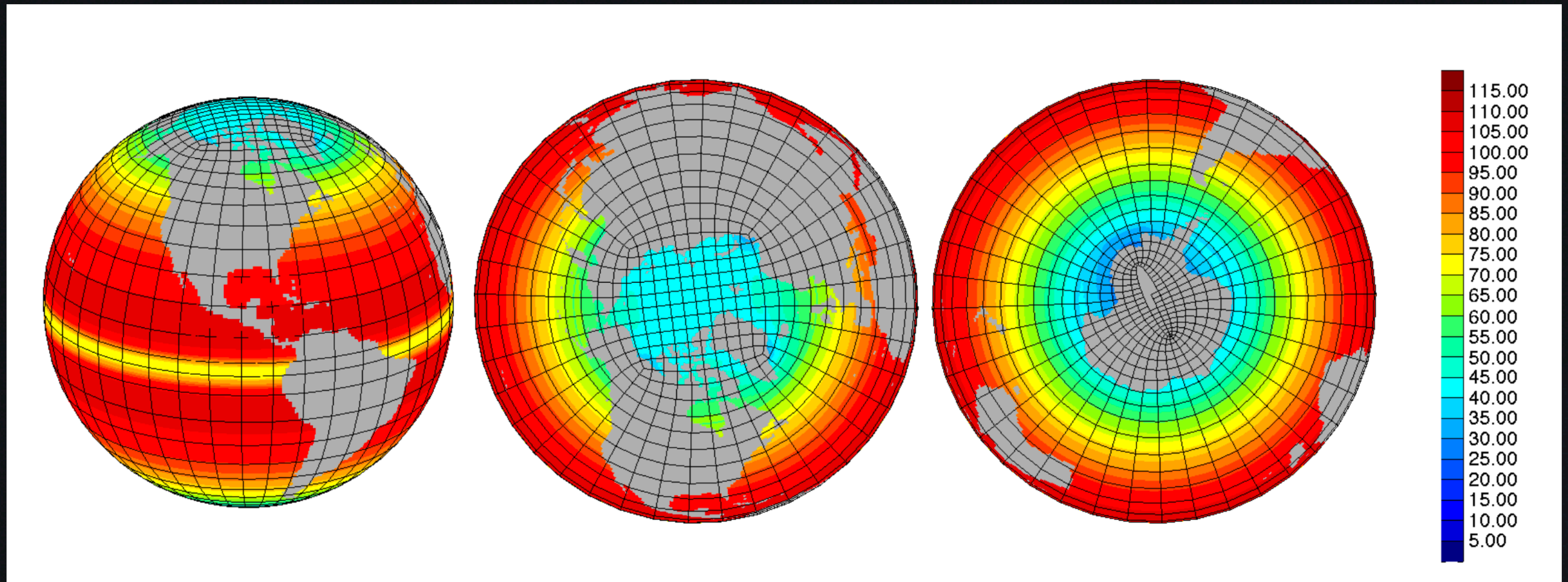
Fig. 9: Ensemble of model predictions (black curves) by a simple multi-layer perceptron (from Flux.jl) trained to predict Chlorophyll concentration (present in green algea and marine microbes) from environmental variables (T, S, but also oxygen, optical backscatter, and solar radiation). The blue line is the *ground truth* that we seek to estimate, and was obtained through proper spatial averaging of the gridded data set from which the training data itself was generated.

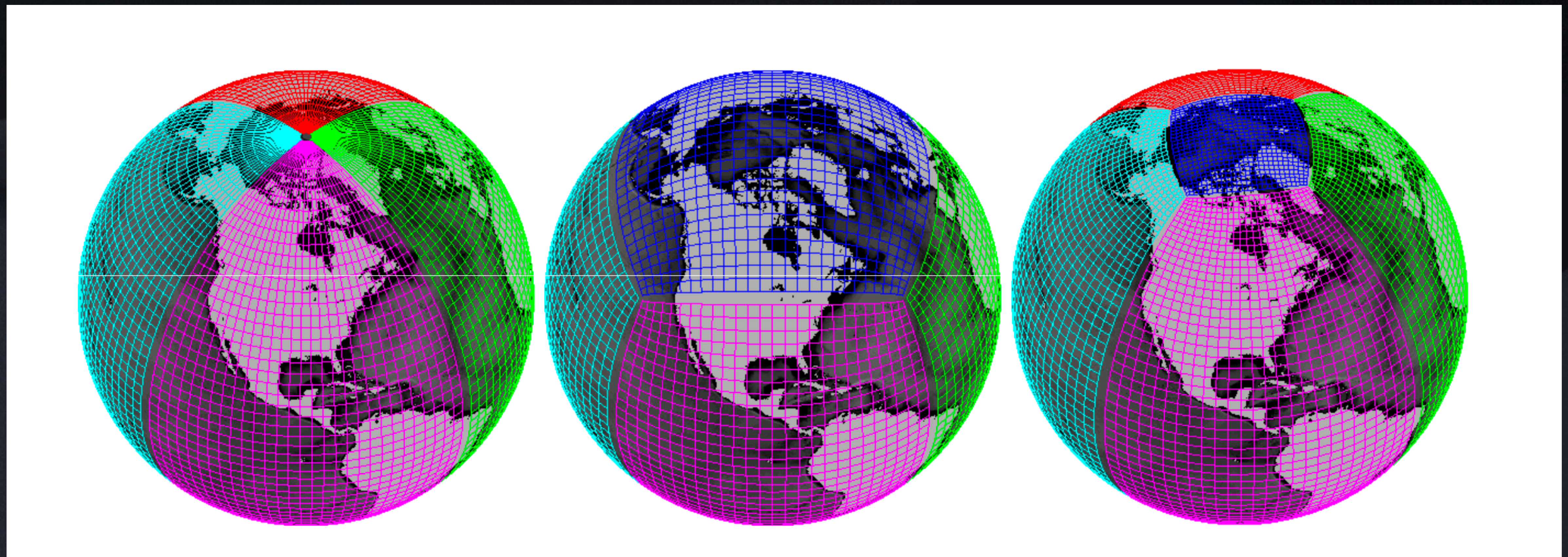
Outline

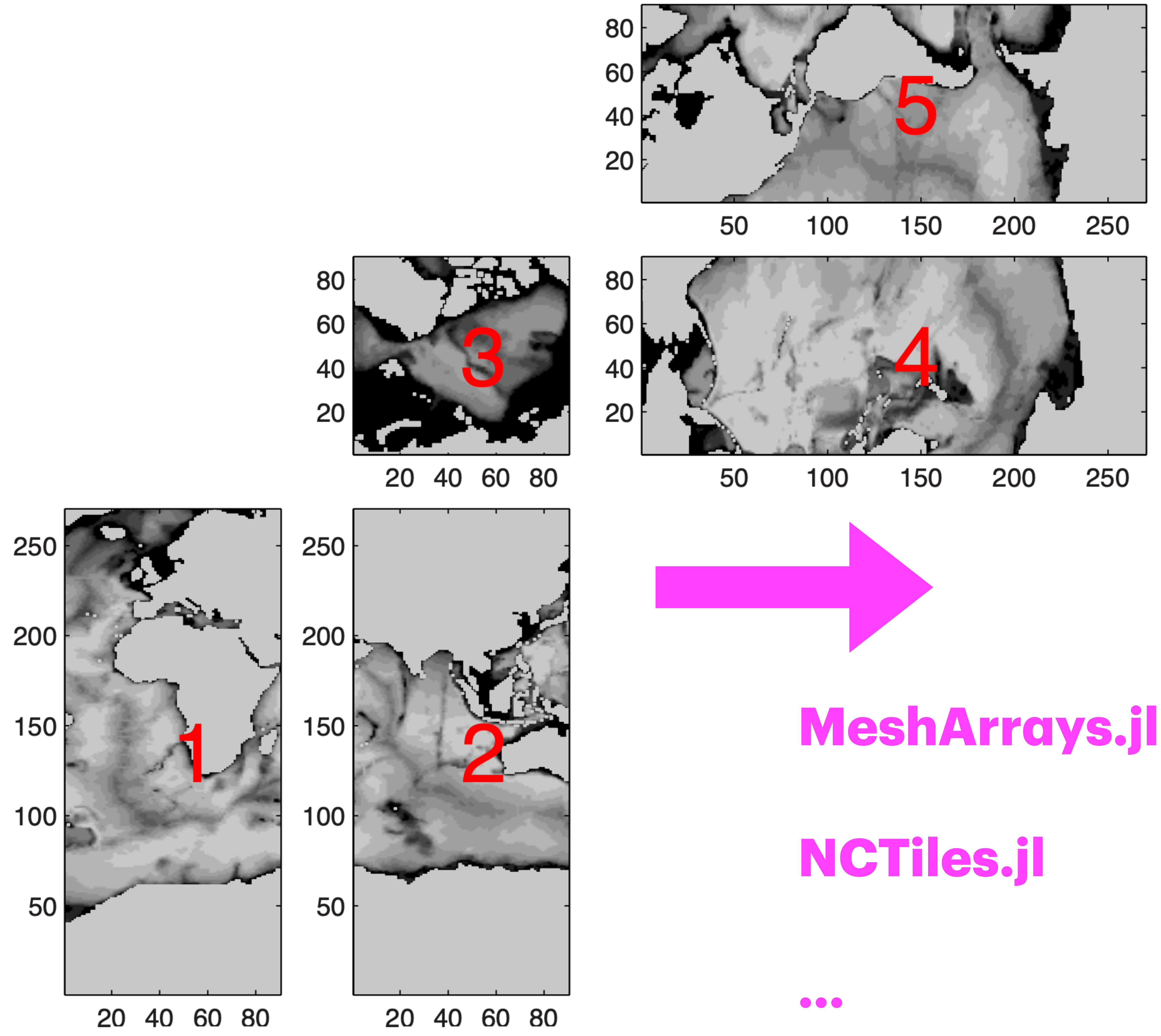
- General Circulation Modeling
- Data Driven Ocean Modeling
- Geospatial & Analysis Perspective
- Bridging Communities
- Today's session

Global Model Grids

(See also Milan's SpeedyWeather.jl)







MeshArrays.jl

NCTiles.jl

...

MeshArrays.jl

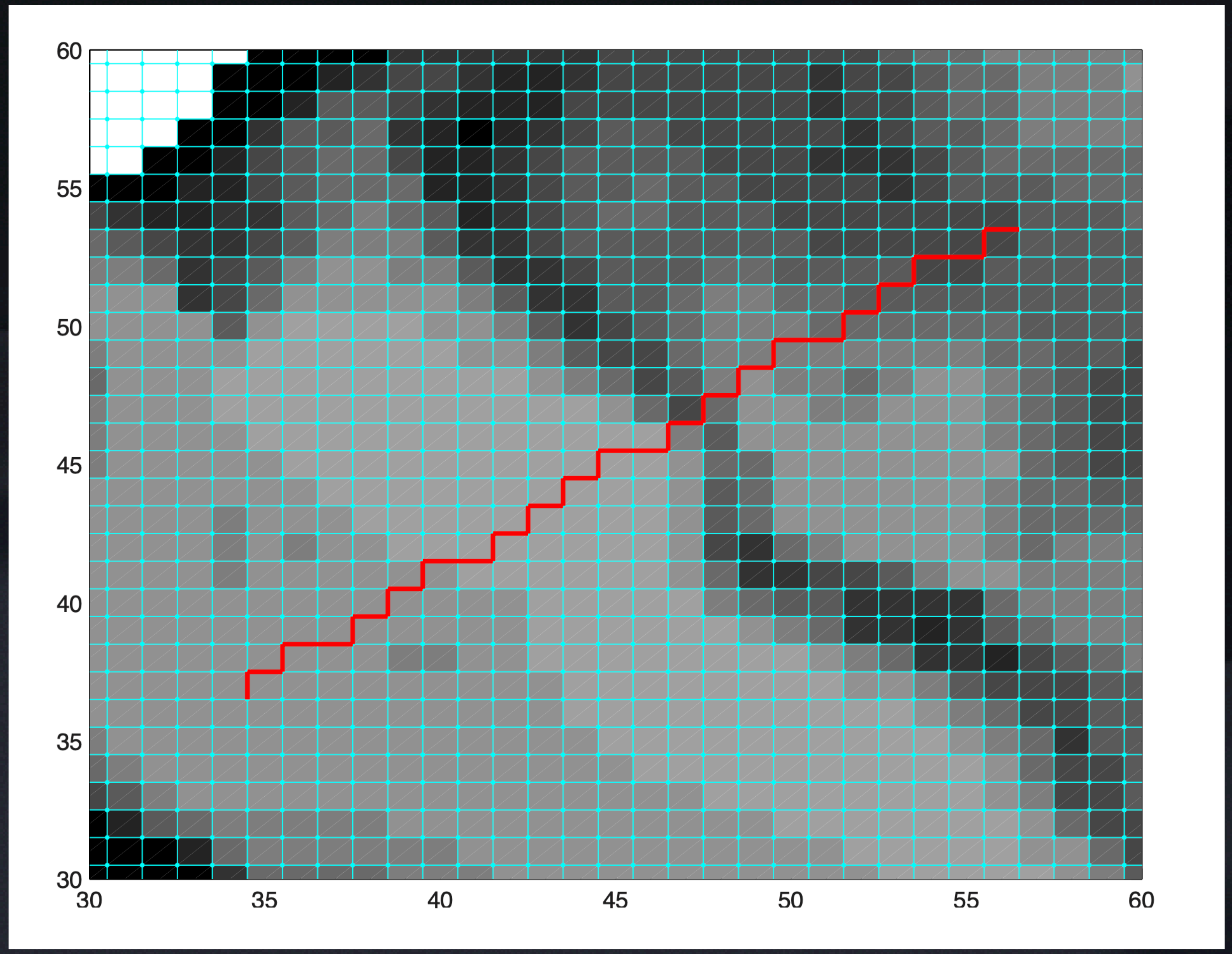
Main Features

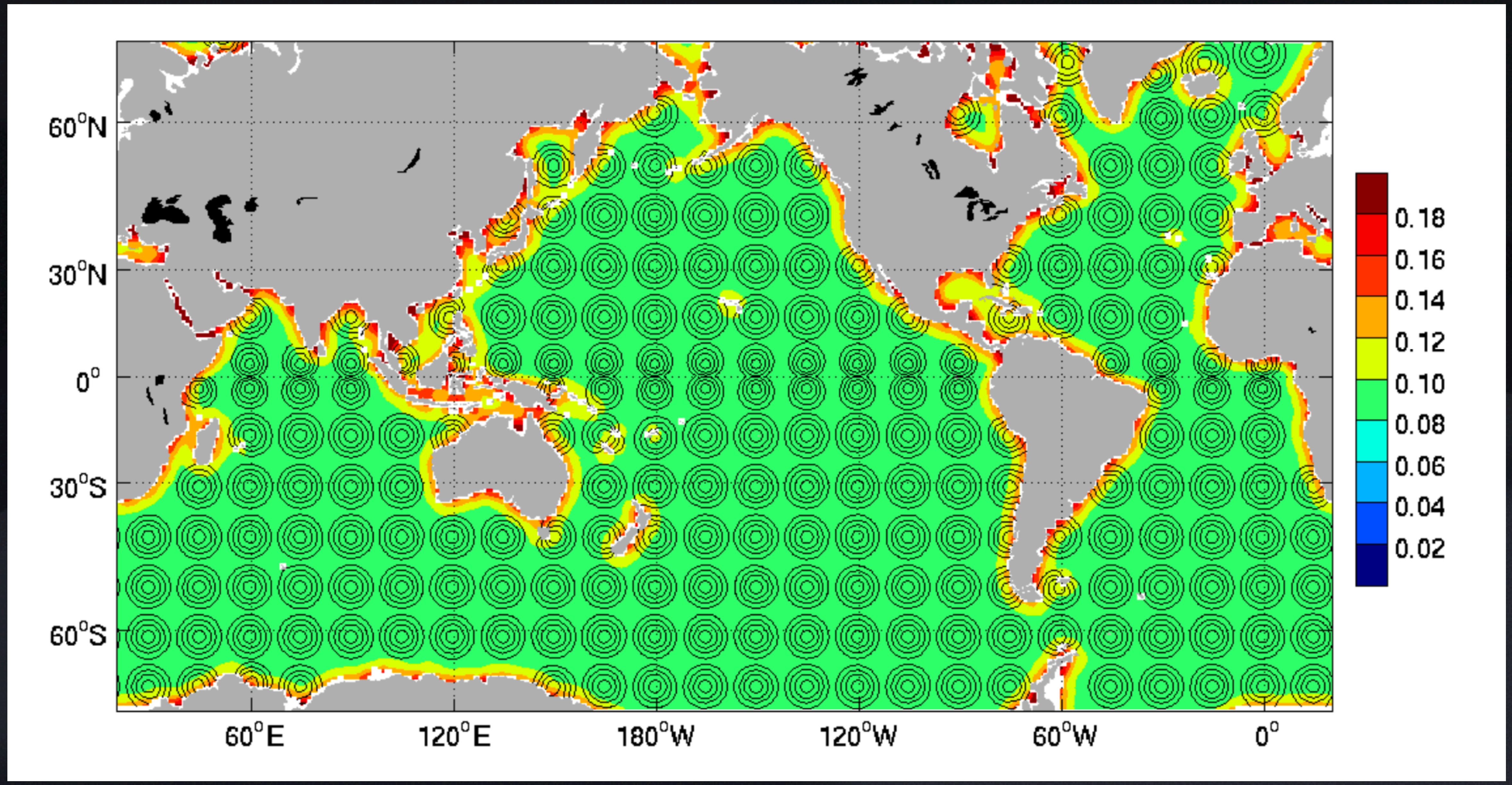
- `MeshArray`, `gcmgrid`, `varmeta`
- full Earth grid examples (C-grids)
- vector fields, transports, budgets
- interpolation, distances, collocation
- visualization (via Makie extension)
- particle tracking via IndividualDisplacements.jl

Interpolation Scheme

The Interpolation Code? Let's Break It Down:

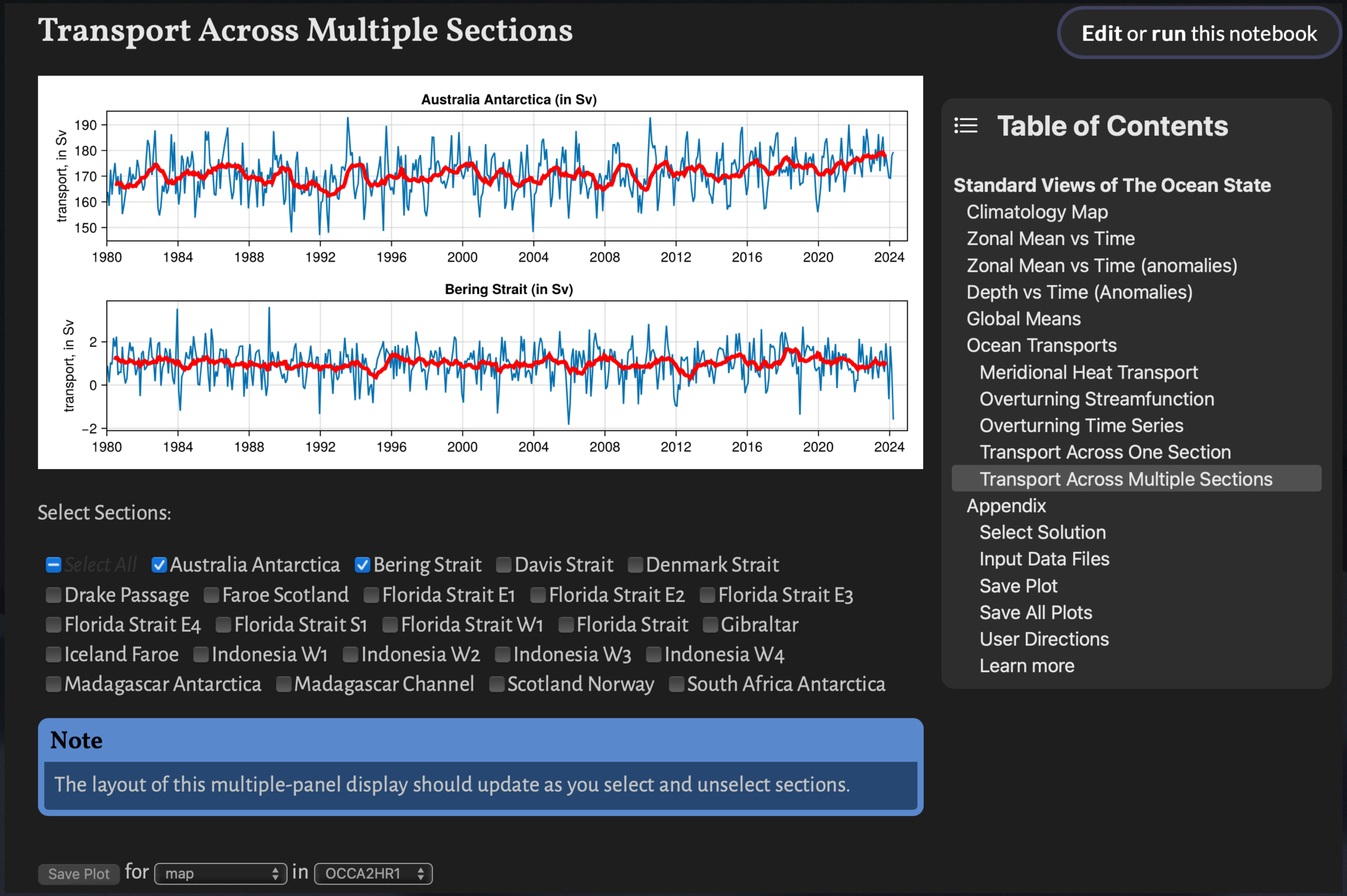
- find nearest neighbor (`MeshArray & set`)
- define subdomain tiles (`MeshArray -> tiles`)
- exchange and start loop (`tile & subset`)
  - local stereographic projection
  - define array of quadrilaterals
  - find enclosing quadrilaterals
  - compute interpolation coefficients





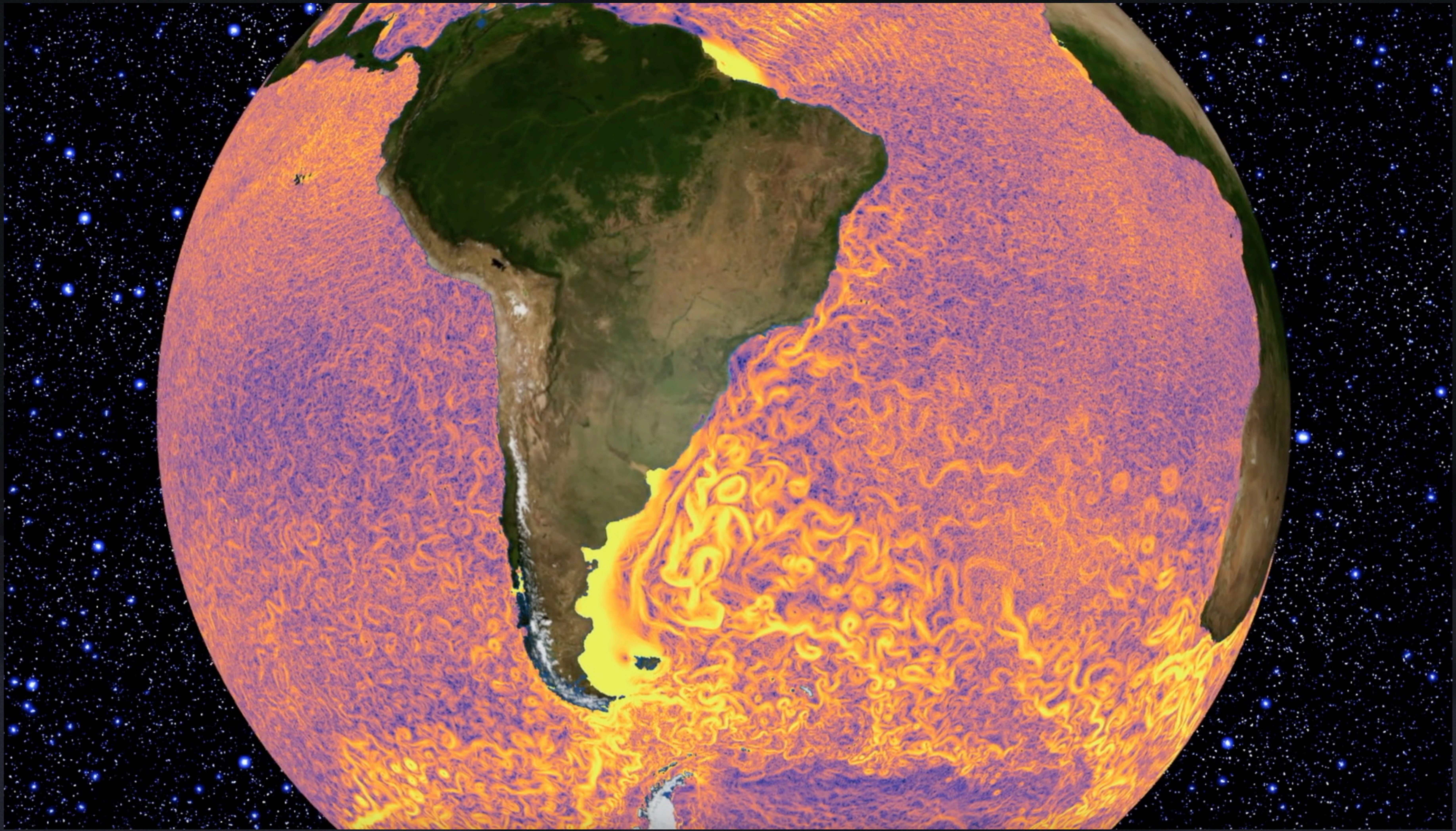
Climatology.jl

User friendly access to interpolated, integrated, etc estimates

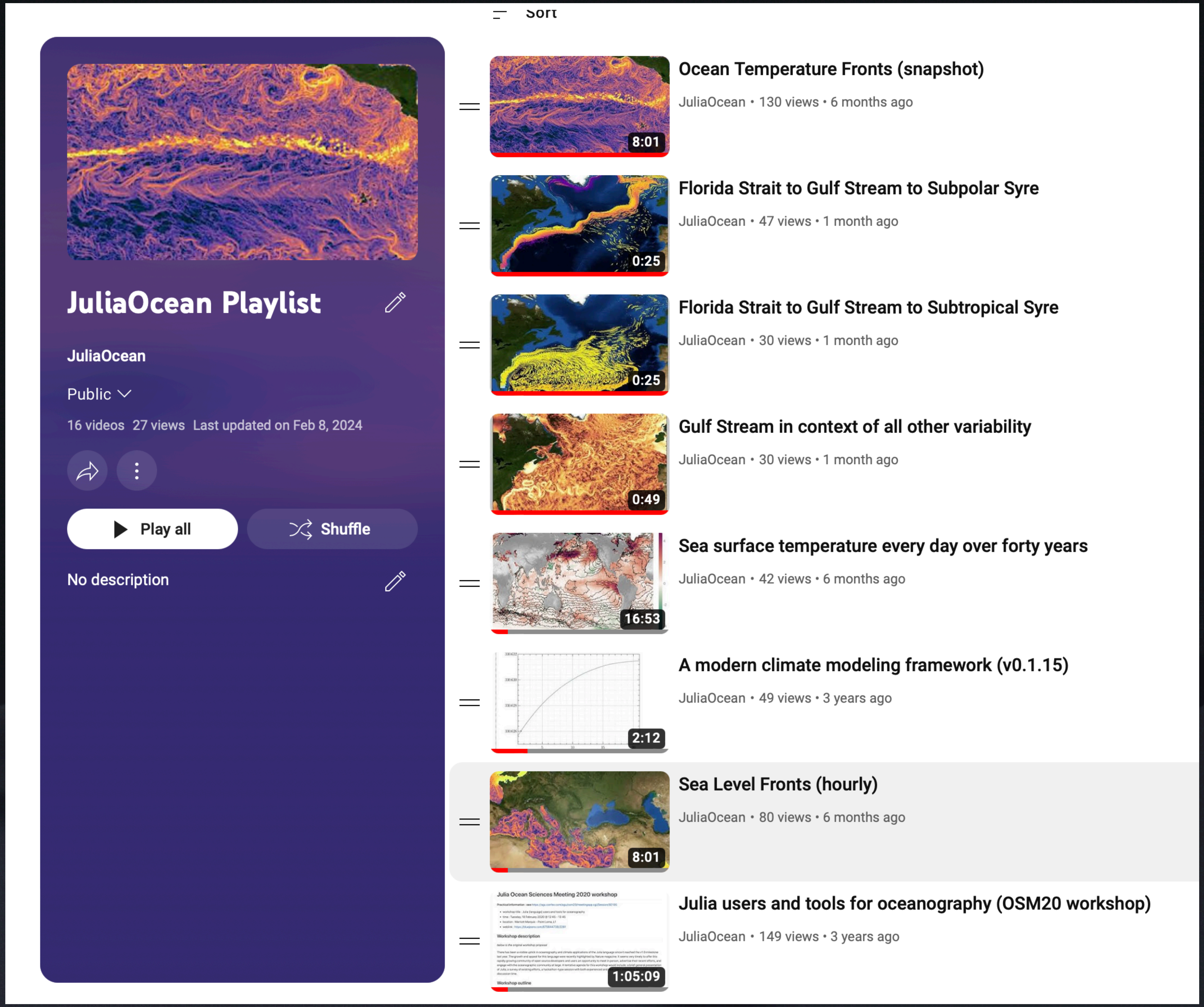


Pre-Julia high-resolution model prototypes

MITgcm on the Lat-Lon-Cap grid at 4km resolution



More Animations?



JuliaOcean playlist

Outline

- General Circulation Modeling
- Data Driven Ocean Modeling
- Geospatial & Analysis Perspective
- Bridging Communities
- Today's session

ECCO-Hackweek 2024

(@Caltech, by Fenty, Fukumori, Wang, Delman, Forget)

Search ⌘ + K

Welcome to ECCO HackWeek!

Details

- Event Logistics
- Schedule
- Team
- Participant Conduct

Welcome to ECCO HackWeek!

📖 On this JupyterBook website you'll find tutorials. All tutorials are Jupyter Notebooks, designed to be run interactively, but also rendered on this website for convenience.

Quick links for the event

- JupyterHub: https://sealevel.oss.eis.smce.nasa.gov
- GitHub organization: ECCO-Hackweek/ecco-2024
- Slack Workspace: https://2024eccohackweek.slack.com
- Presentations: https://drive.google.com/drive/folders/1EmQ4el3x79ZULBKuzRfOK1RI-9EanCA6?usp=drive_link

Next
Event Logistics

Julia Tutorials

Getting Started with the P-Cluster

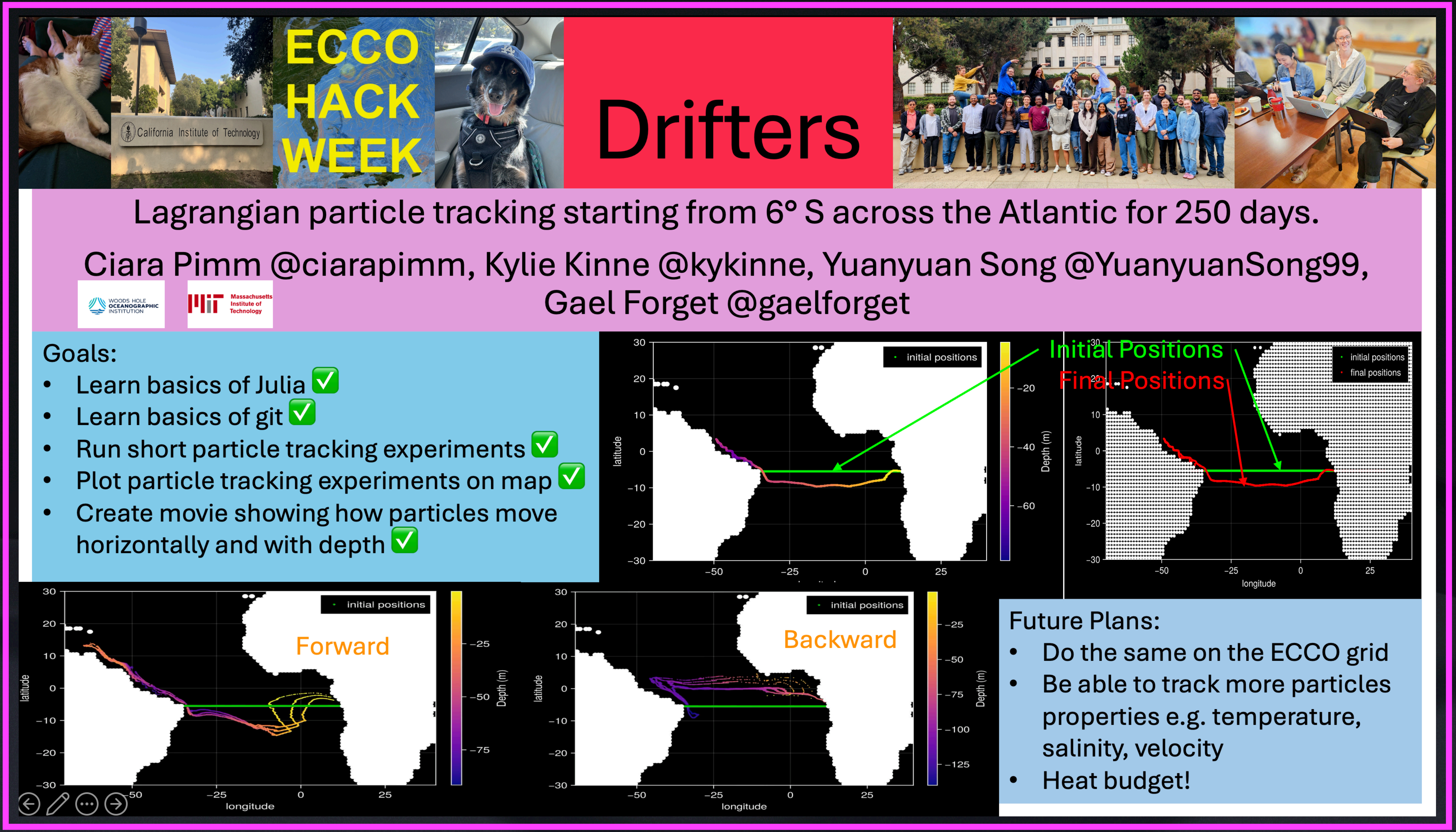
Guidelines to Set Up Julia

Tutorials

- ECCO general information
- ECCO data access
- ECCO v4 computations
- **Julia Tutorials**
  - Geography and Visualization
  - ECCO analyses in Julia
  - Ocean Transports in ECCO
  - The Global Ocean Observing System

Guidelines	Install, Pkg, Julia basics
Visualise the ECCO model grid	Grids, Tiles, Maps, Sections, Projections, Polygons
Explore standard ECCO diagnostics	Zonal Means, Global Means, Time Series, Maps
Visualize and Compute ECCO transports	Transports, streamfunction, Meridional Transports, Transect Throughflows
Track the global observing system	Ocean-OPS, Argo, drifters, OceanRobots.jl
Explore Argo data and compare to ECCO	obtaining data, formatting, sampling ECCO, cost funtions
Explore SST data and compare to ECCO	OISST data, SST in ECCO, Anomaly Maps, Time Series, Marine Heat Waves
Explore SLA data and and compare to ECCO	Altimetry, PODAAC, ECCO, GRACE, cost funtions
Explore NSLCT data and compare to ECCO	Sea Level Change Team, observations, projections
Use the MITgcm.jl interface	setup, build, run, parameters, inputs, outputs
Run ECCO from Julia	configuration, HPC, diagnostics, perturbations, adjoint runs
Tracking particles in the Ocean	Pathways, Water Masses, Drifters, Floats, Plastics
Automatic Differentiation and Optimization	Adjoint, Automatic Differentiation, Optimization, Line Search
Interface to Models in Various Languages	MITgcm, Other Models, Automated Workflows

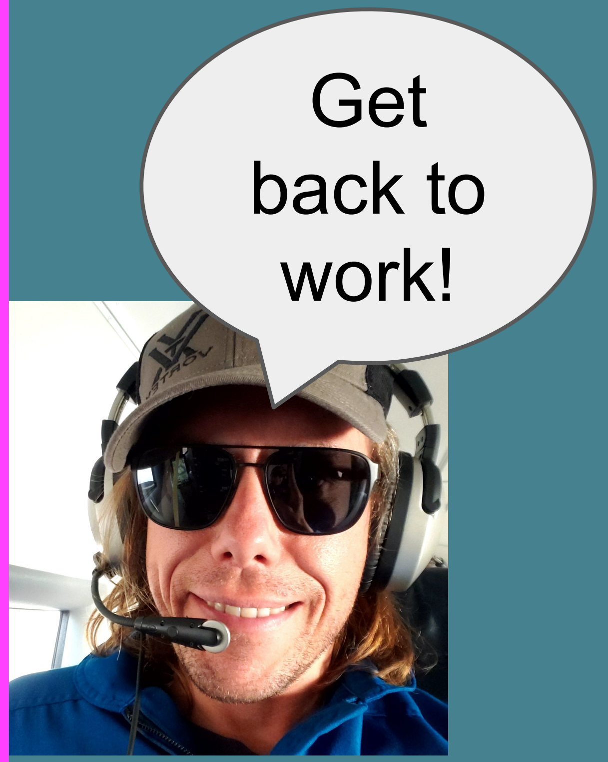
Drifters.jl Project



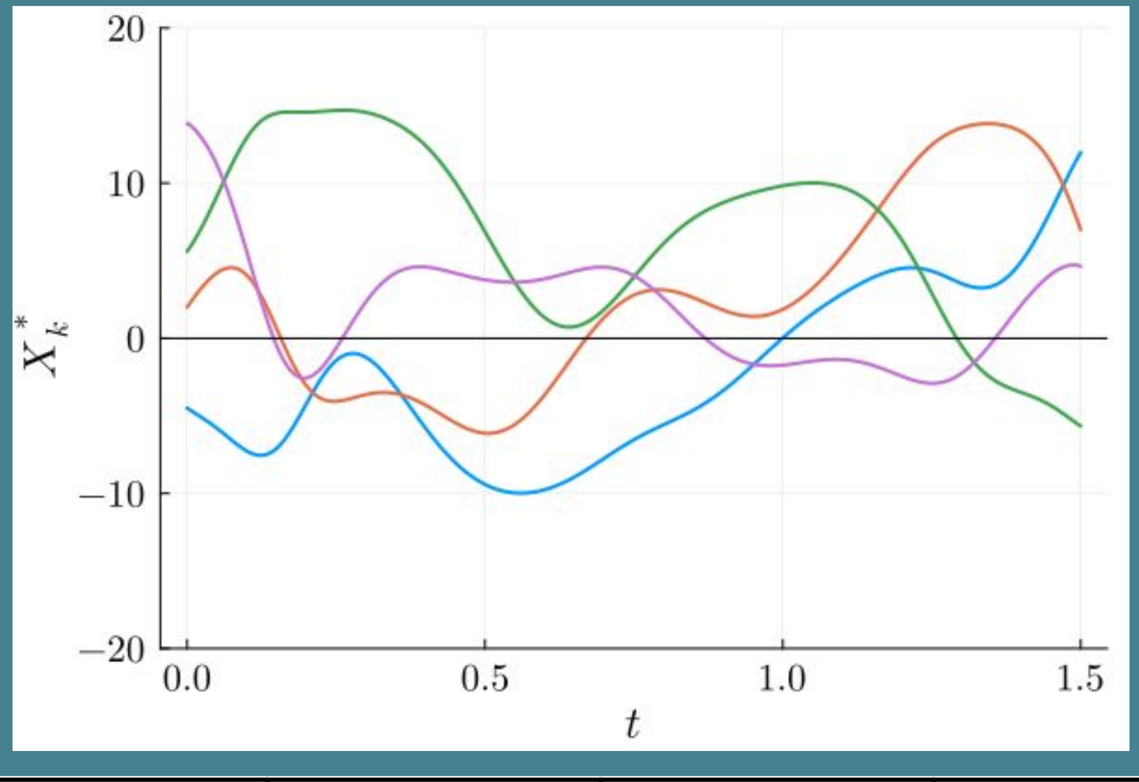
ECCO.jl Project

Toy ECCO: AD Tool Intercomparisons

Shreyas Gaikwad, Max Trostel, Ellen Davenport, Gaël Forget

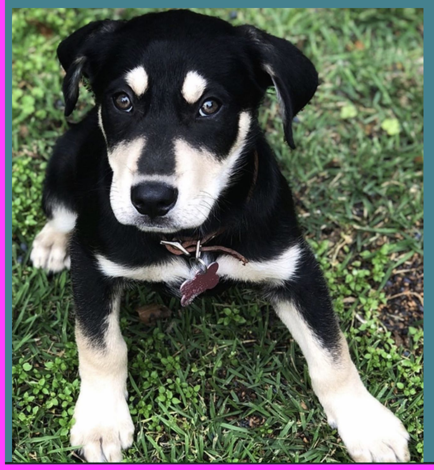


	EXF bulk formulae	Mountain Glacier	L96
JAX		X	X
PyTorch		X	X
Enzyme	X		X
Tapenade		X	
Julia	X	X	X
Python		X	X
Fortran	X	X	
Optimization	X		X



- 3 languages (Python, Julia, Fortran)
- 4 AD tools (Enzyme, PyTorch, JAX, Tapenade)
- 3 Algorithms (bulk formulae, L96, mountain glacier)
- 16 toy problem experiments!!!!!

	EXF bulk formulae	Mountain Glacier	L96	With compile -> pre-compiled
JAX		4000s	17s	Time to run forward + adjoint
PyTorch		790s	0.60s	
Enzyme	X		27s -> 0.0025s	
Tapenade		0.52s		
Julia	X	7.7s	0.15s -> 0.000056s	Time to run foward
Python		1.5s	0.22s	
Fortran	X	0.14s		
Optimization	X		X	



Outline

- General Circulation Modeling
- Data Driven Ocean Modeling
- Geospatial & Analysis Perspective
- Bridging Communities
- Today's session

- 10:05 - 10:45 Training denoising diffusion models on incomplete satellite data in Julia with Flux.jl Mr. Alexander Barth
- 10:45 – 11:25 Realistic Ocean Simulations in Pure Julia with ClimaOcean.jl Mr. Simone Silvestri
- 11:25 – 12:00 Composable Ocean Modelling with ClimateModels.jl, MITgcm.jl, and Climatology.jl Mr. Gaël Forget

- 13:00 – 13:50 Hands-on Session on Regional Ocean Modelling via ROMS.jl Mr. Alexander Barth
- 13:50 – 14:40 Hands-on Session with Oceananigans.jl and ClimaOcean.jl Mr. Simone Silvestri
- 14:40 – 15:30 Hands-on Session on Tracking Ocean Currents with OceanRobots.jl, MeshArrays.jl, and Drifters.jl Mr. Gaël Forget

- 16:00 – 16:20 Round Table Mr. Alexander Barth, Mr. Simone Silvestri, & Mr. Gaël Forget