

Digital Twins for Ocean Robots and Marine Ecosystems

Annual ECCO meeting

Gaël Forget

Caltech, LA

2023/01/26

Symposium on Advances in Ocean Observations (2022)

Bring together a small yet highly motivated group of experts focusing on smarter methods in ocean observation. The aim is to generate ideas across science and technology, to advance ocean observation in novel ways.

Organizers

Aida Alvera Azcárate, Physical Oceanography, Ocean Remote Sensing, Belgium

Jo Eidsvik, Statistical Sampling, Norway

Kanna Rajan, Autonomous Systems, AI, United States and Portugal

July 3rd to July 7th, 2022

[URL](#)



JuliaEO23 : Global Workshop on Earth Observation with Julia

- 40 in person, 300 virtual
- 20+ Julia notebooks
- 20h recorded video
- Docker, Dataverse
- Geospatial data, Earthdata, SAR, sentinel 2, rasters, big data, oceanography, modeling, cloud computing, classification, stats, AI, viz, ...

Organizers

Joao Pinelo, Iga Szczesniak, Andre Valente

AIR Centre

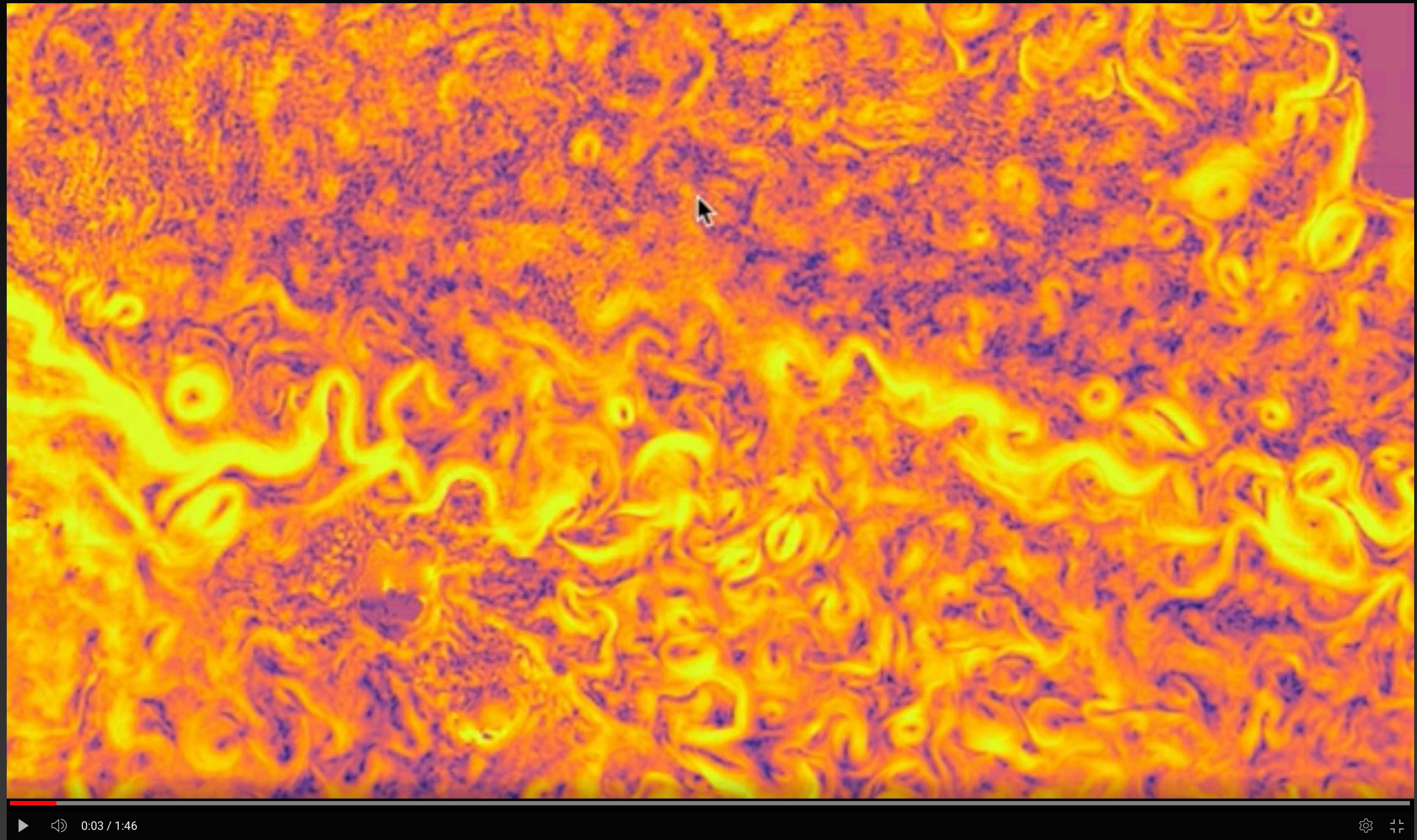
Gaël Forget, MIT

January 9th to 13th, 2023

[URL](#)



Google Maps for us in Julia!



CBIOMES

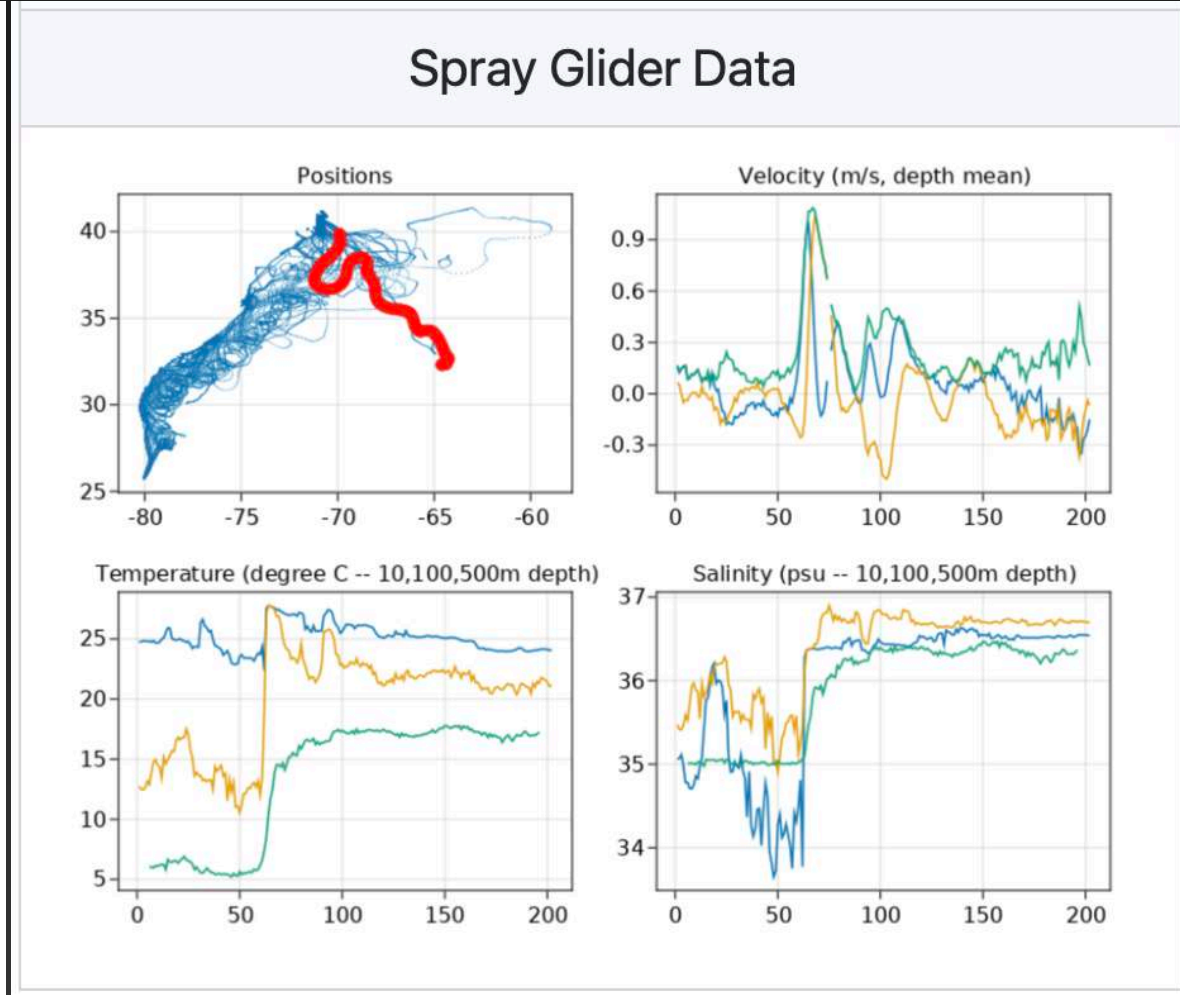
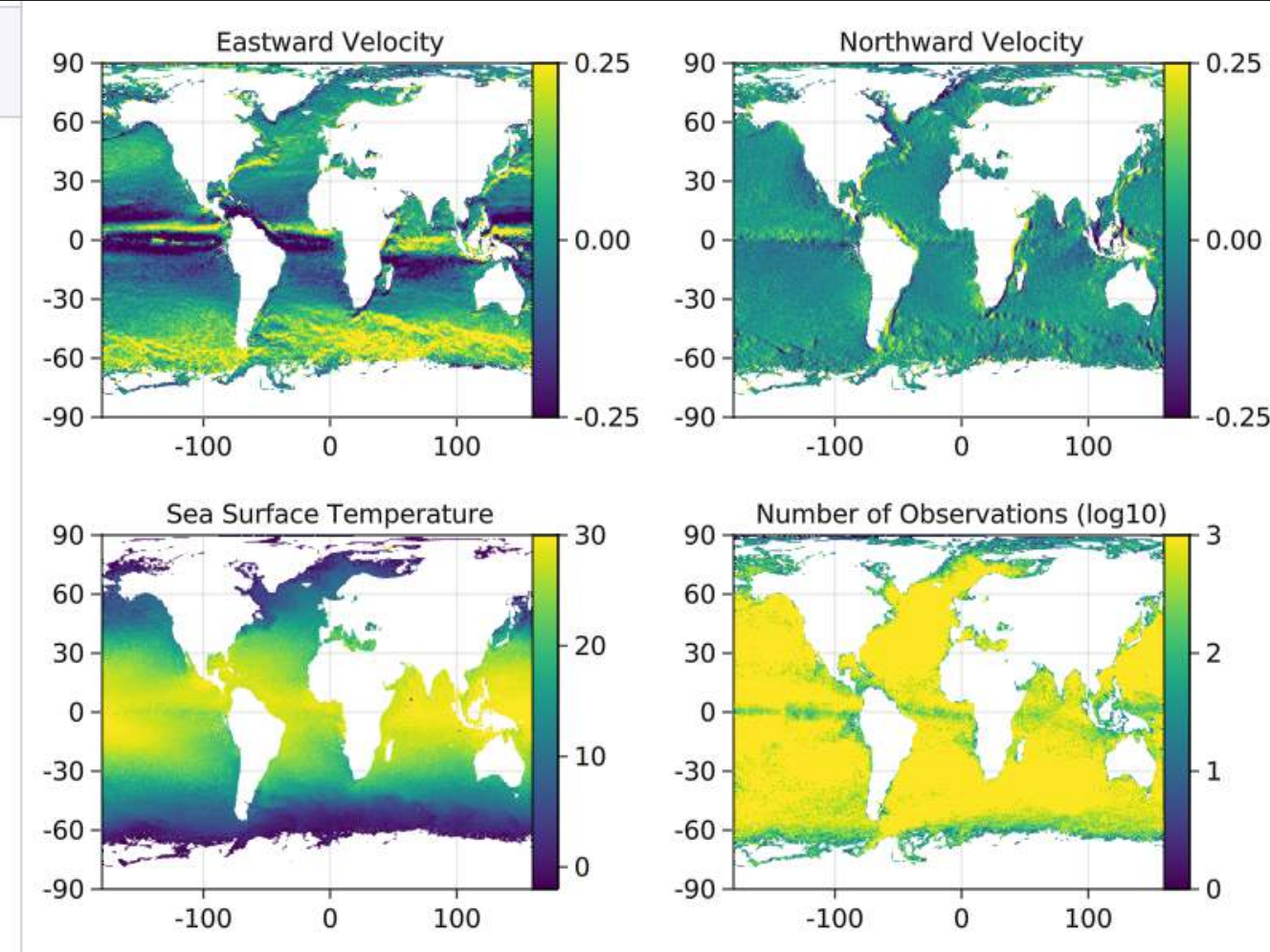
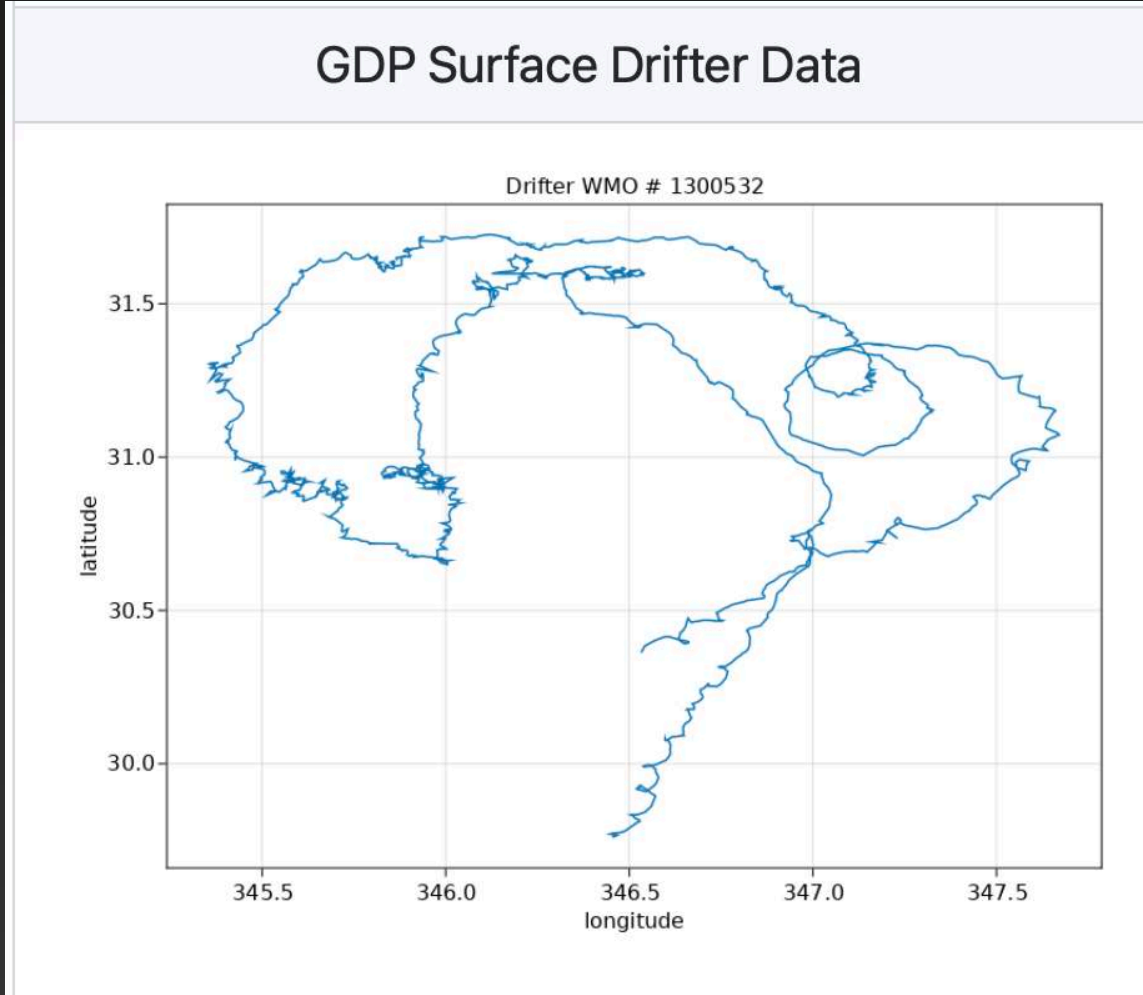
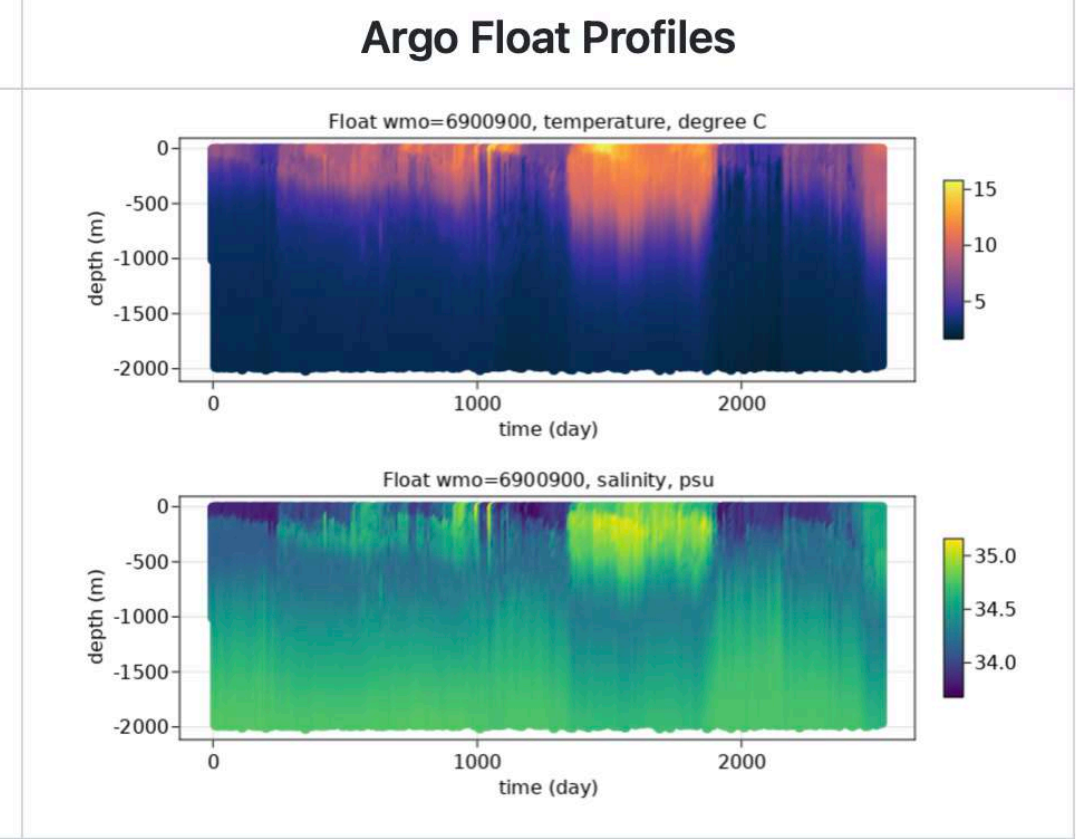
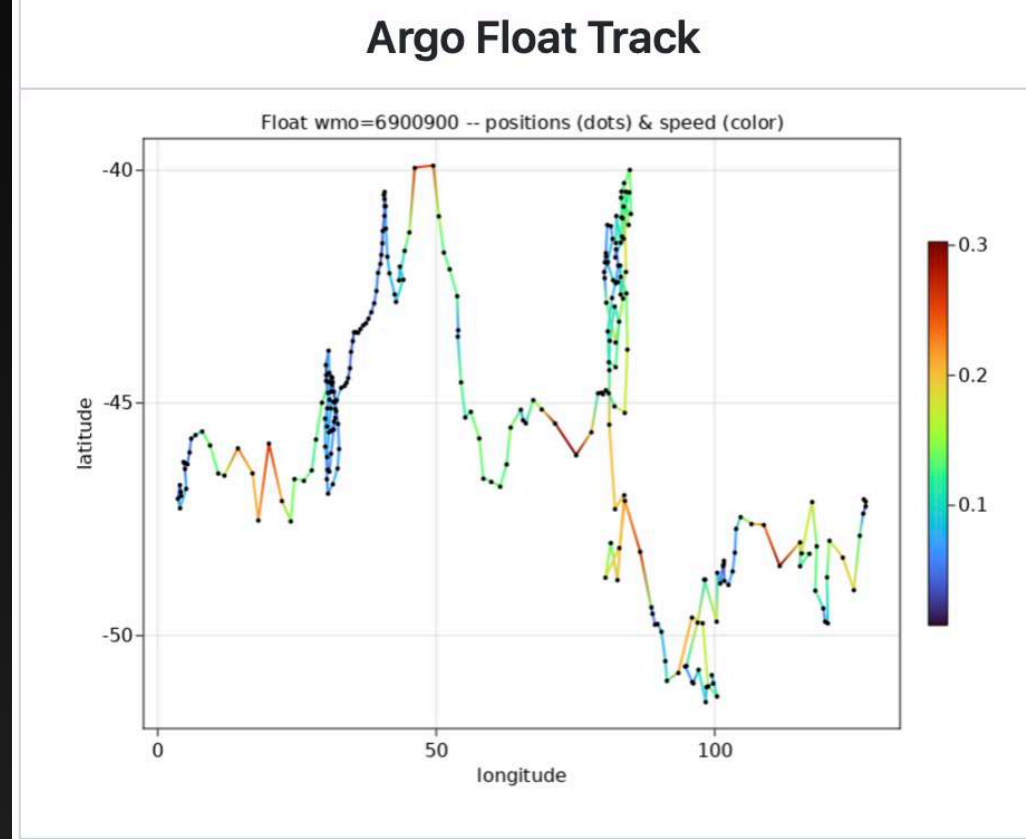
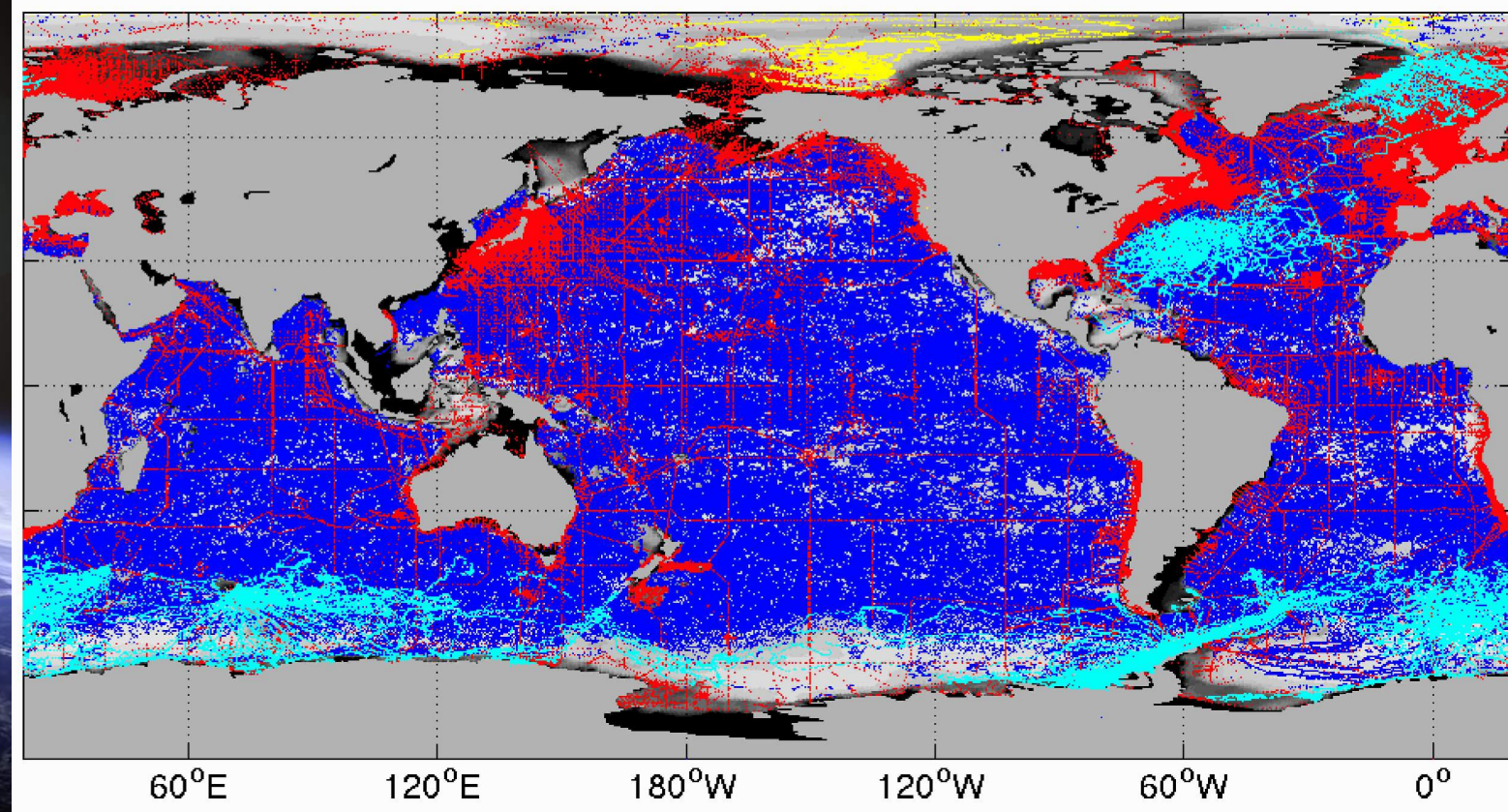
cbiomes

Simons Collaboration on Computational
Biogeochemical Modeling of Marine Ecosystems

- Johnson et al 2023 (subm)
- Hyun et al 2022
- Britten et al 2022
- Wu & Forget 2022
- Follett et al 2021
- Forget 2021
- Sonnewald et al 2020



Ocean Robots



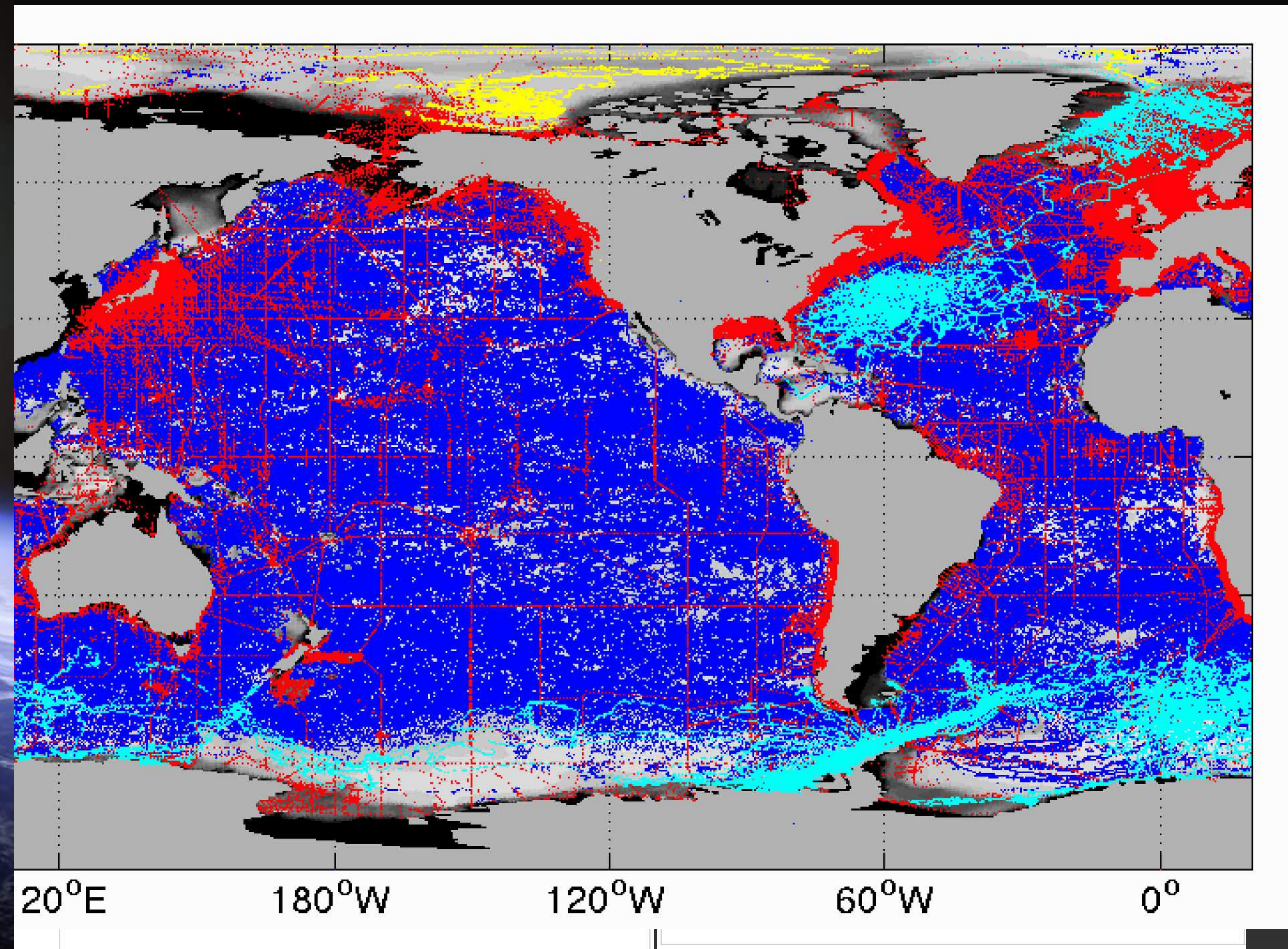
[ECCO version 4](#)
Forget et al 2015 (GMD)

[OceanRobots.jl](#)
Forget 2021 (JuliaCon)

Ocean Robots

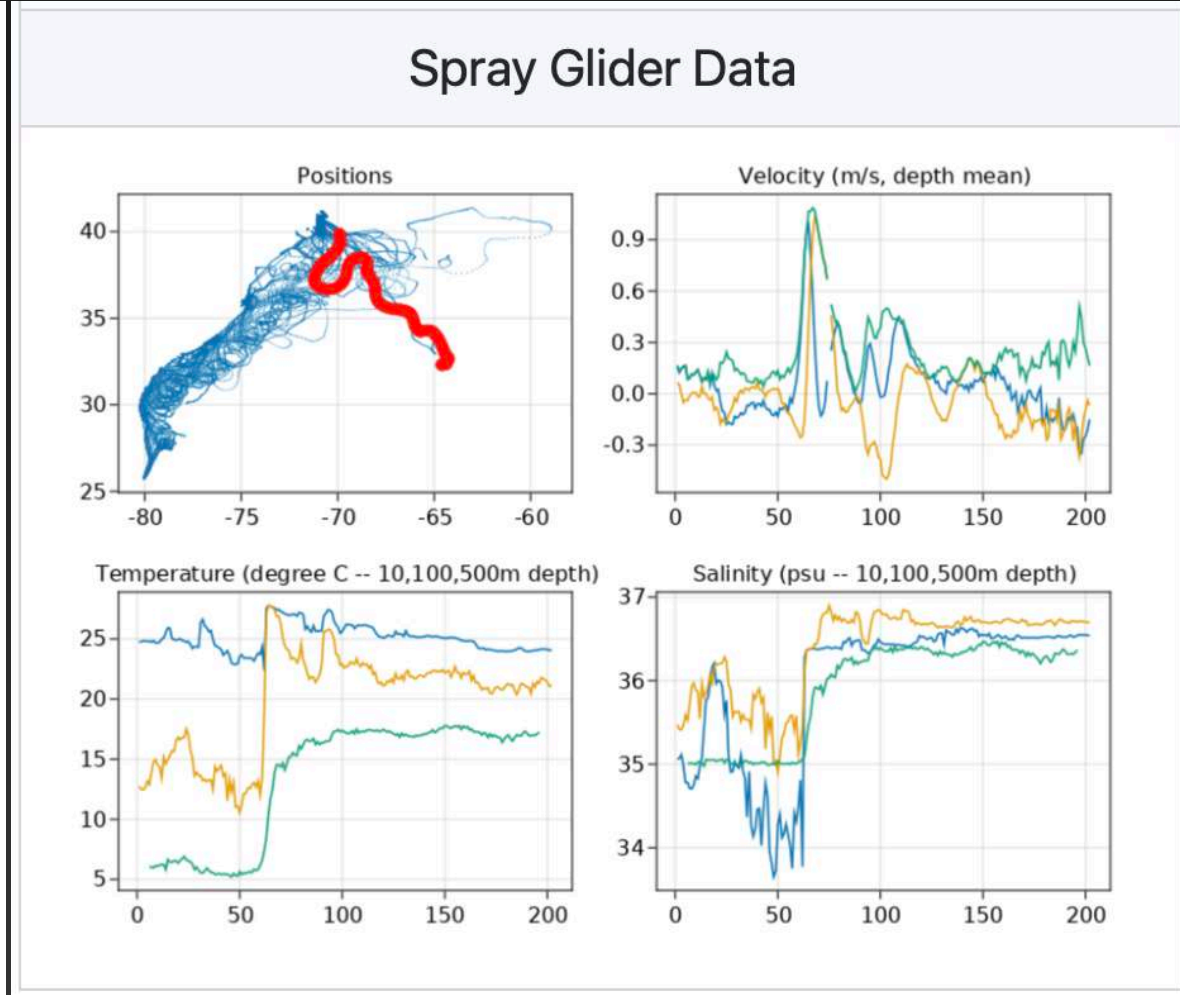
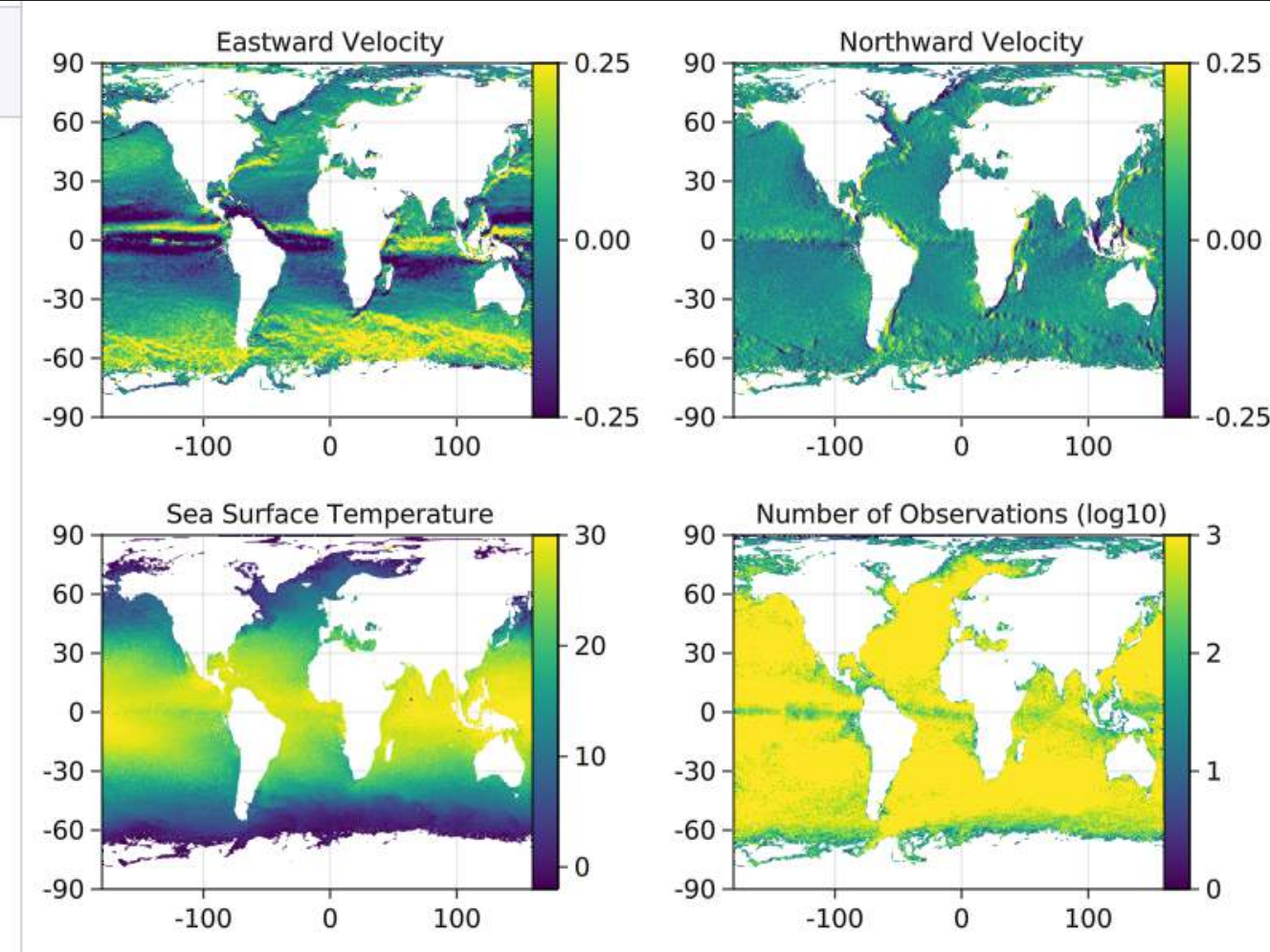
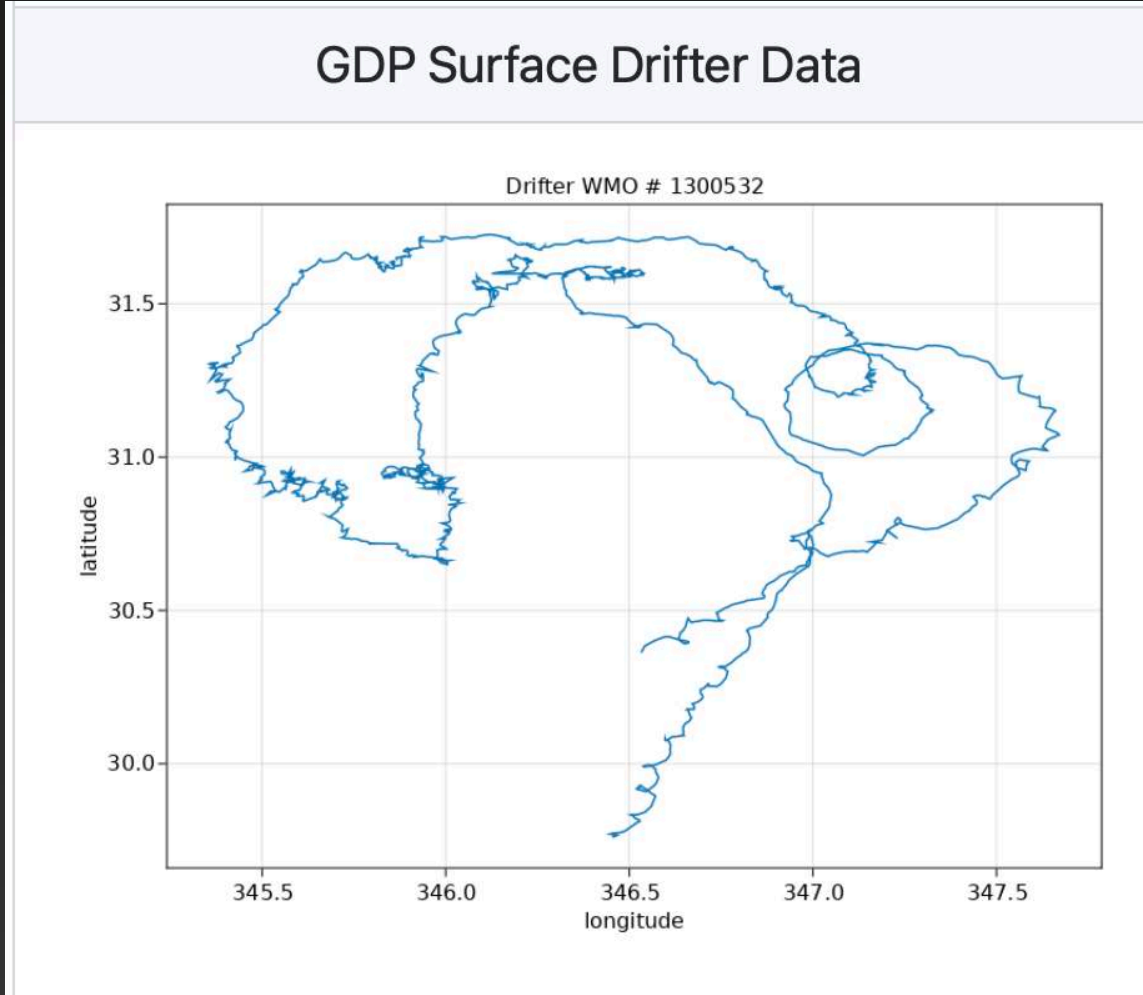
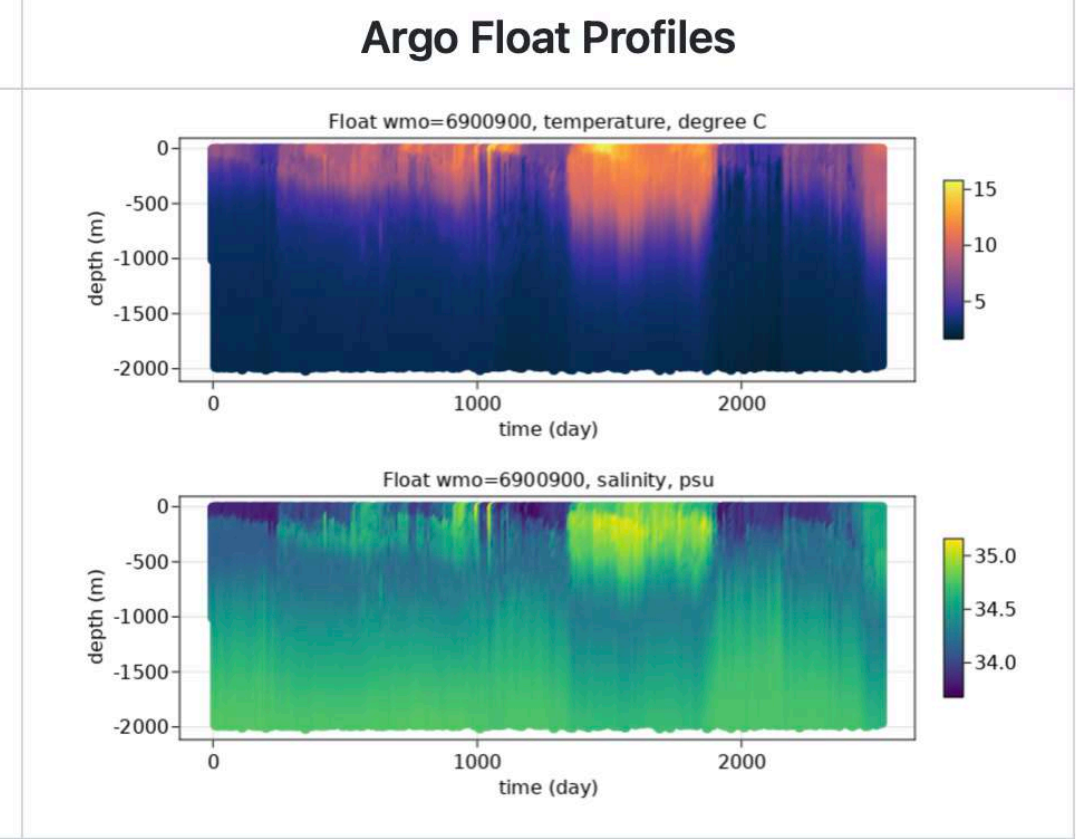
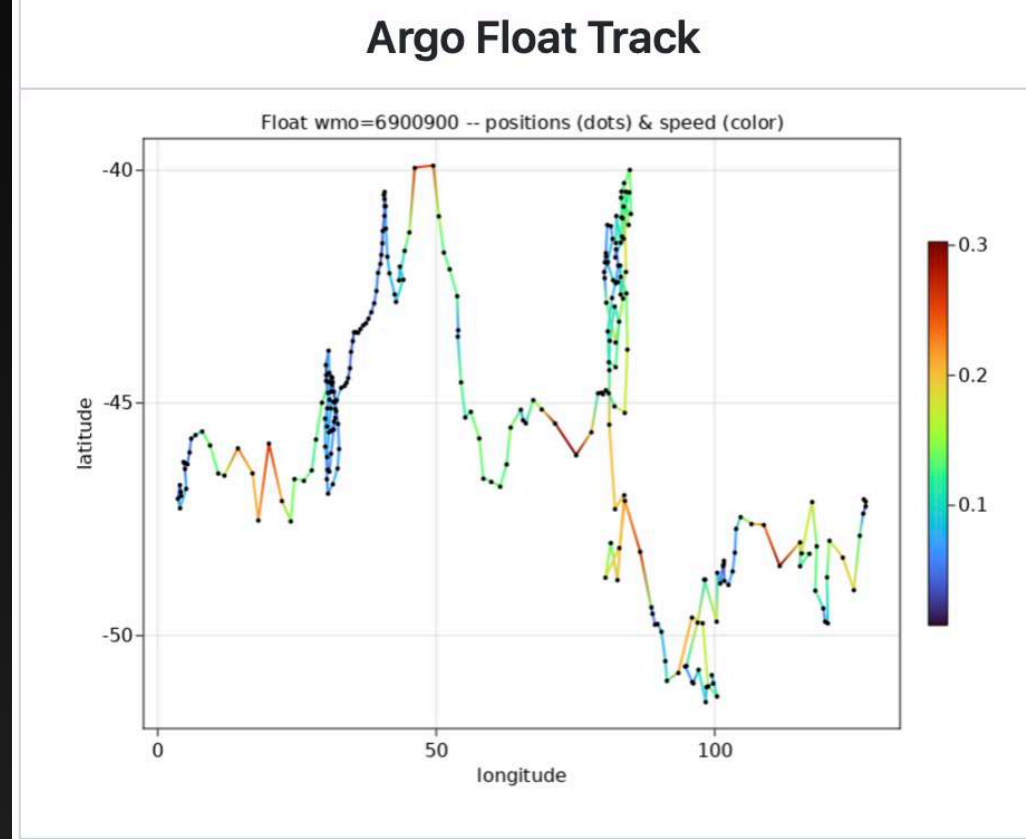
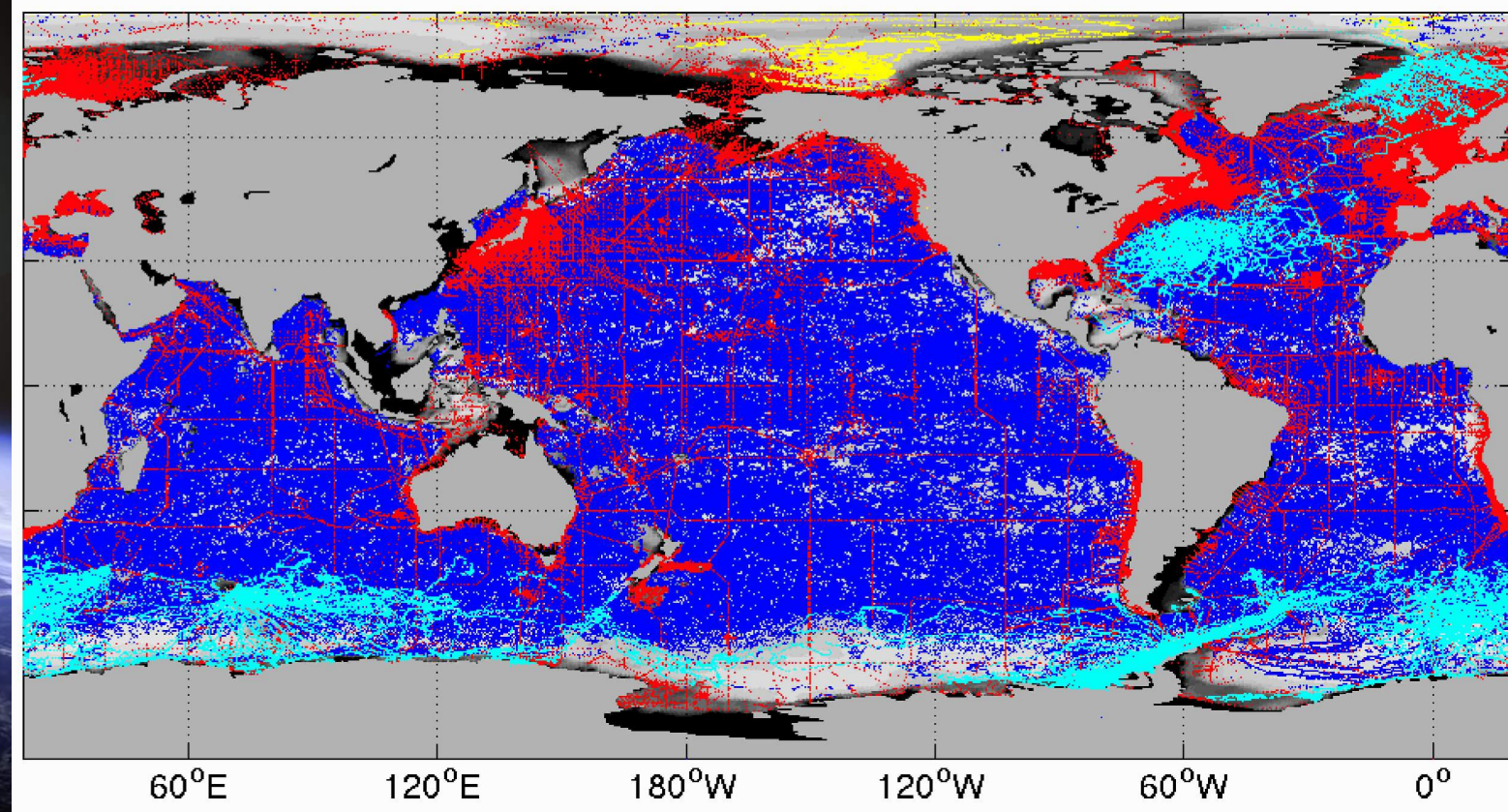


[ECCO version 4](#)
Forget et al 2015 (GMD)



[OceanRobots.jl](#)
Forget 2021 (JuliaCon)

Ocean Robots



[ECCO version 4](#)
Forget et al 2015 (GMD)

[OceanRobots.jl](#)
Forget 2021 (JuliaCon)

Ocean Data Sets

Collected In Situ, From Satellites, or Syntheses

▼ Data (Julia)

- [Dataverse.jl](#) 📖 : interfaces to [Dataverse](#) APIs, collections, datasets, etc stars 6
- [OceanRobots.jl](#) 📖 : simulation and analysis of data generated by ocean robots stars 12
- [ArgoData.jl](#) 📖 : Argo data processing and analysis stars 12
- [OceanColorData.jl](#) 📖 : Ocean color data processing and analysis stars 7
- [OceanStateEstimation.jl](#) 📖 : downloading, reading, displaying, and analyzing ocean state estimates stars 5

- Packages (e.g., using [OceanRobots.jl](#))
- Notebooks (see docs & example folders)



ClimateModels.jl – A Simple Interface To Climate Models

Gaël Forget, JuliaCon2021


ClimateModels.jl provides a uniform interface to climate models of varying complexity and completeness. Models that range from low dimensional to whole Earth System models are ran and analyzed via this simple interface.

Key Features

- Climate Model Interface
- Tracked Workflow Framework
- Cloud + On-Premise File Support

Standard Workflow

- create ModelConfig data structure

 [Live docs](#)

What Are Digital Twins?

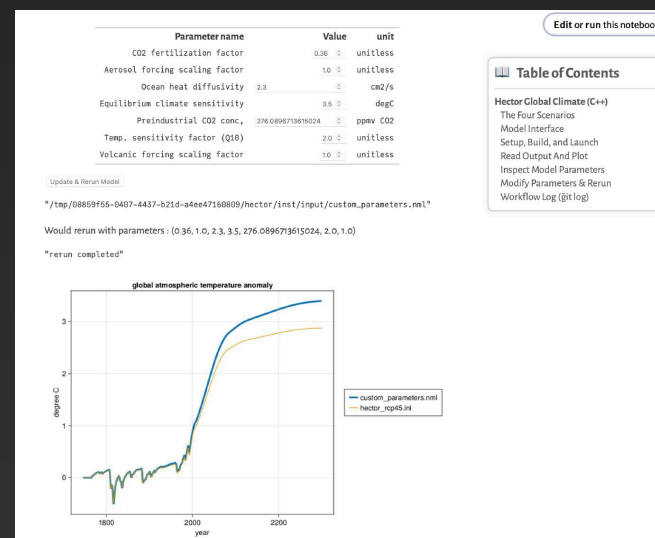
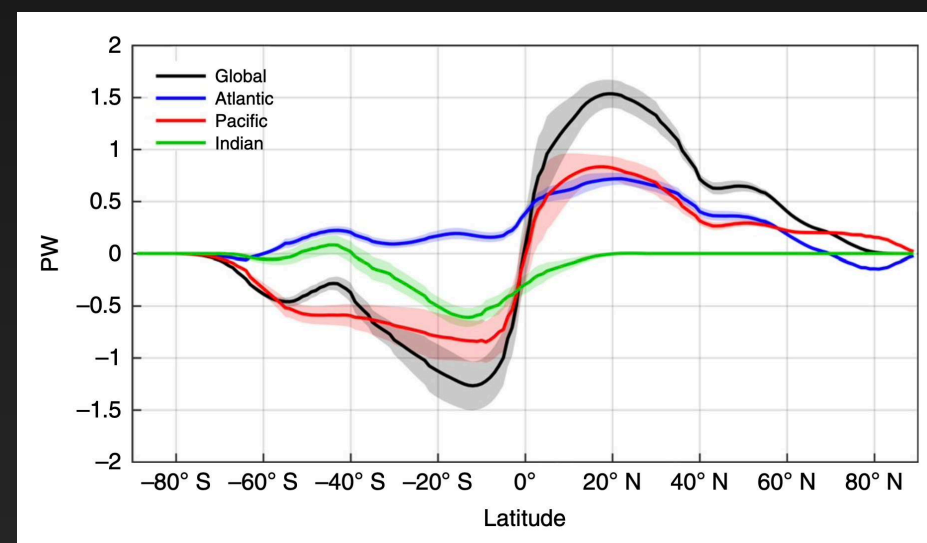
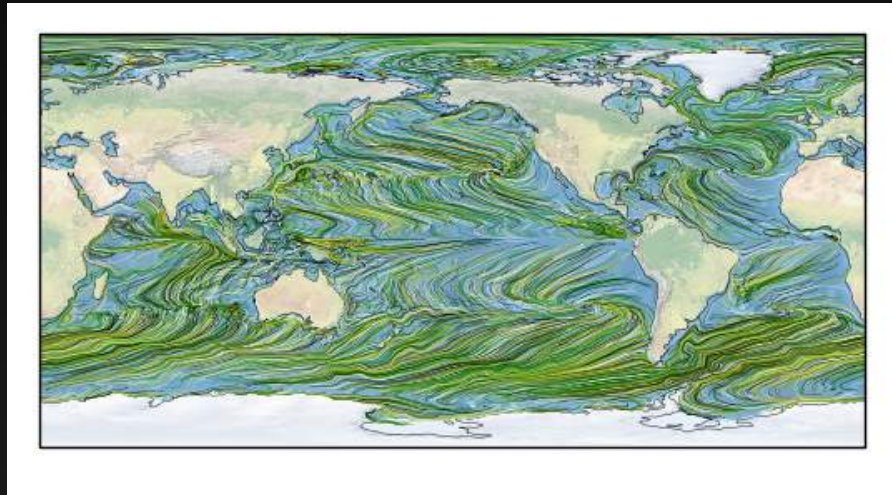
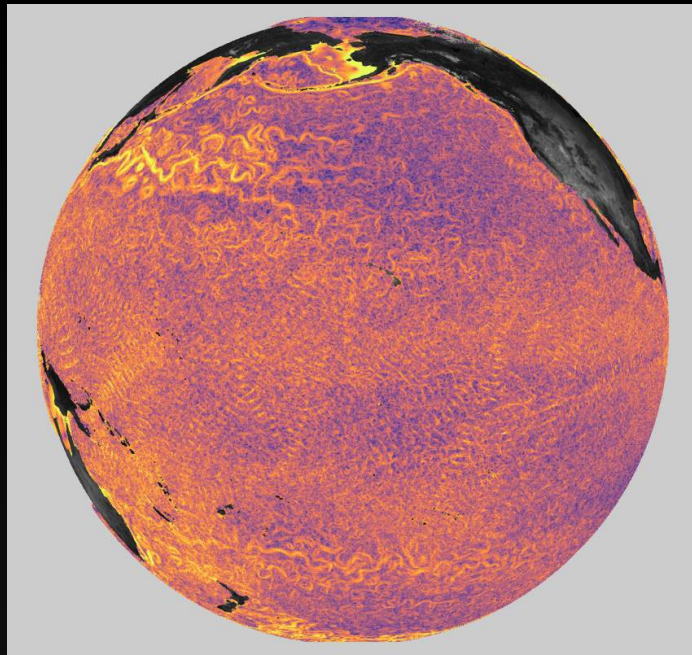
What Are Digital Twins?

- Digital Twins
 - computer representation of real systems that users can interact with
 - + two-way connection framework with real system, enabling mutual benefits

What Are Digital Twins?

- Digital Twins
 - computer representation of real systems that users can interact with
 - + two-way connection framework with real system, enabling mutual benefits
- Our Real System
 - natural system (marine ecosystem < ocean < climate < forcing)
 - + Ocean observing system (autonomous / manned platforms)

Model Hierarchy



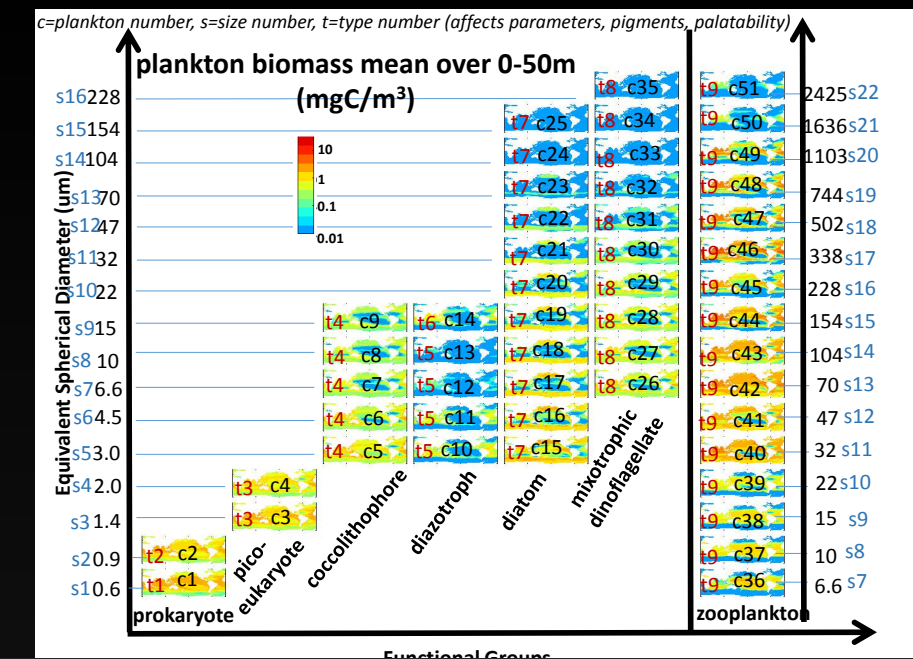
ClimateModels.jl

This package provides a uniform interface to climate models of varying complexity and completeness. Models that range from low dimensional to whole Earth System models can be run and/or analyzed via this framework.

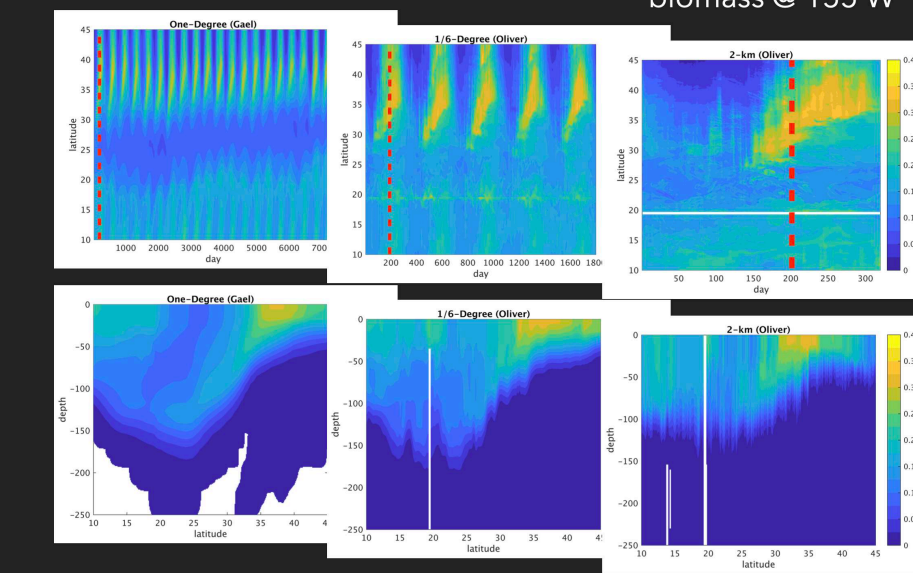
It also supports e.g. cloud computing workflows that start from previous model output available over the internet. Version control, using *git*, is included to allow for workflow documentation and reproducibility.

The [JuliaCon 2021 Presentation](#) provides a brief (8') overview and demo of the package.

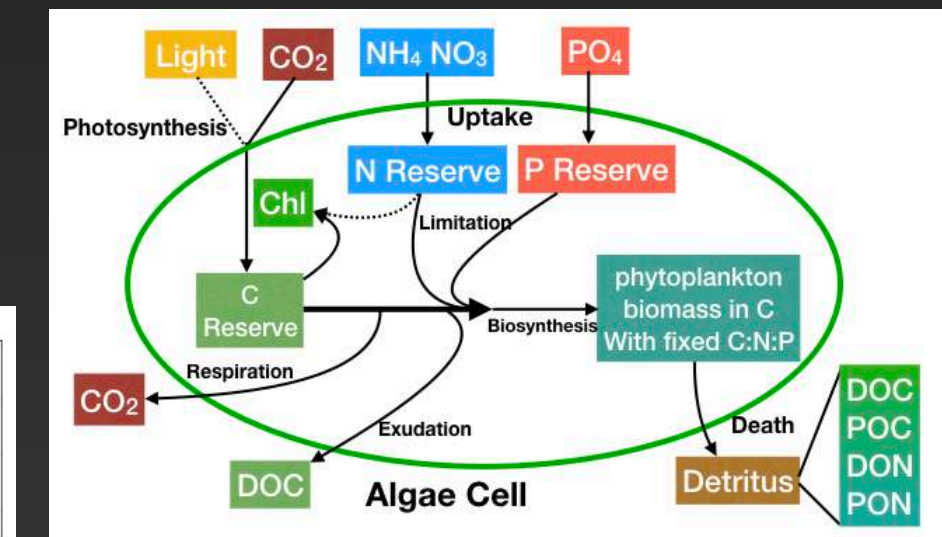
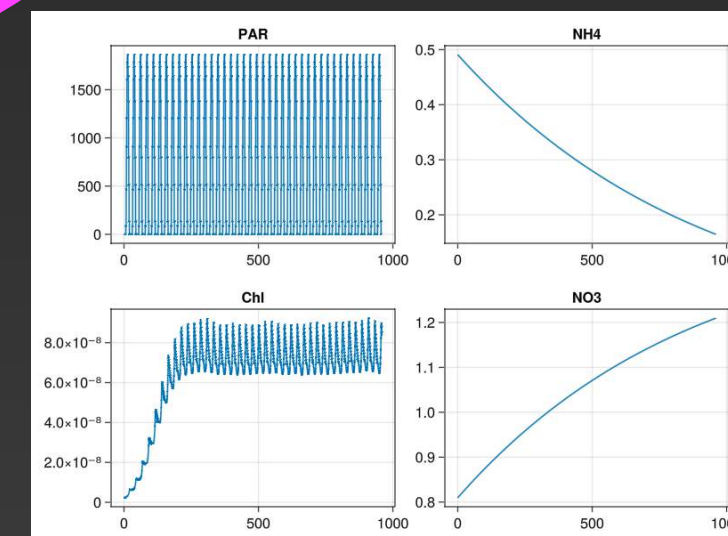
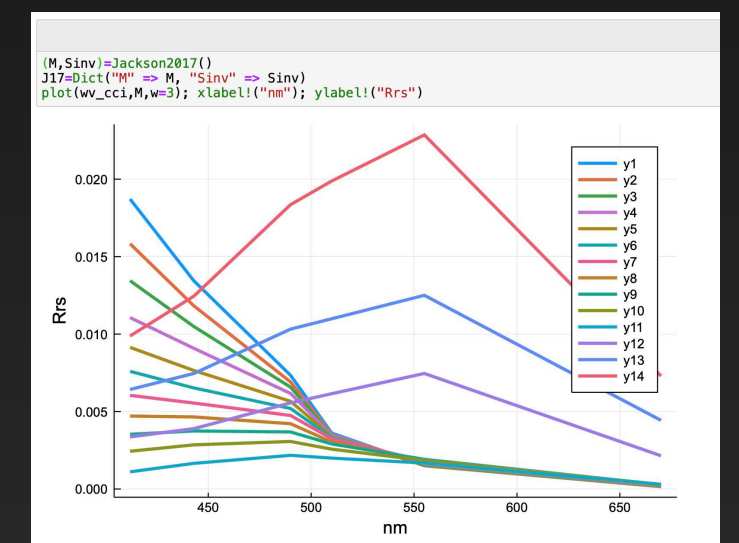
Please refer to [Examples](#) and [User Manual](#) for more detail.

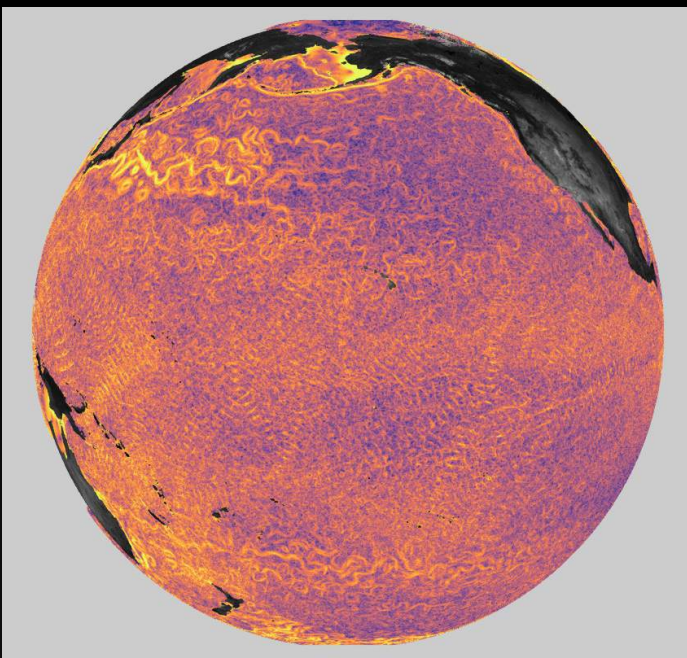


GLOBAL OCEAN MODEL (3/3)

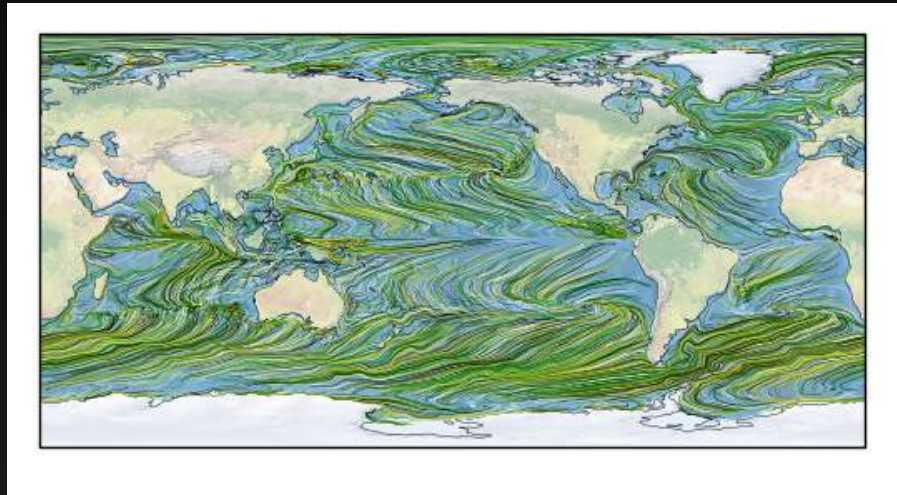
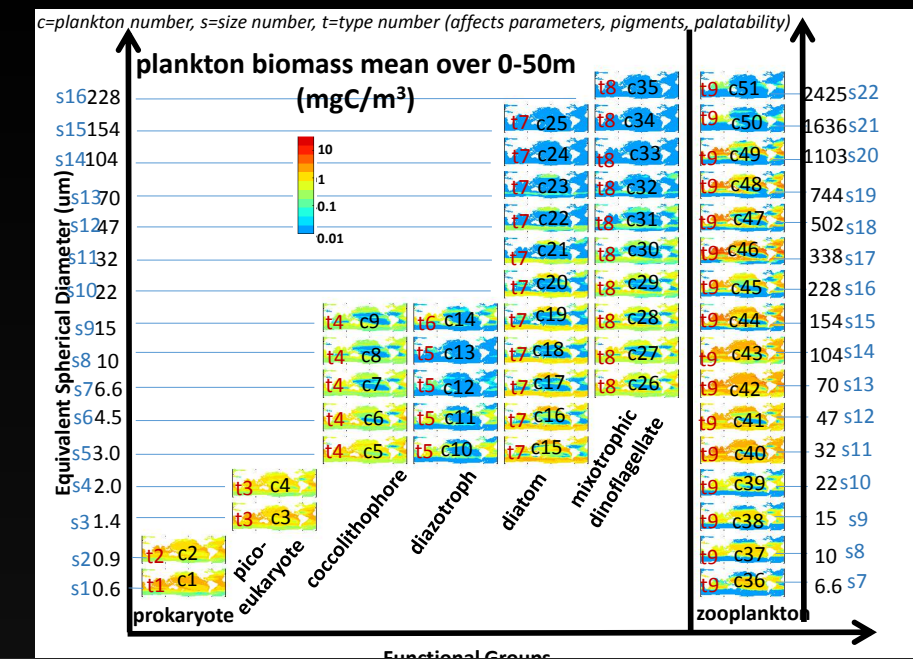


Pico-prokaryote biomass @ 155 W





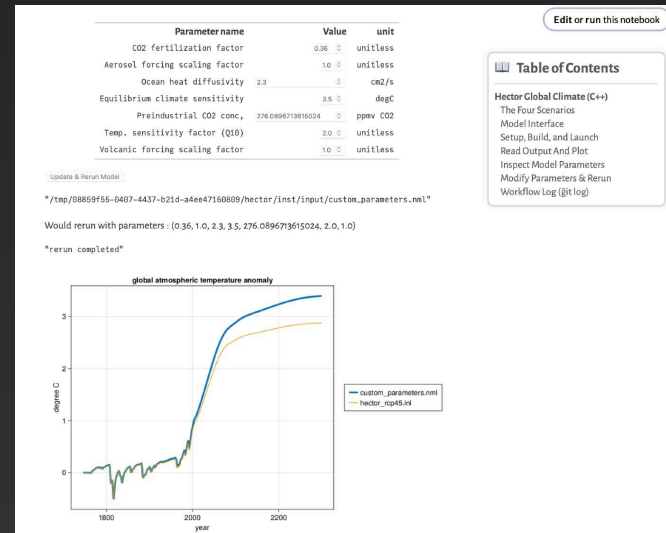
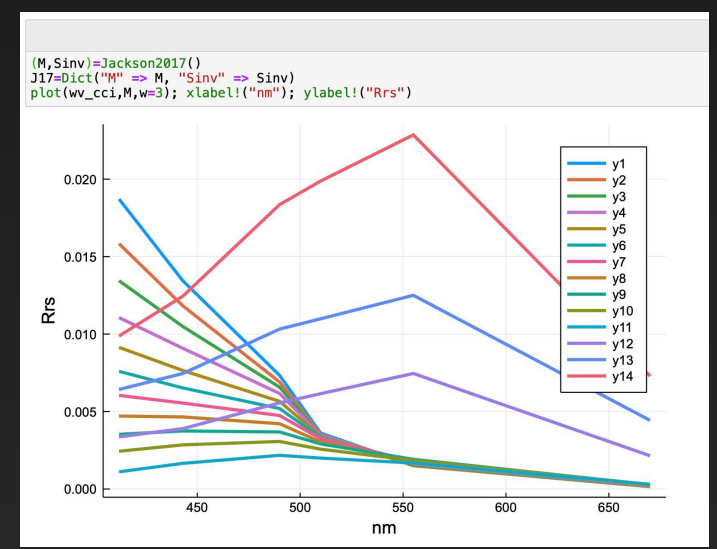
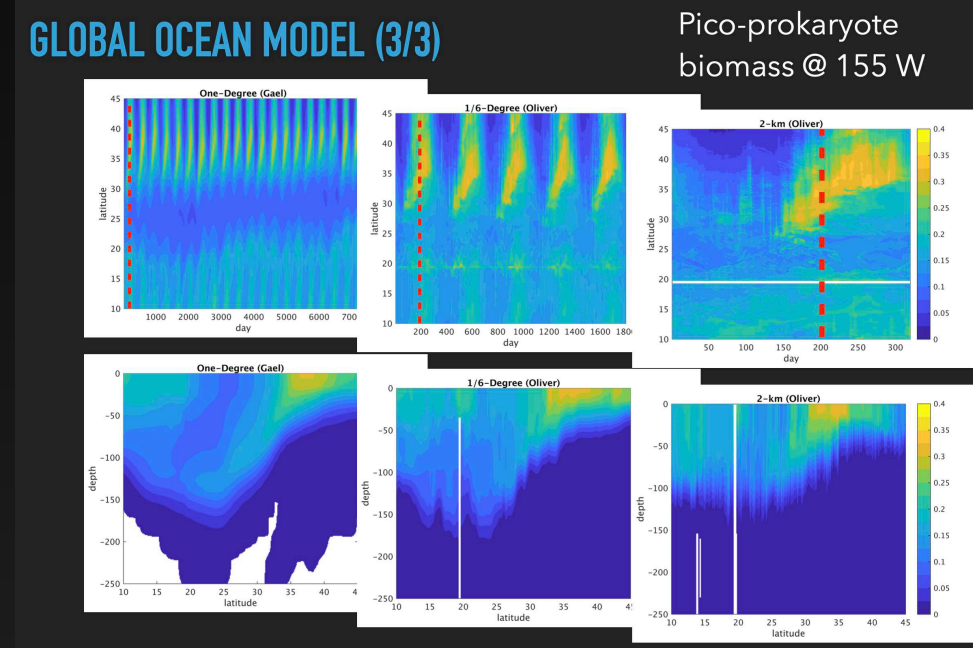
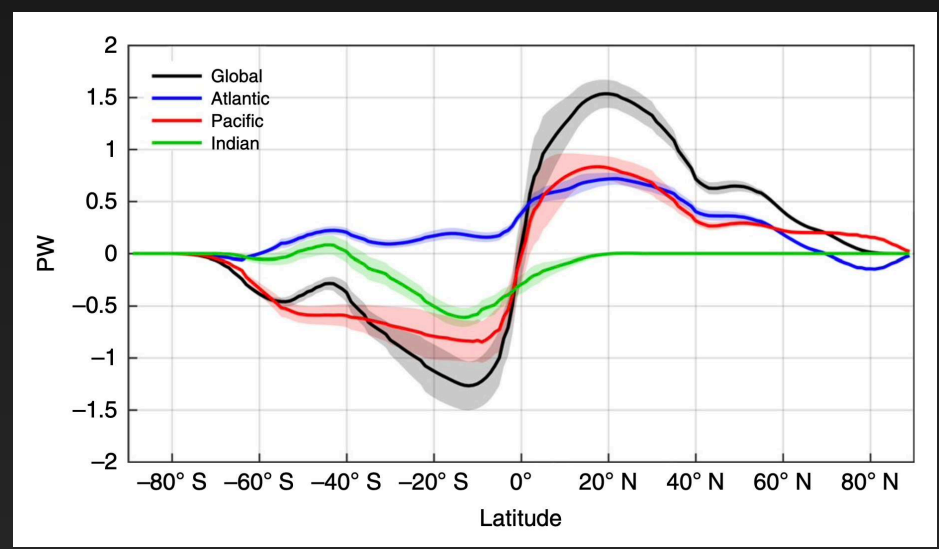
Model Hierarchy



ClimateModels.jl

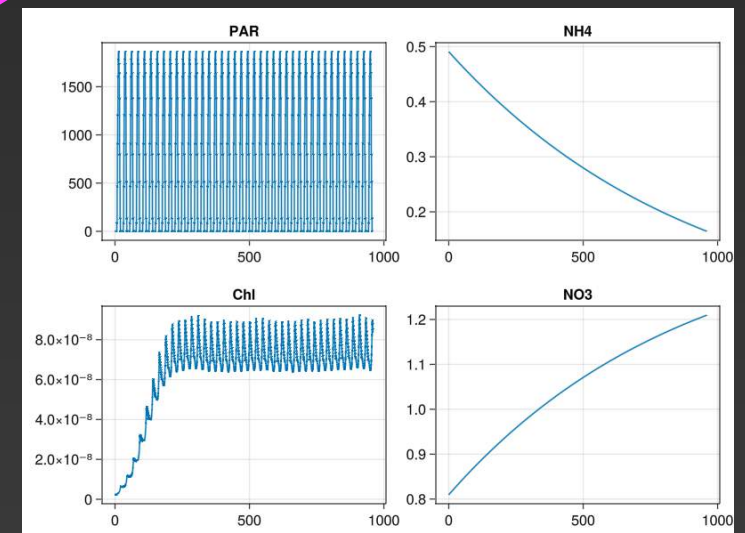
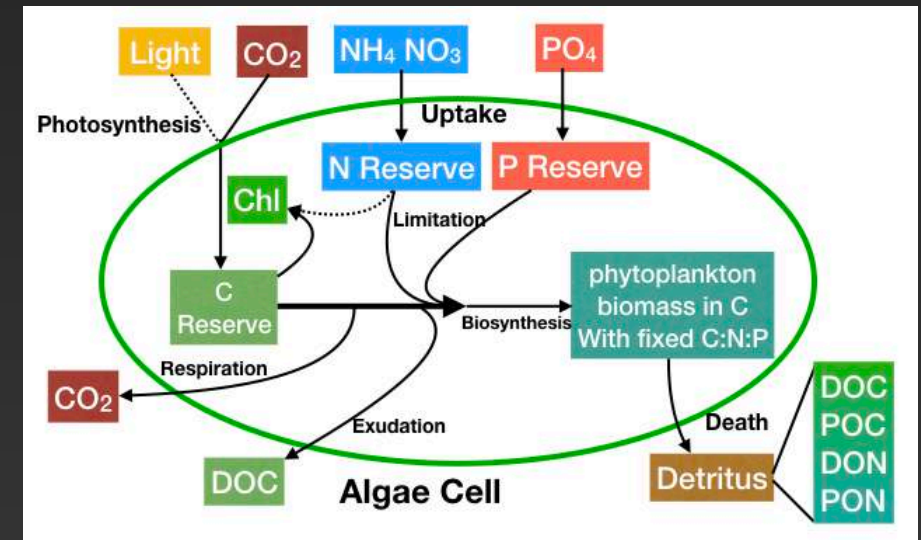
This package provides a uniform interface to climate models of varying complexity and completeness. Models that range from low dimensional to whole Earth System models can be run and/or analyzed via this framework.

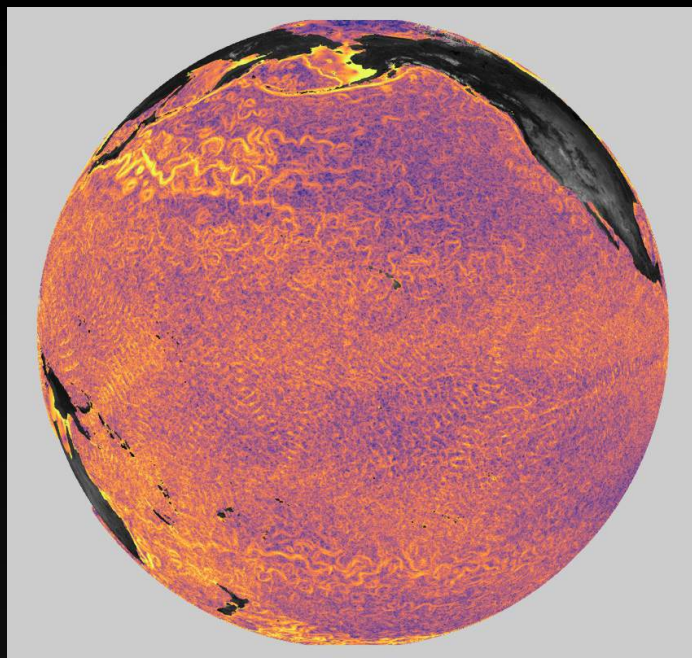
It also supports e.g. cloud computing workflows that start from previous model output available over the internet. Version control, using *git*, is included to allow for workflow documentation and reproducibility.



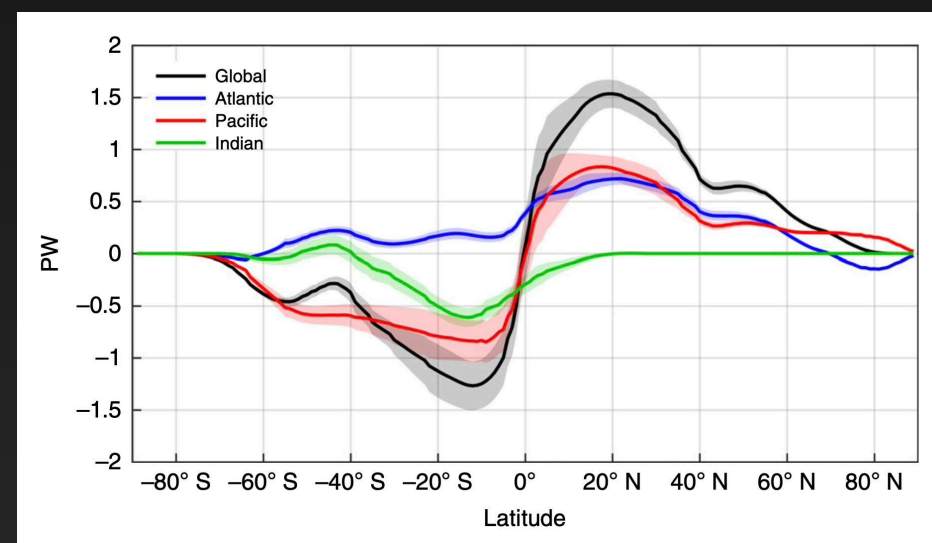
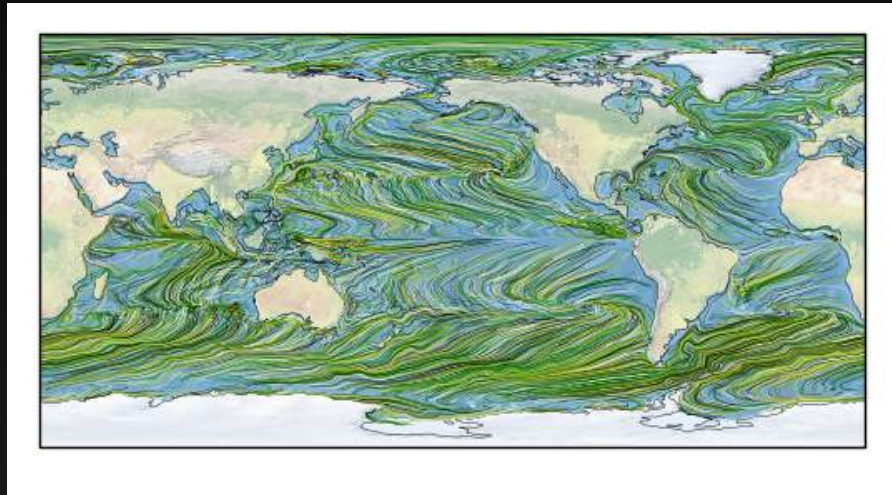
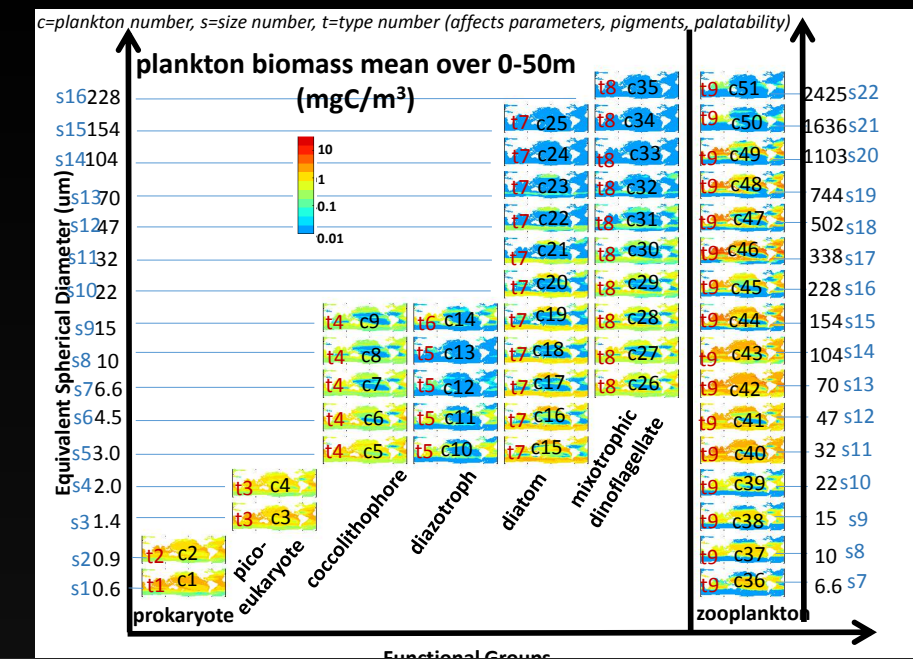
The [JuliaCon 2021 Presentation](#) provides a brief (8') overview and demo of the package.

Please refer to [Examples](#) and [User Manual](#) for more detail.





Model Hierarchy



[Edit or run this notebook](#)

| Parameter name | Value | unit |
|---------------------------------|-------------------|----------|
| CO2 fertilization factor | 0.36 | unitless |
| Aerosol forcing scaling factor | 1.0 | unitless |
| Ocean heat diffusivity | 2.3 | cm2/s |
| Equilibrium climate sensitivity | 3.5 | degC |
| Preindustrial CO2 conc, | 276.0896713615024 | ppmv CO2 |
| Temp. sensitivity factor (Q10) | 2.0 | unitless |
| Volcanic forcing scaling factor | 1.0 | unitless |

Table of Contents

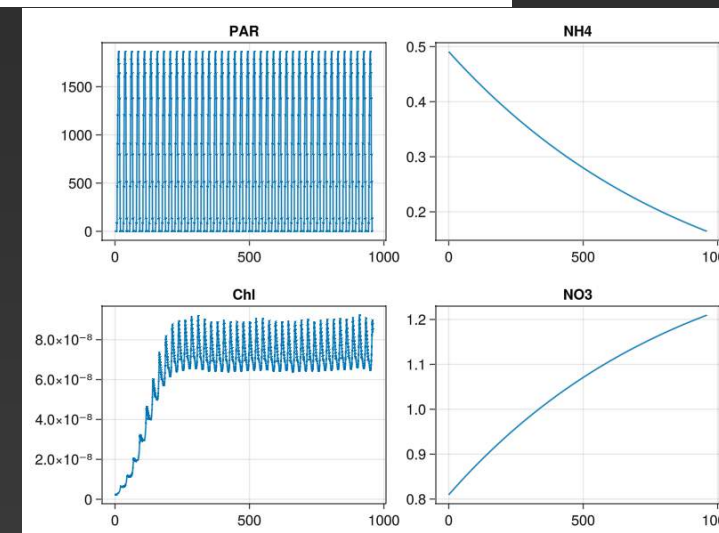
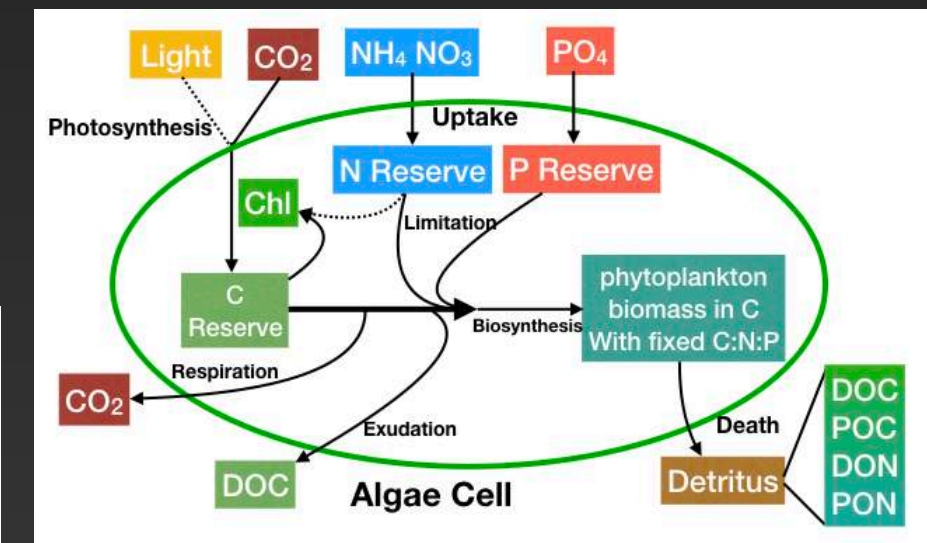
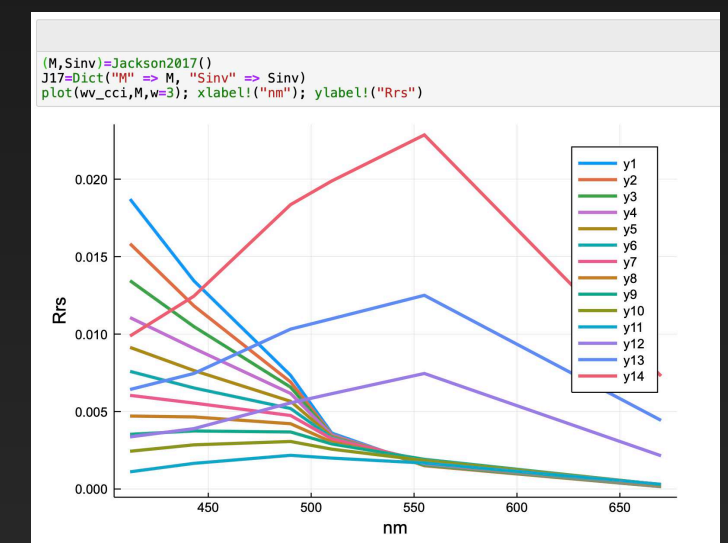
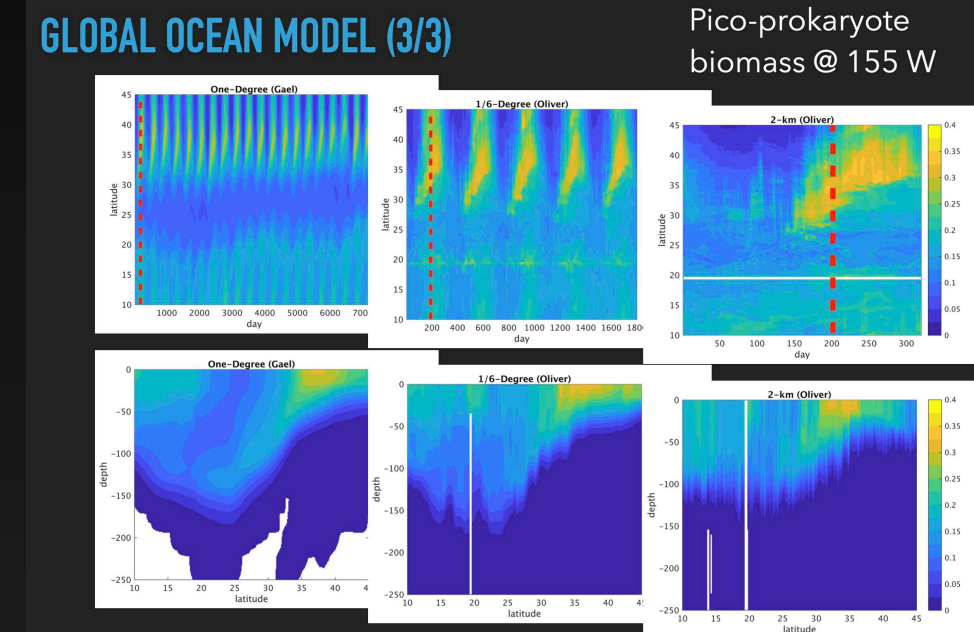
- Hector Global Climate (C++)
- The Four Scenarios
- Model Interface
- Setup, Build, and Launch
- Read Output And Plot
- Inspect Model Parameters
- Modify Parameters & Rerun
- Workflow Log (git log)

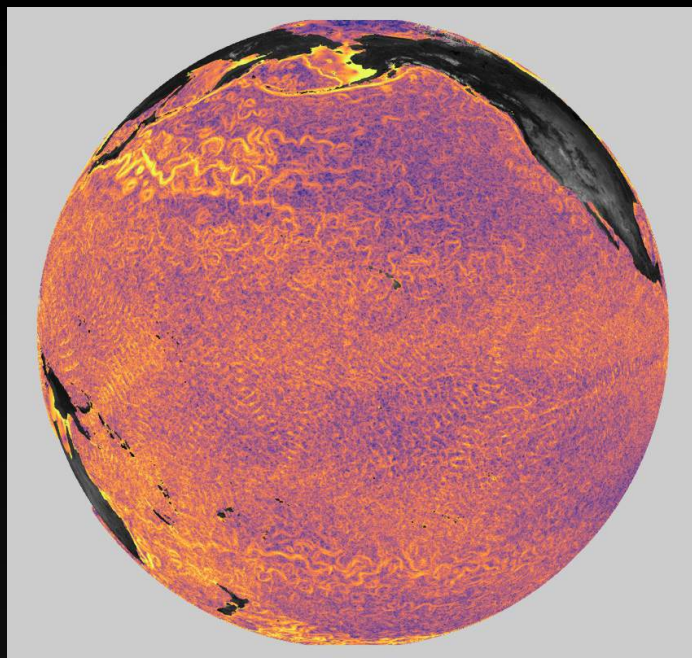
Update & Rerun Model

```
"/tmp/08859f55-0407-4437-b21d-a4ee47160809/hector/inst/input/custom_parameters.nml"
```

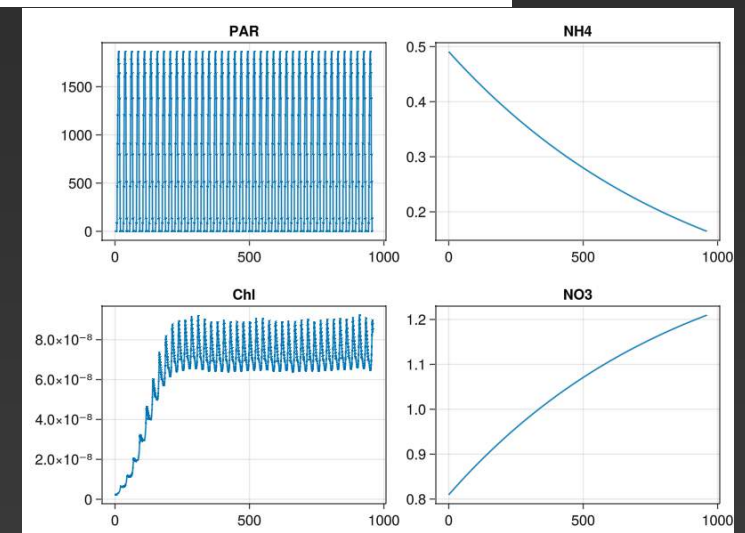
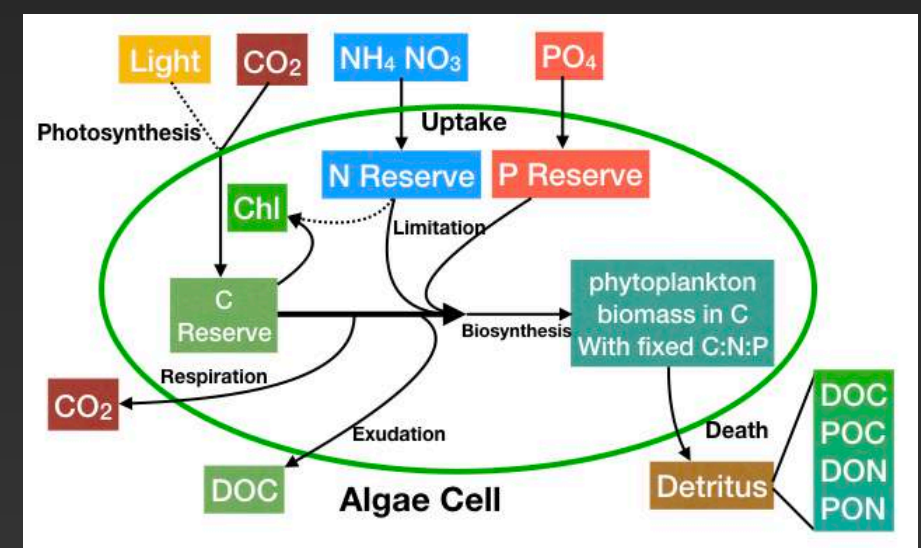
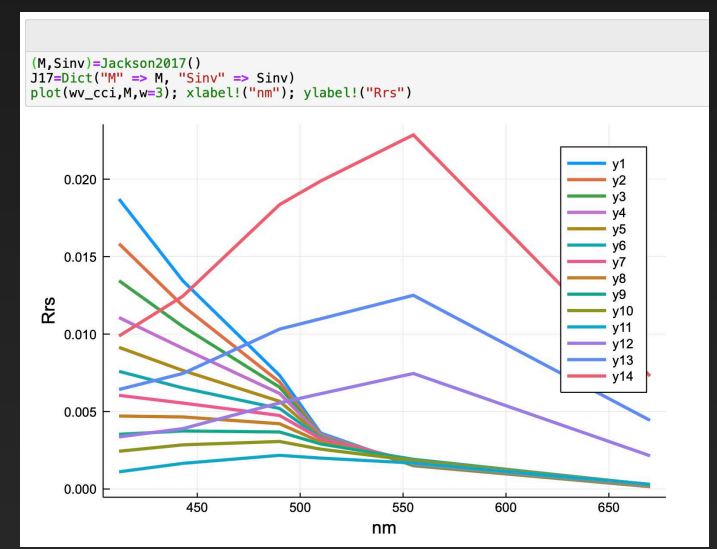
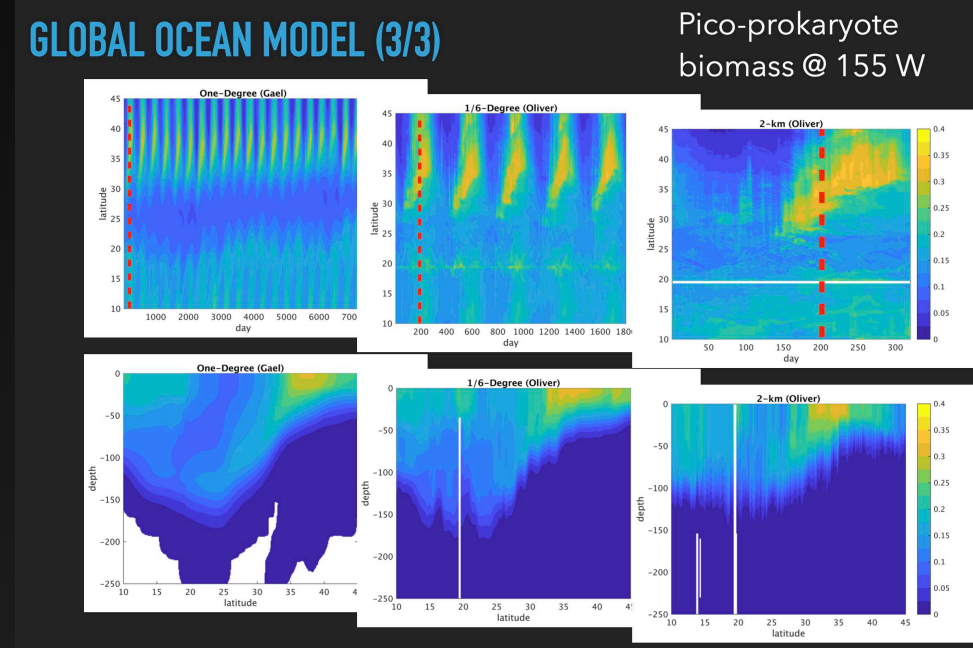
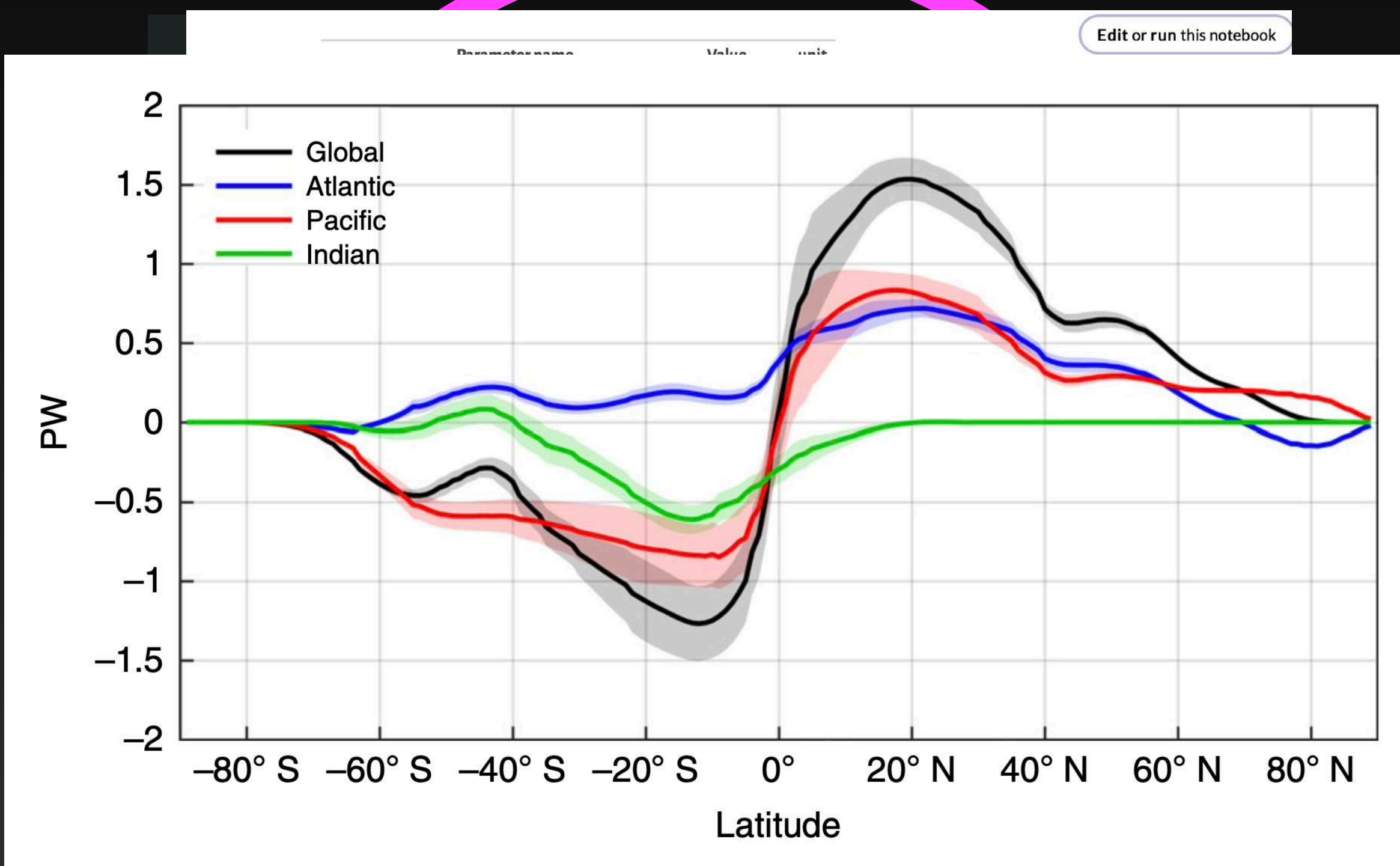
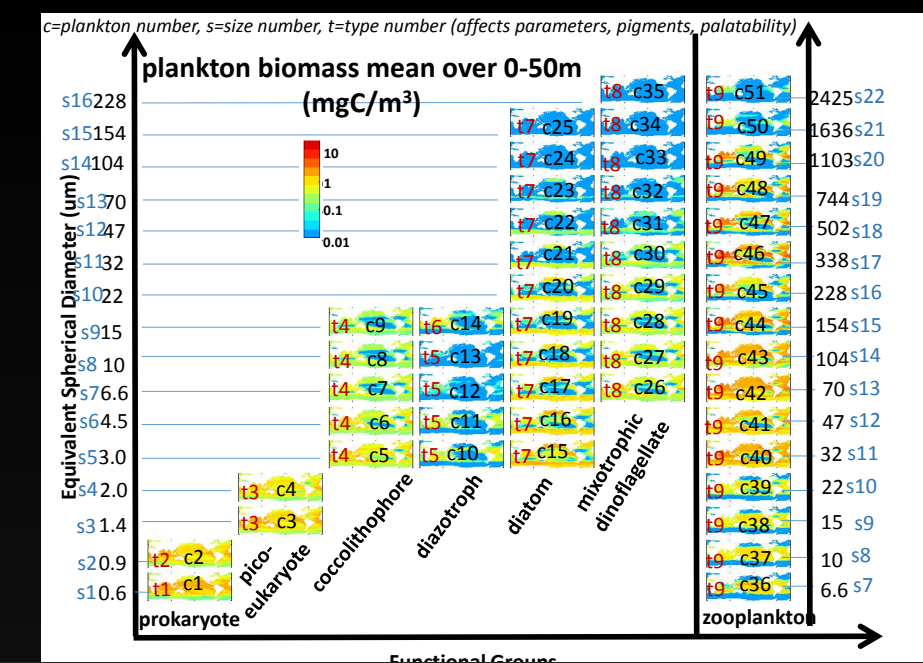
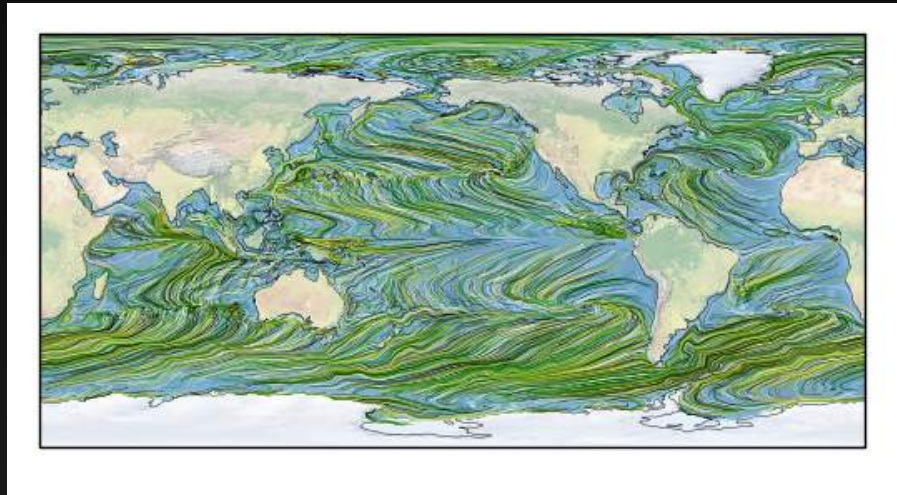
Would rerun with parameters : (0.36, 1.0, 2.3, 3.5, 276.0896713615024, 2.0, 1.0)

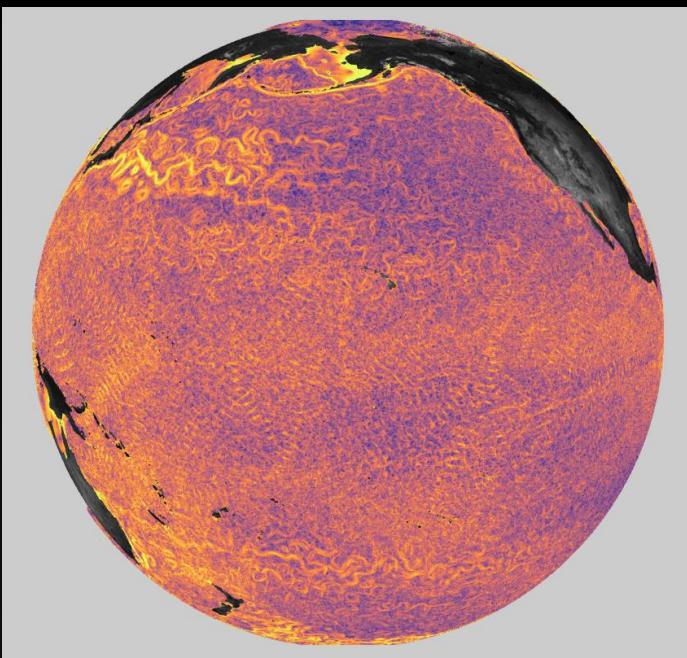
"rerun completed"



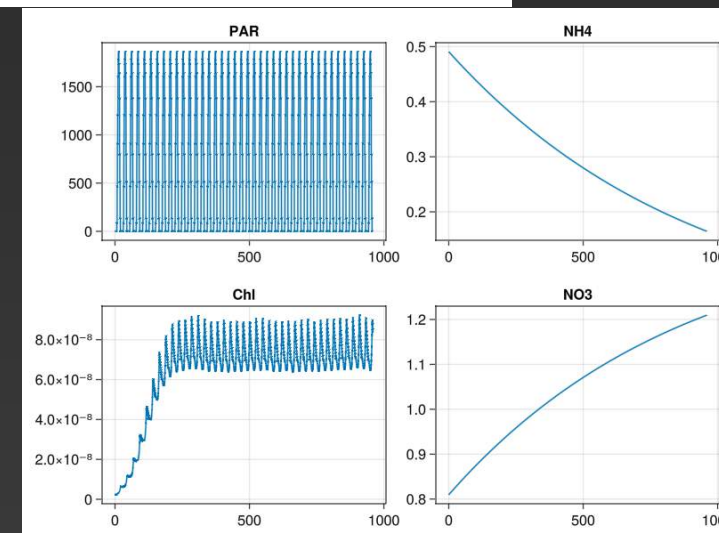
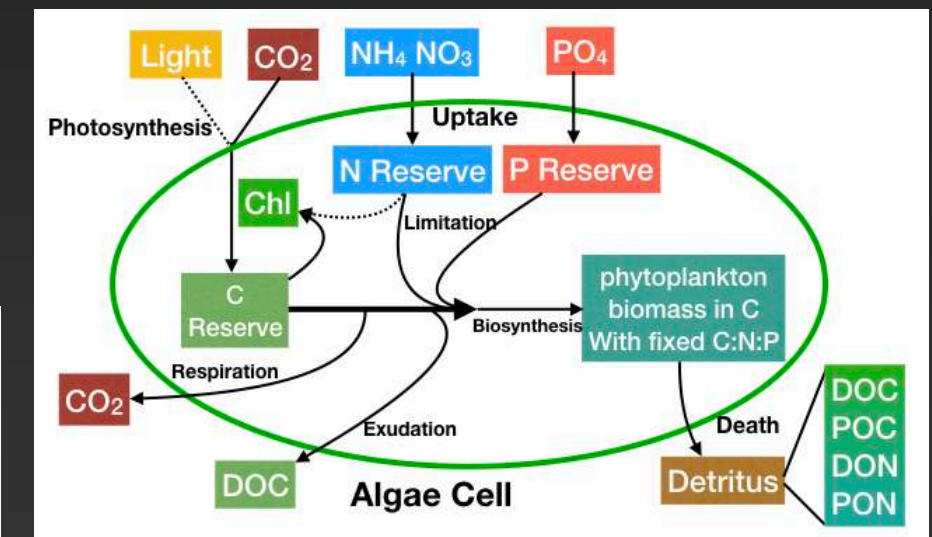
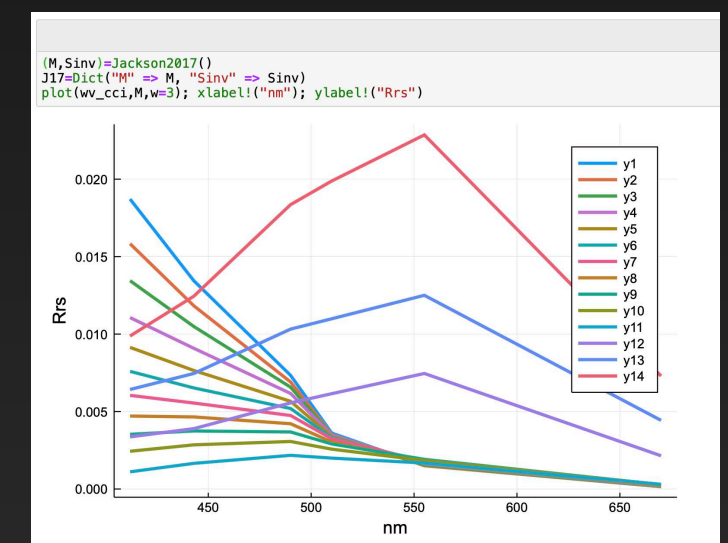
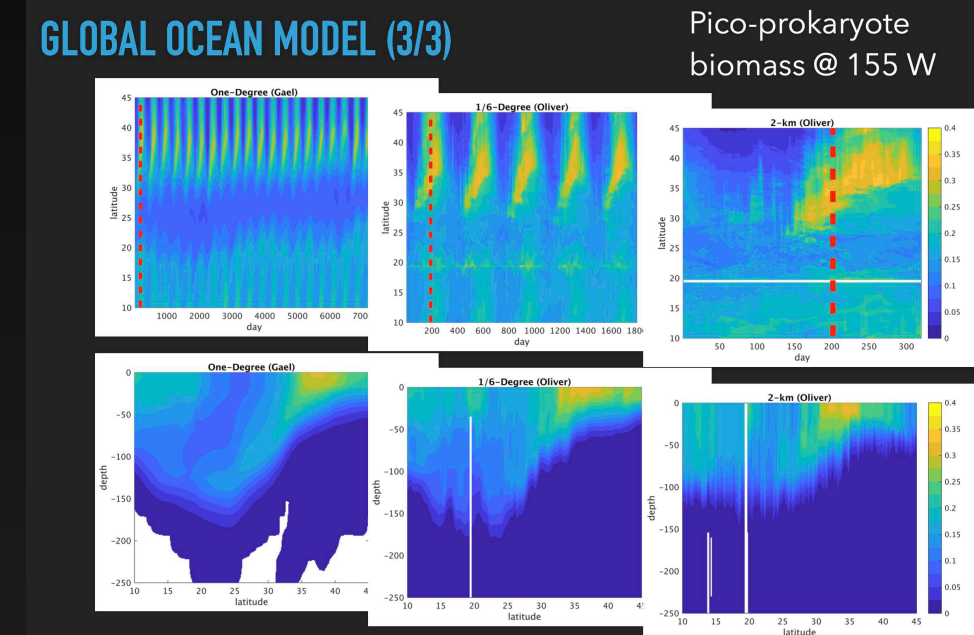
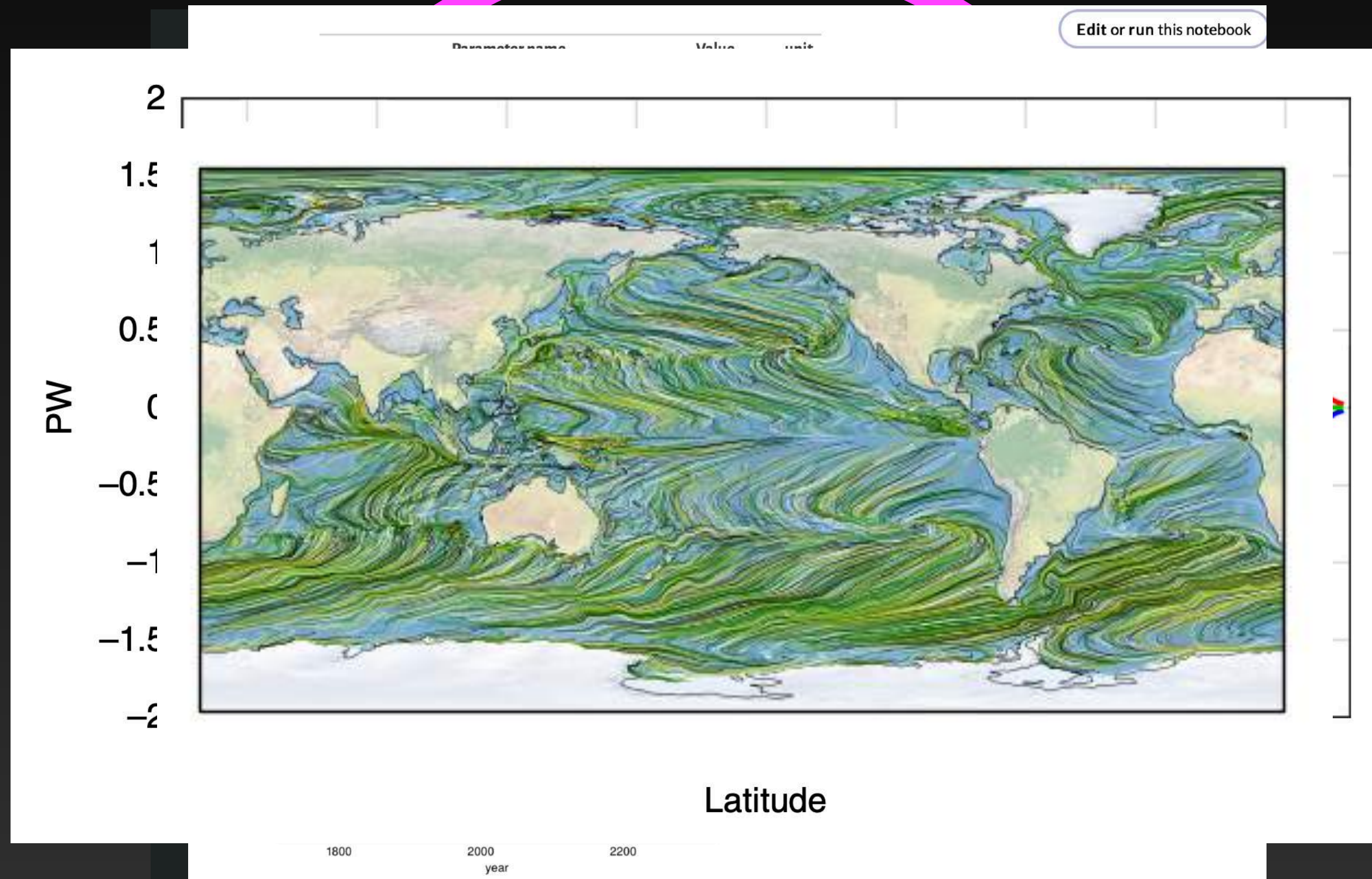
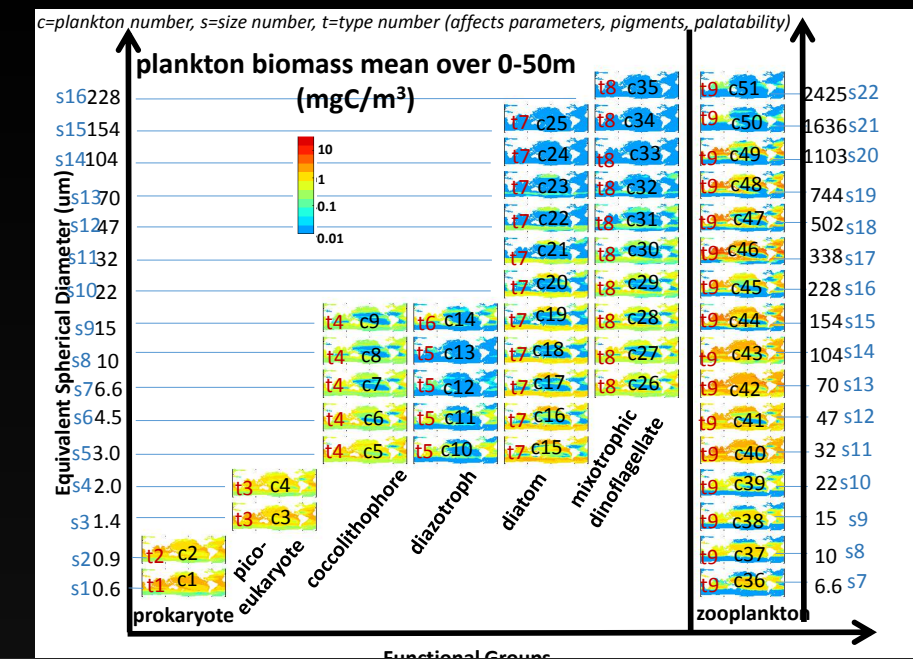


Model Hierarchy

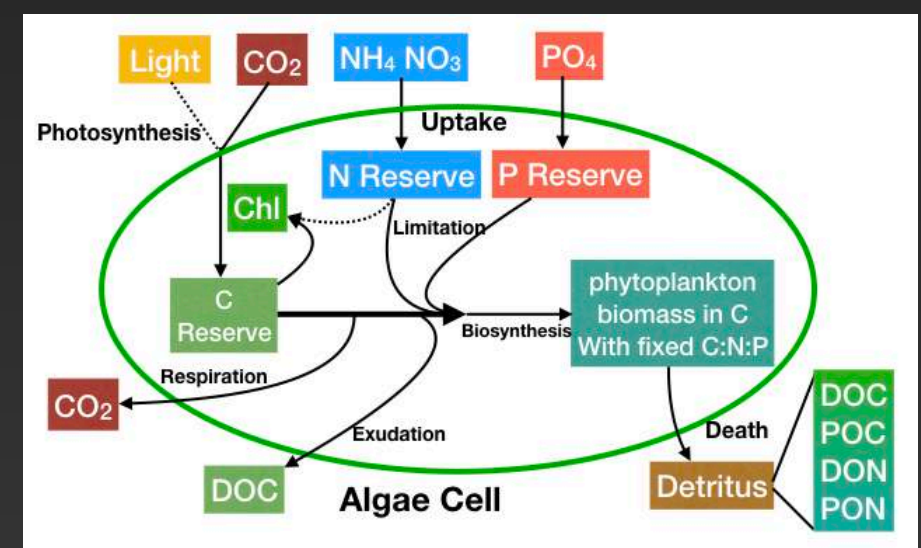
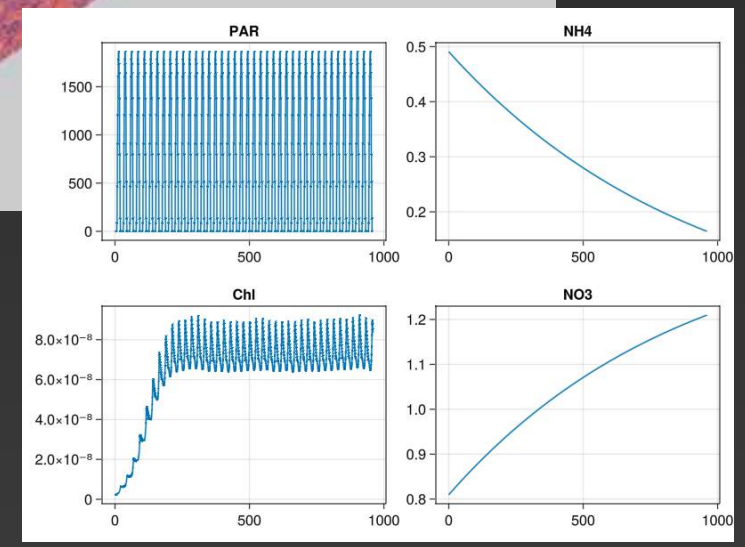
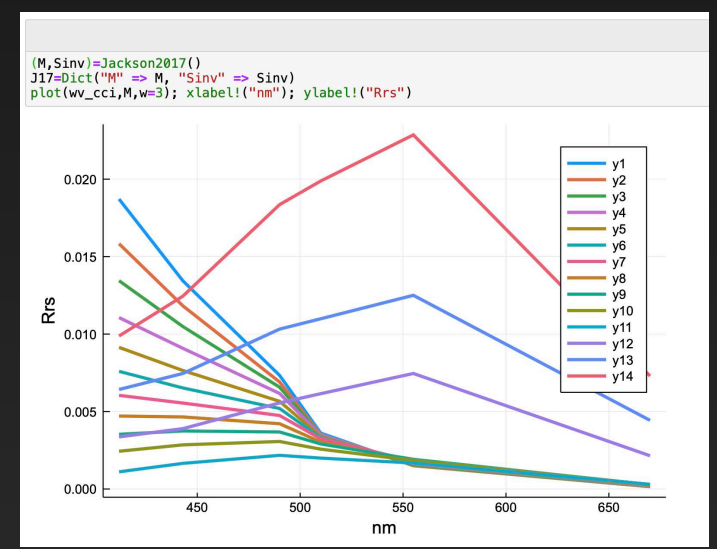
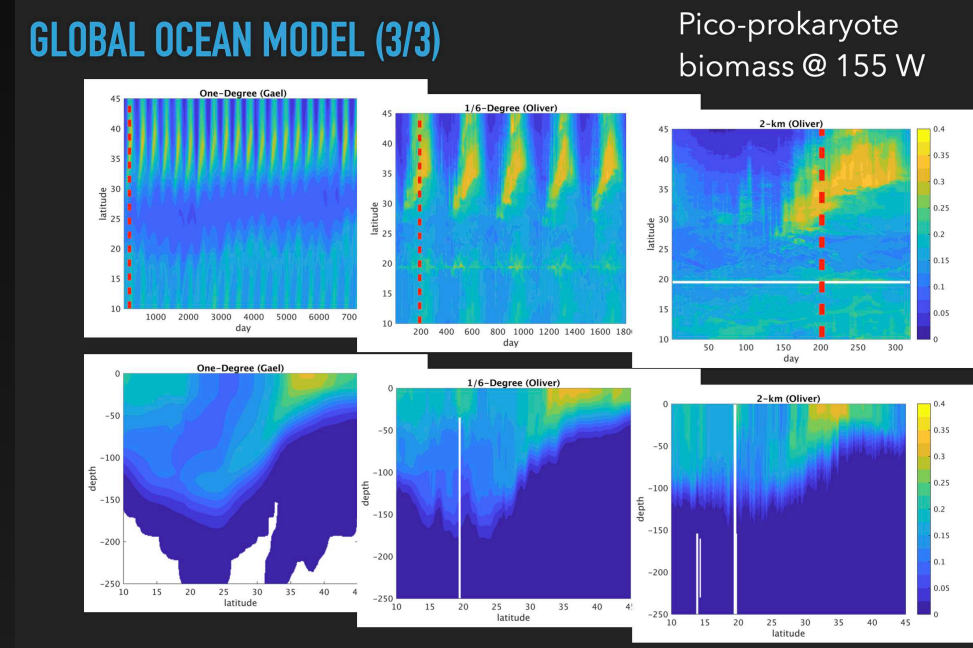
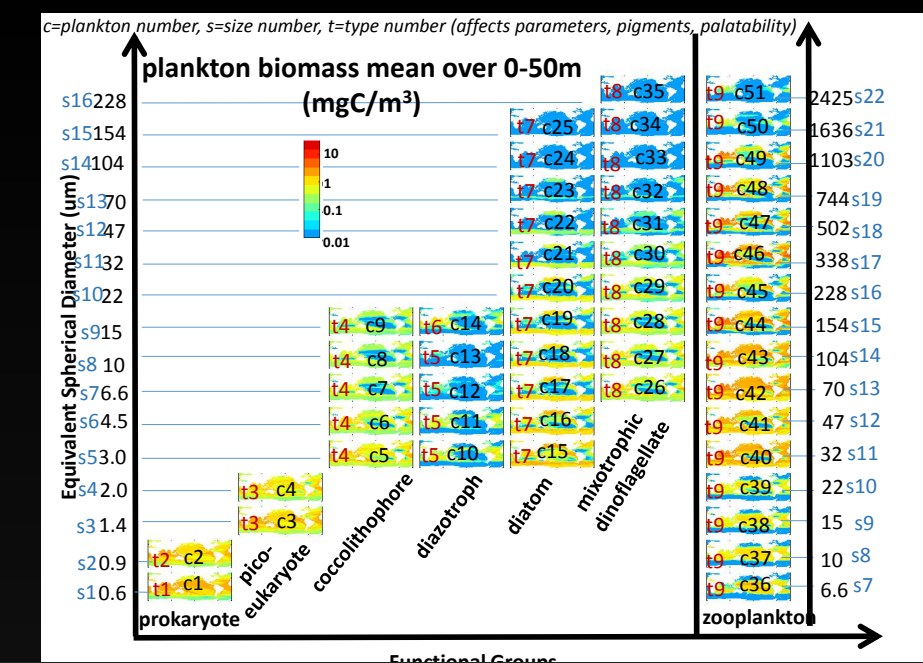
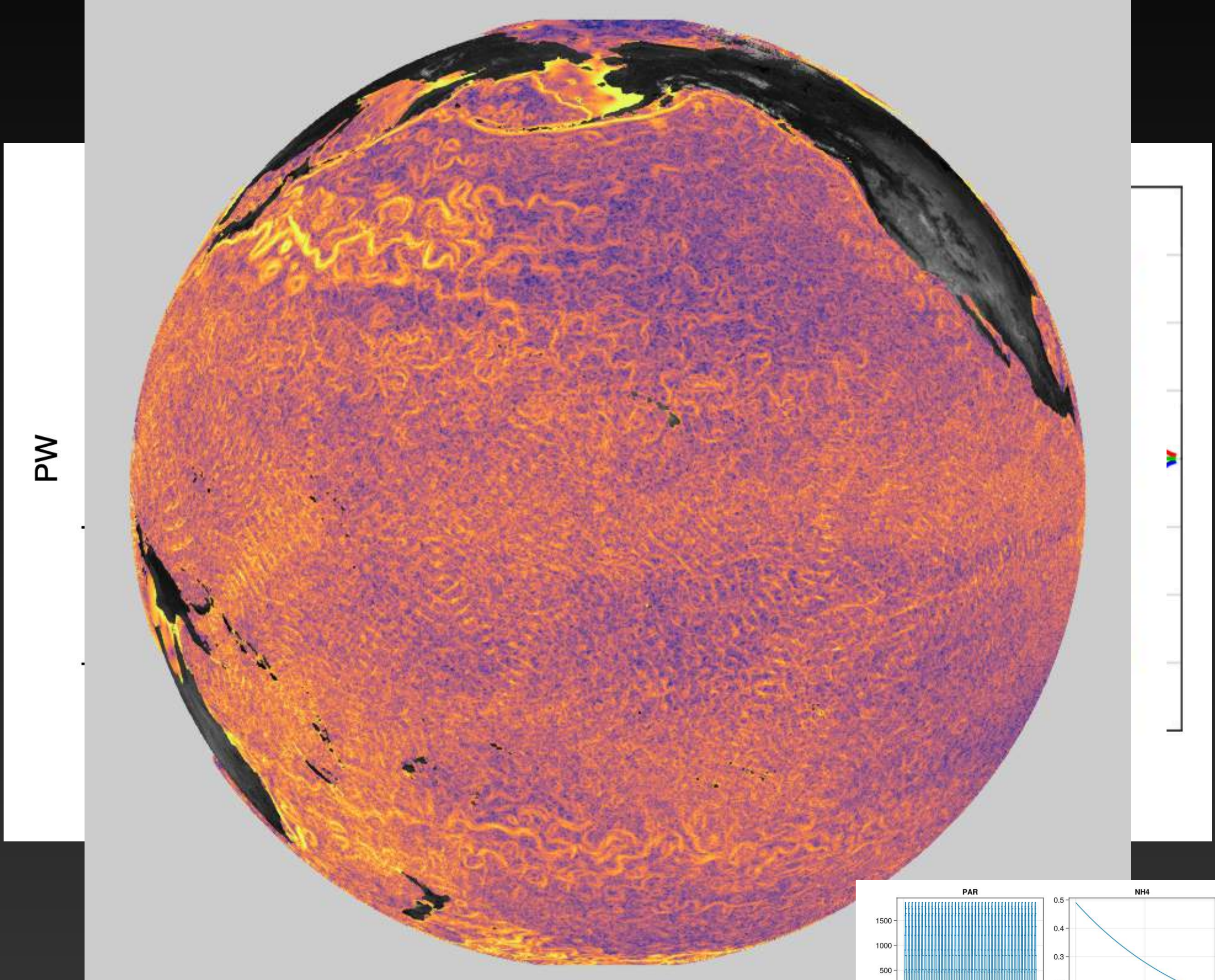




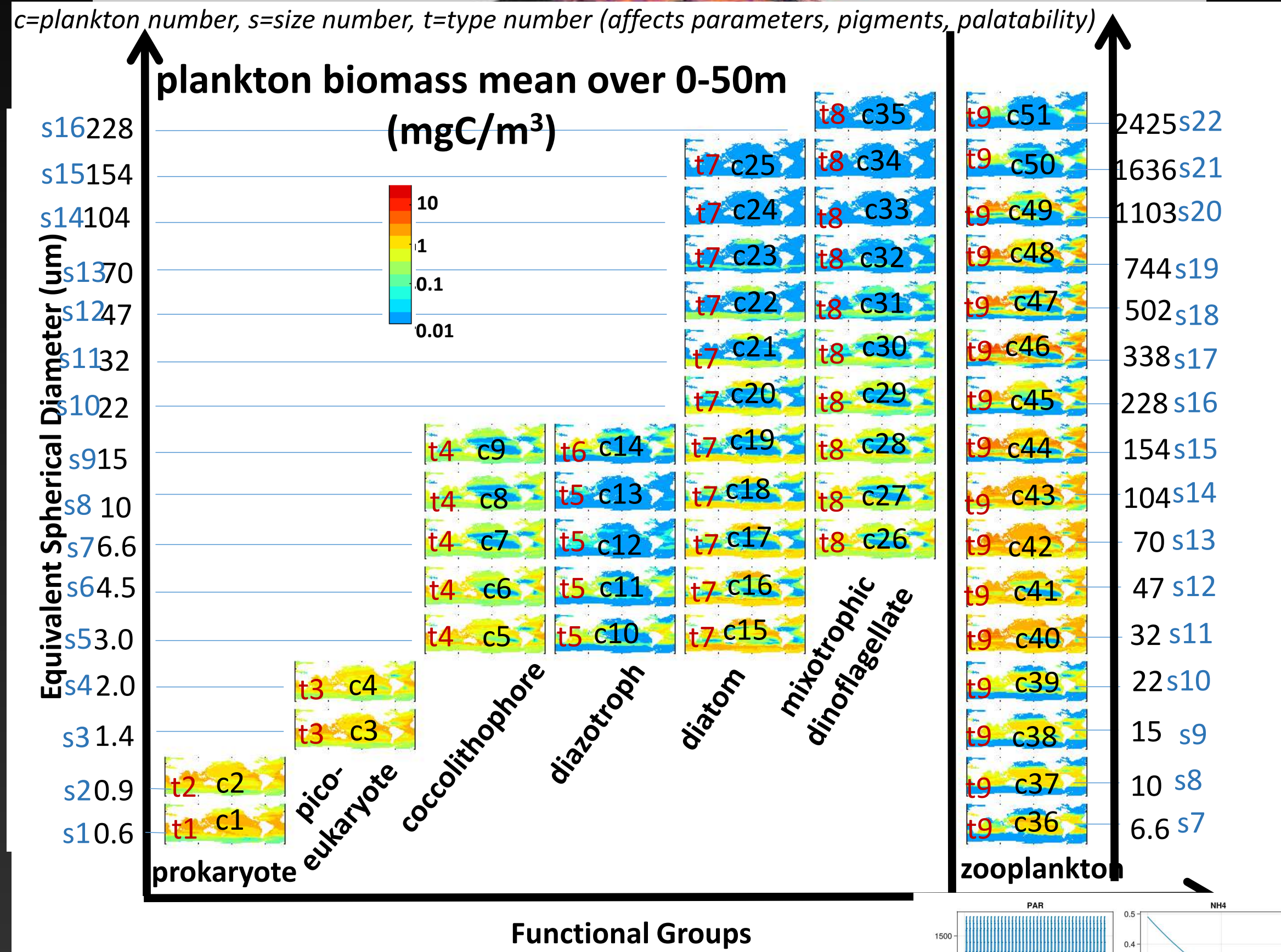
Model Hierarchy



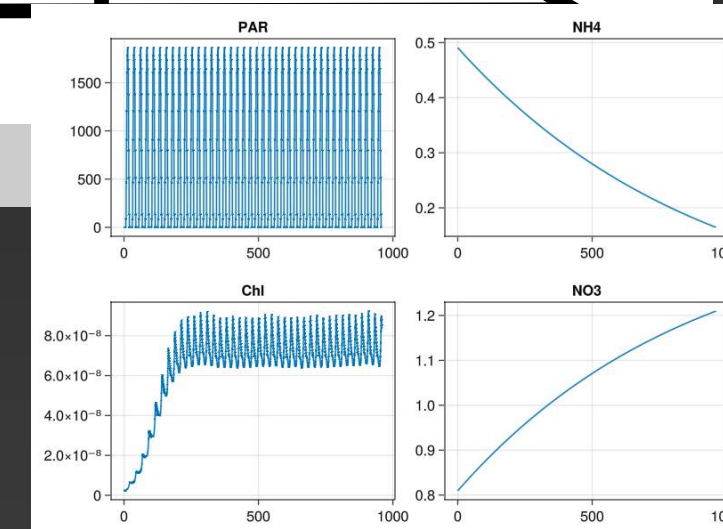
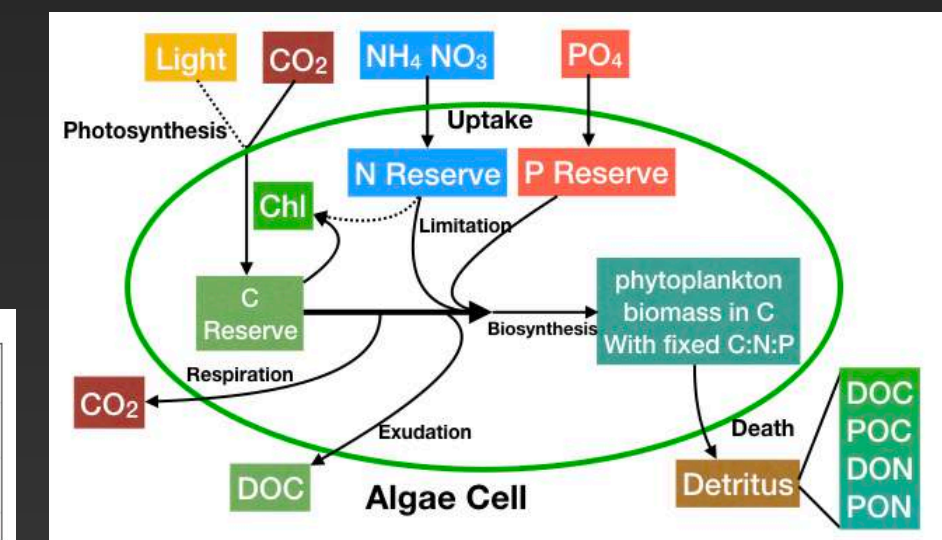
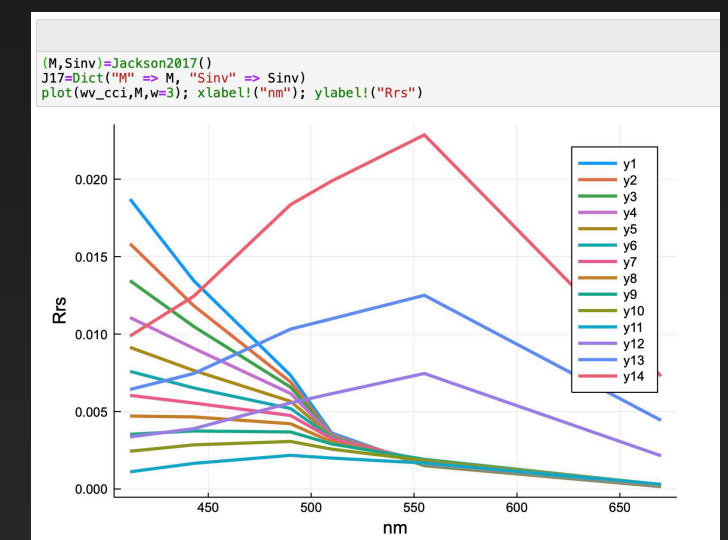
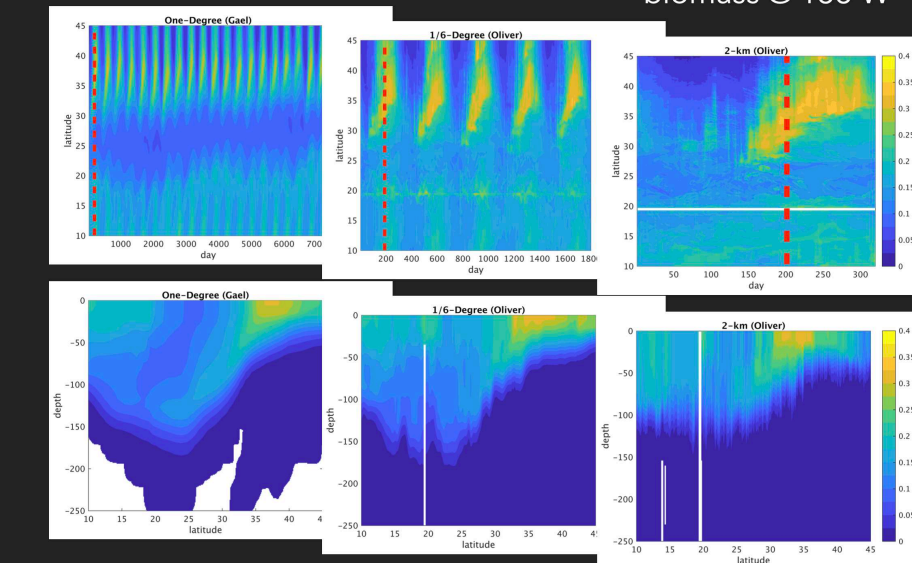
Model Hierarchy



Model Hierarchy



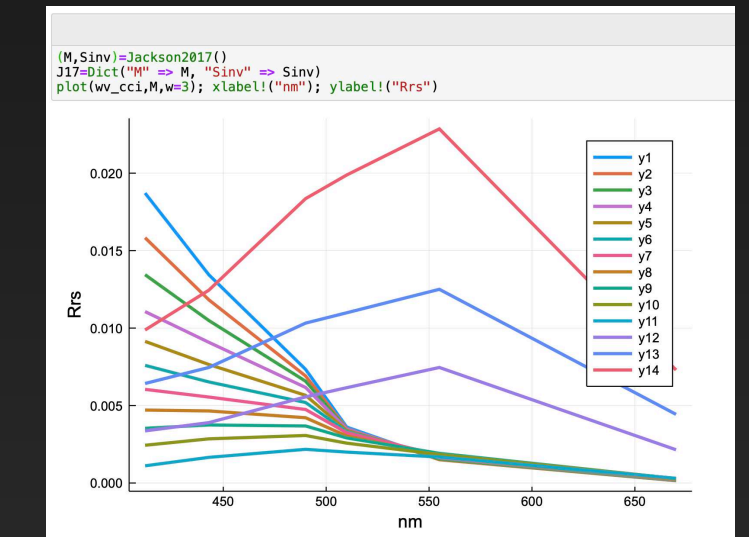
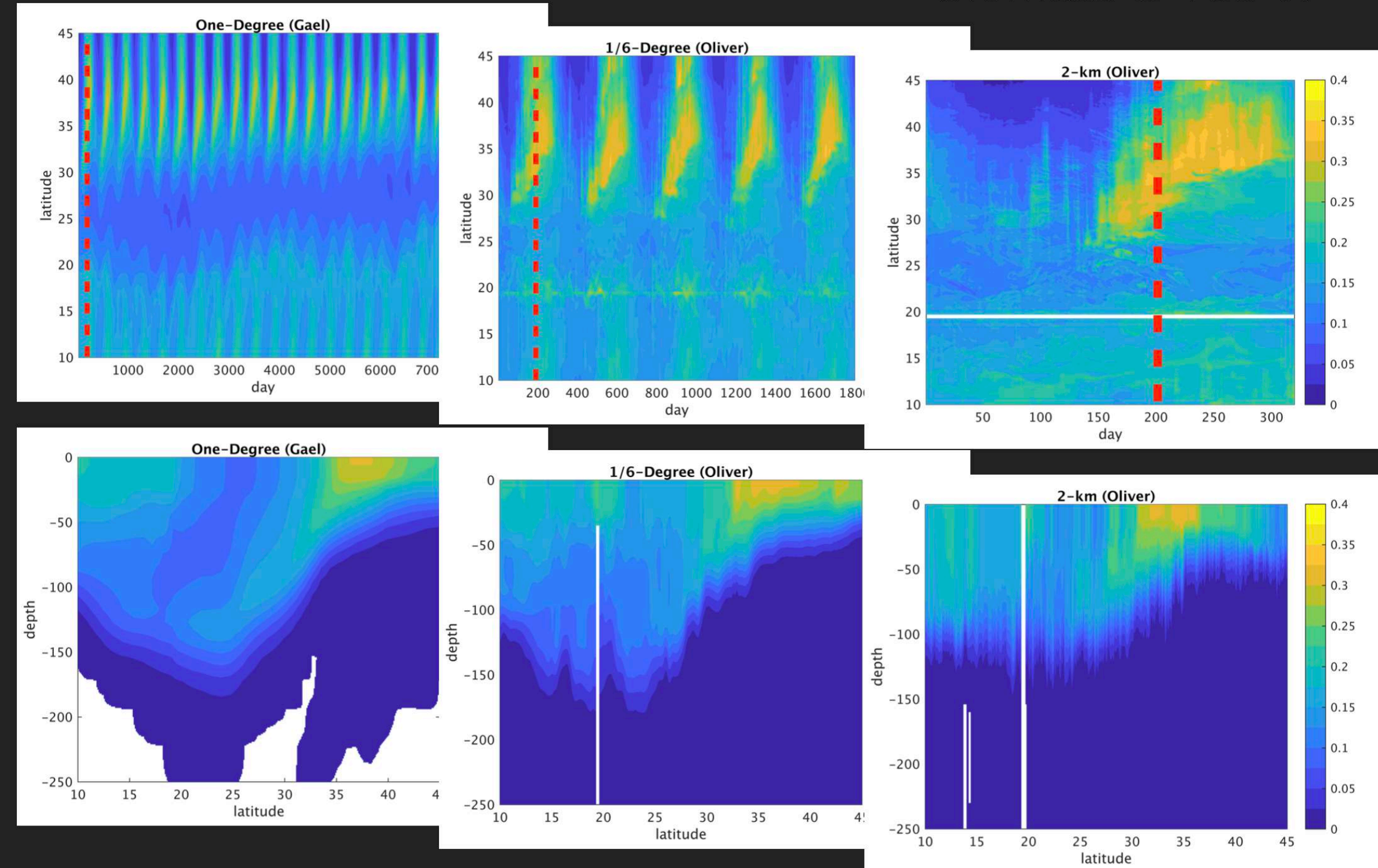
GLOBAL OCEAN MODEL (3/3)



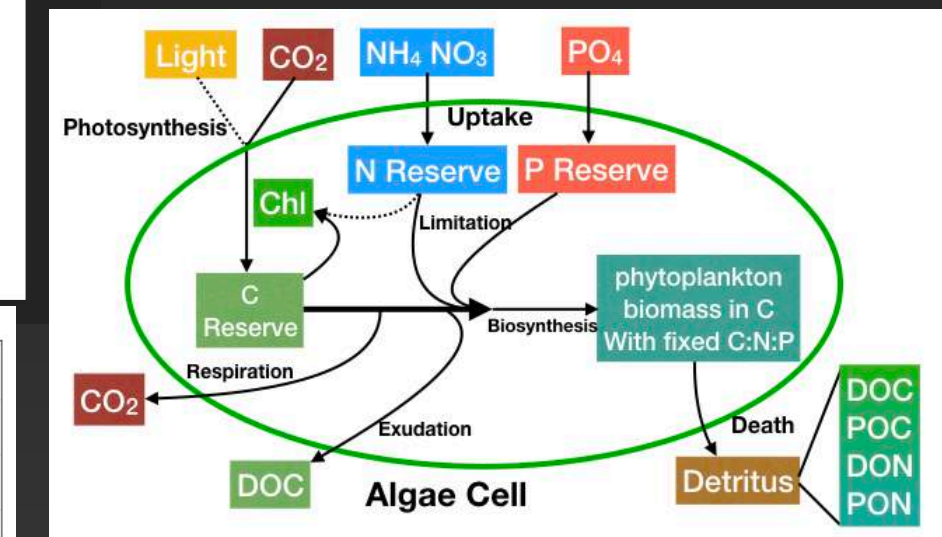
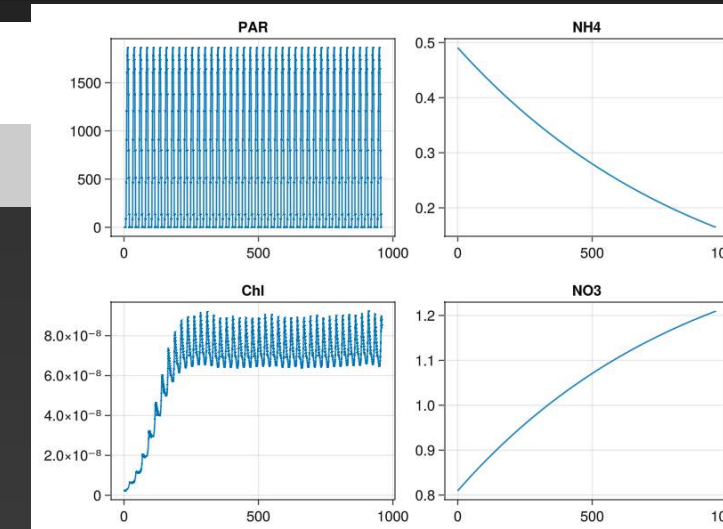
Model Hierarchy

GLOBAL OCEAN MODEL (3/3)

Pico-prokaryote biomass @ 155 W



Functional Groups

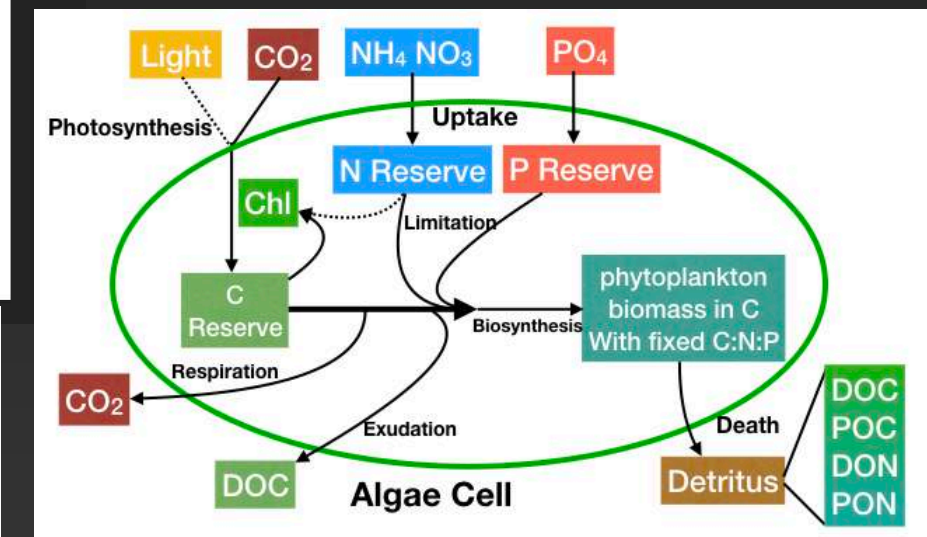
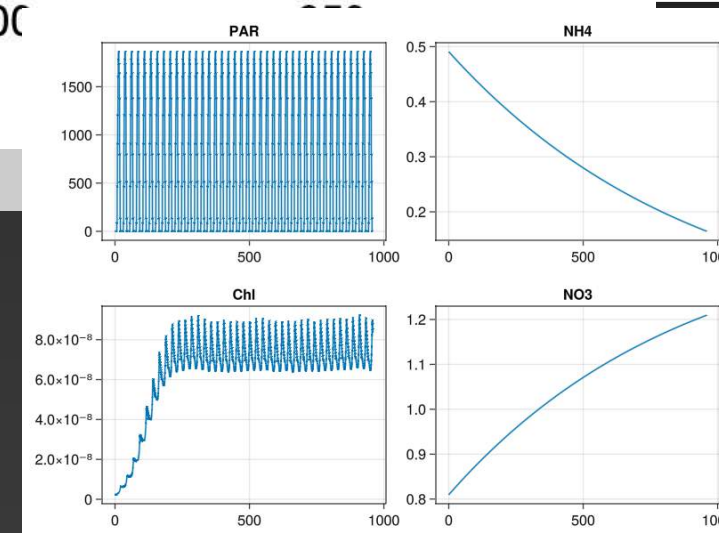
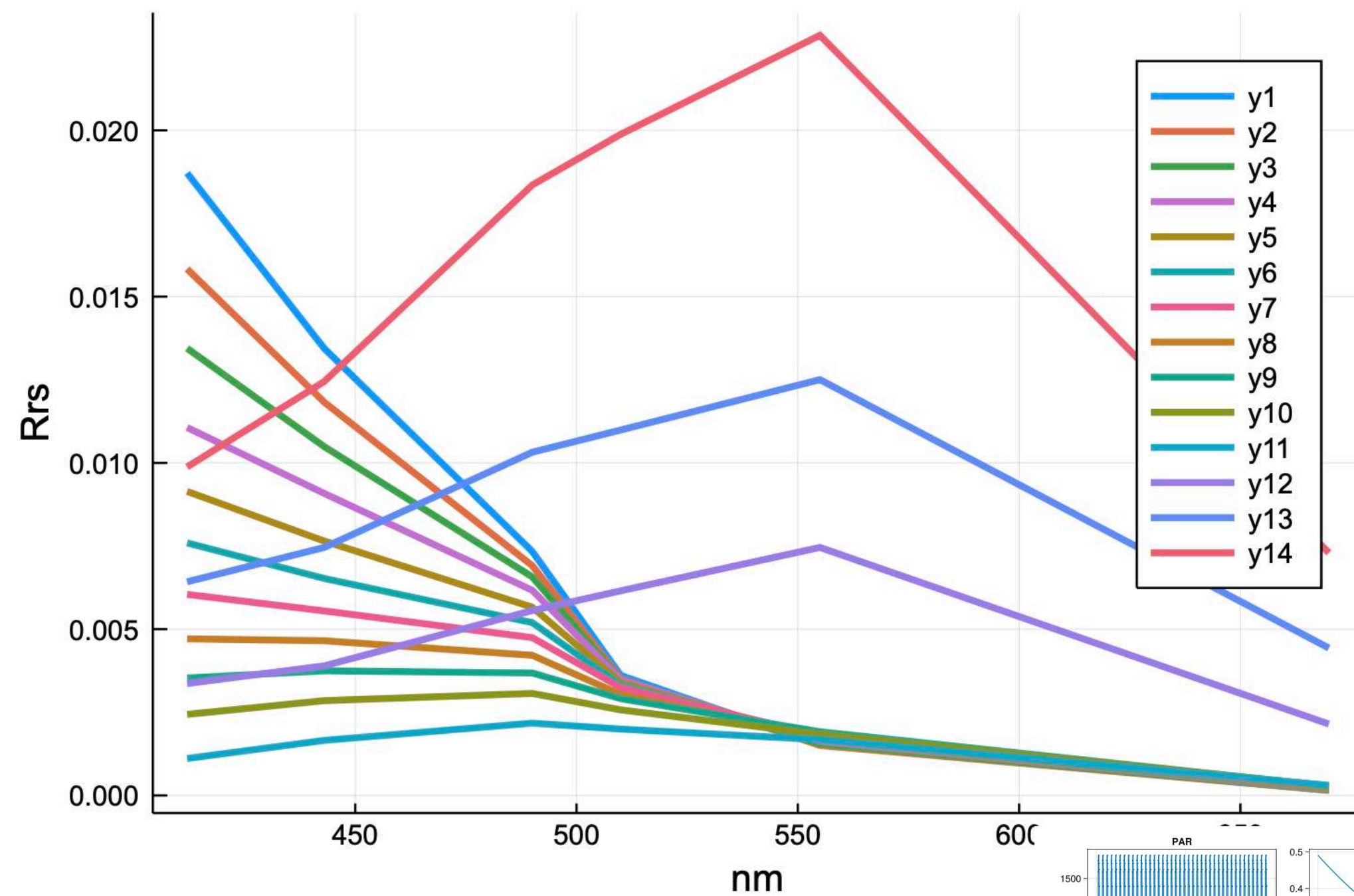


Model Hierarchy

GLOBAL OCEAN MODEL (GOM)

Pico-prokaryote

```
(M,Sinv)=Jackson2017()
J17=Dict("M" => M, "Sinv" => Sinv)
plot(wv_cci,M,w=3); xlabel!("nm"); ylabel!("Rrs")
```

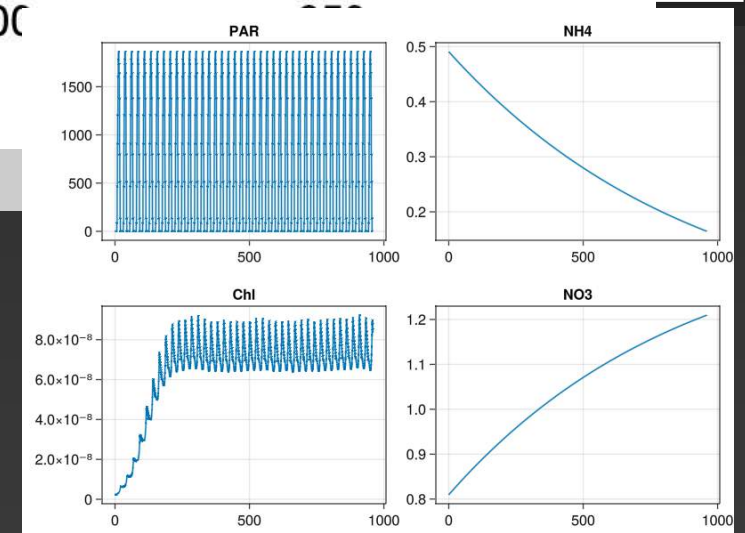
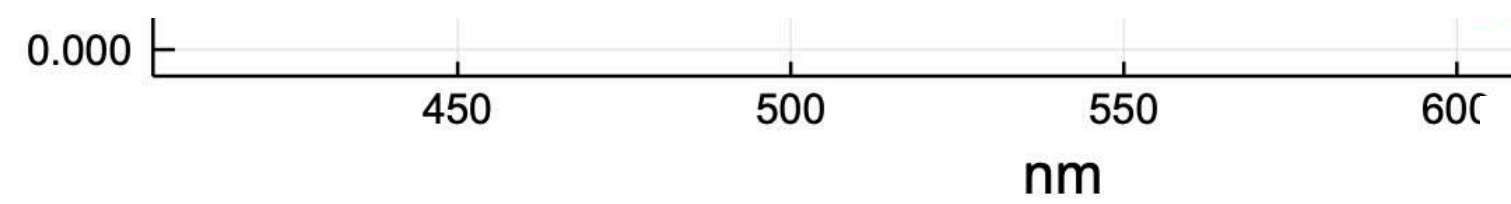
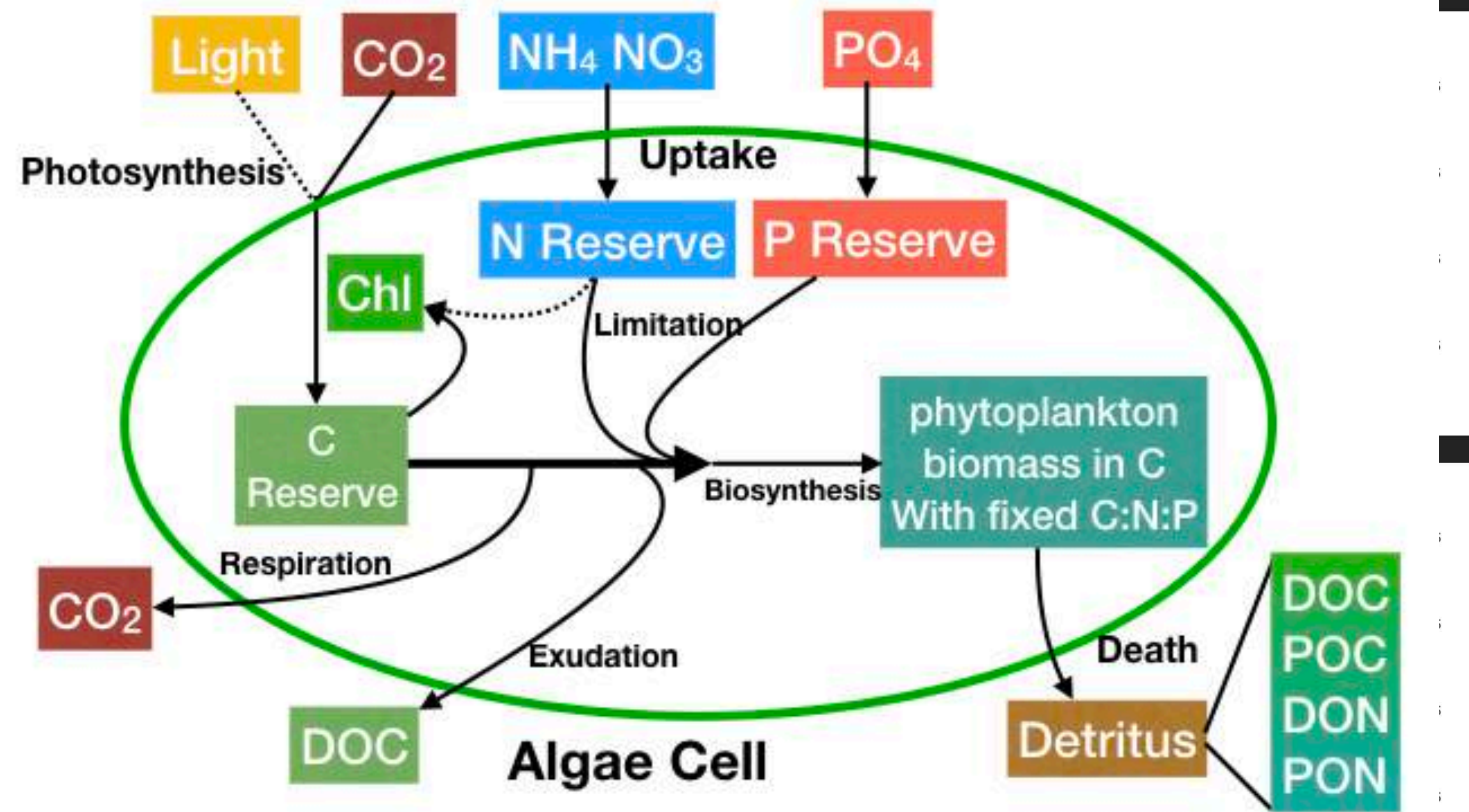


Model Hierarchy

GLOBAL OCEAN MODEL (GOM)

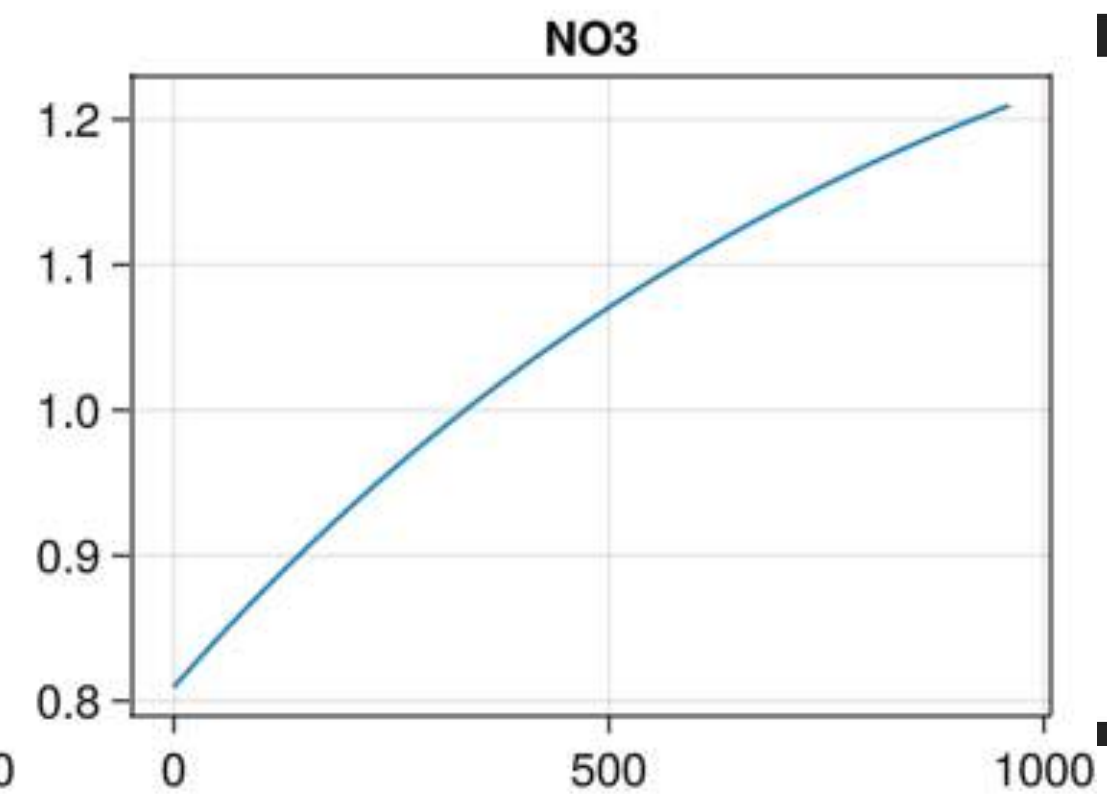
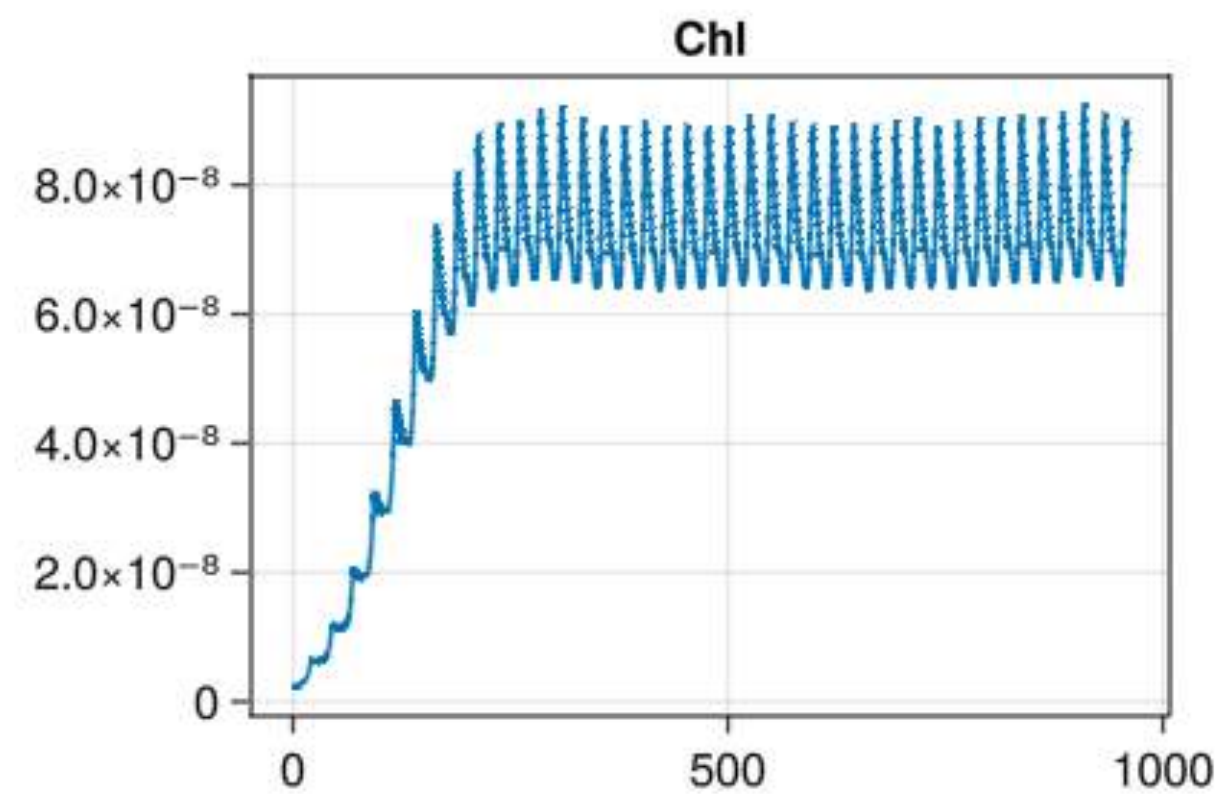
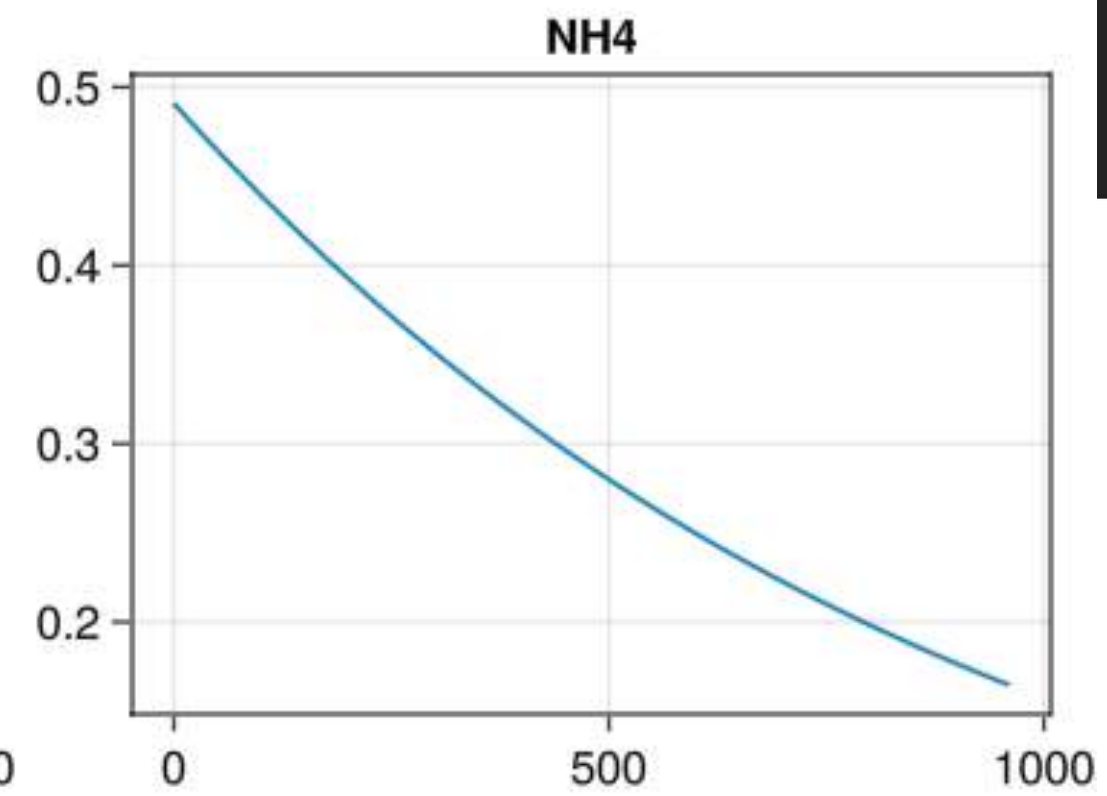
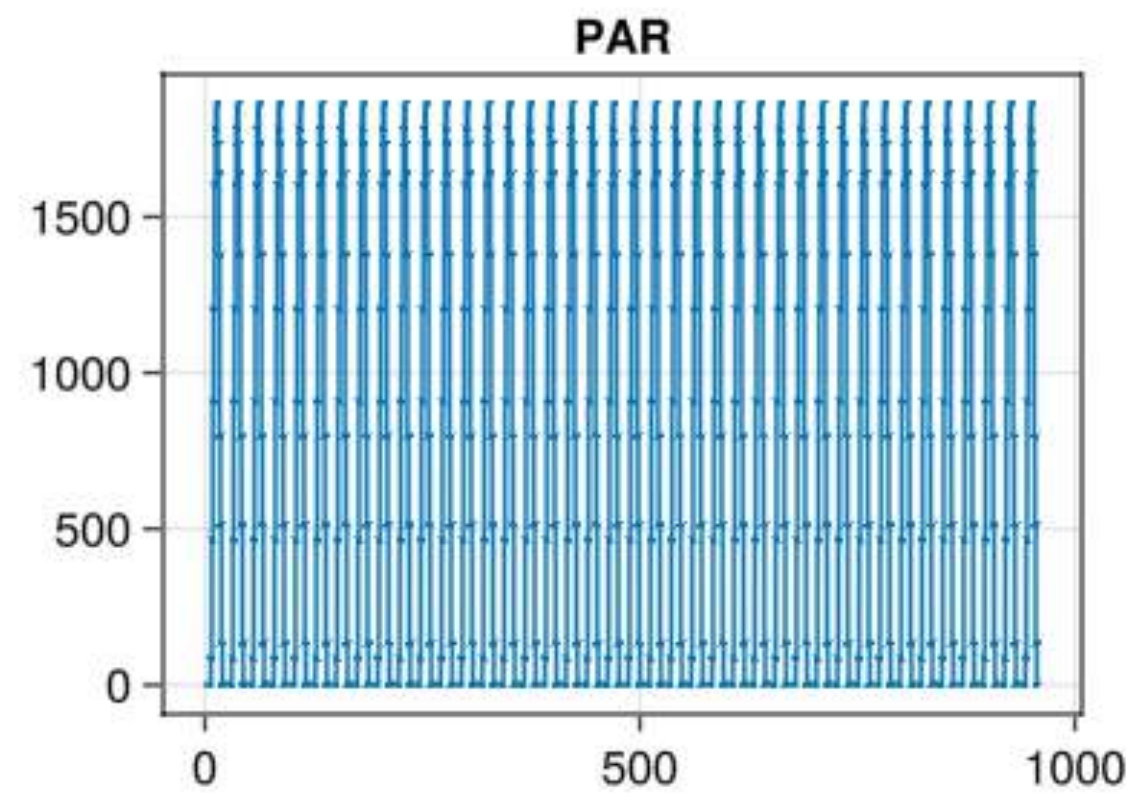
Pico-prokaryote

(M. G. ... 2017)



Model Hierarchy

GL



Models (1)

Eulerian, Lagrangian, Physics, Optics, Chemistry, Biology, AI

▼ Modeling (Julia)

- [ClimateModels.jl](#) 📖 : uniform interface to climate models of varying complexity and completeness stars 35
- [MITgcmTools.jl](#) 📖 : framework to interact with MITgcm (setup, run, output, plot, etc) stars 17
- [MeshArrays.jl](#) 📖 : gridded Earth variables, domain decomposition, and C-grid support stars 35
- [IndividualDisplacements.jl](#) 📖 : trajectory simulations for point particles in Ocean, Atmosphere, etc stars 32
- [PlanktonIndividuals.jl](#) 📖 : simulate the behaviors of an ensemble of phytoplankton individuals stars 21

- Packages (e.g., using ClimateModels.jl)
- Notebooks (see docs & example folders)

Modeling Ocean Robots

- **Vessels**

- [IndividualDisplacements.jl](#) for pathway simulations

- $\frac{dx}{dt} = u$ from ECCO + km-scale model

- $\frac{dx}{dt} = u + v$ to include floatability, ballasting, ...

- **Sensors**

- [MeshArrays.jl](#) for sampling gridded model variables (pointwise or integrated)

- Any derivative of variables resolved in CBIOMES-global

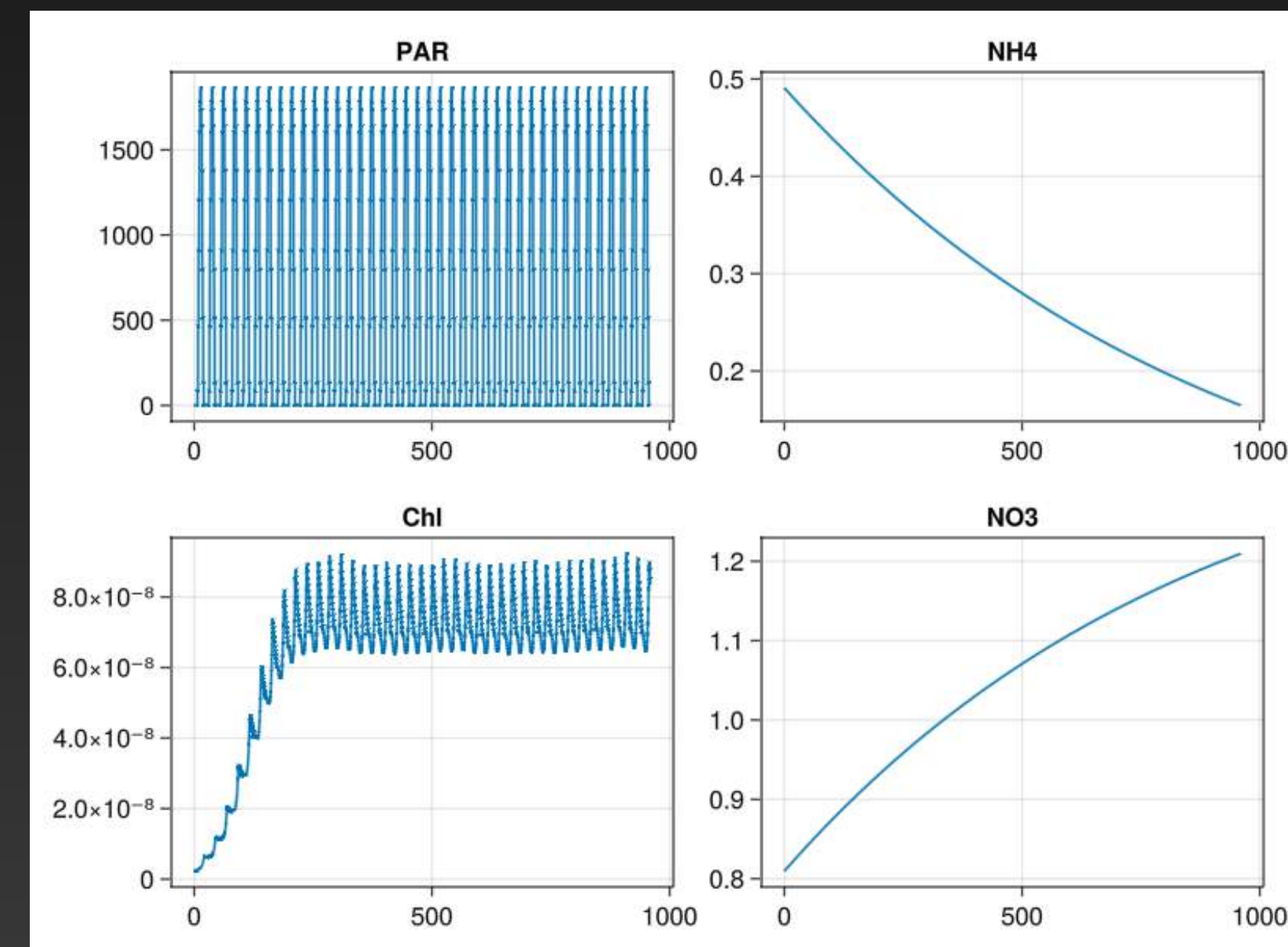
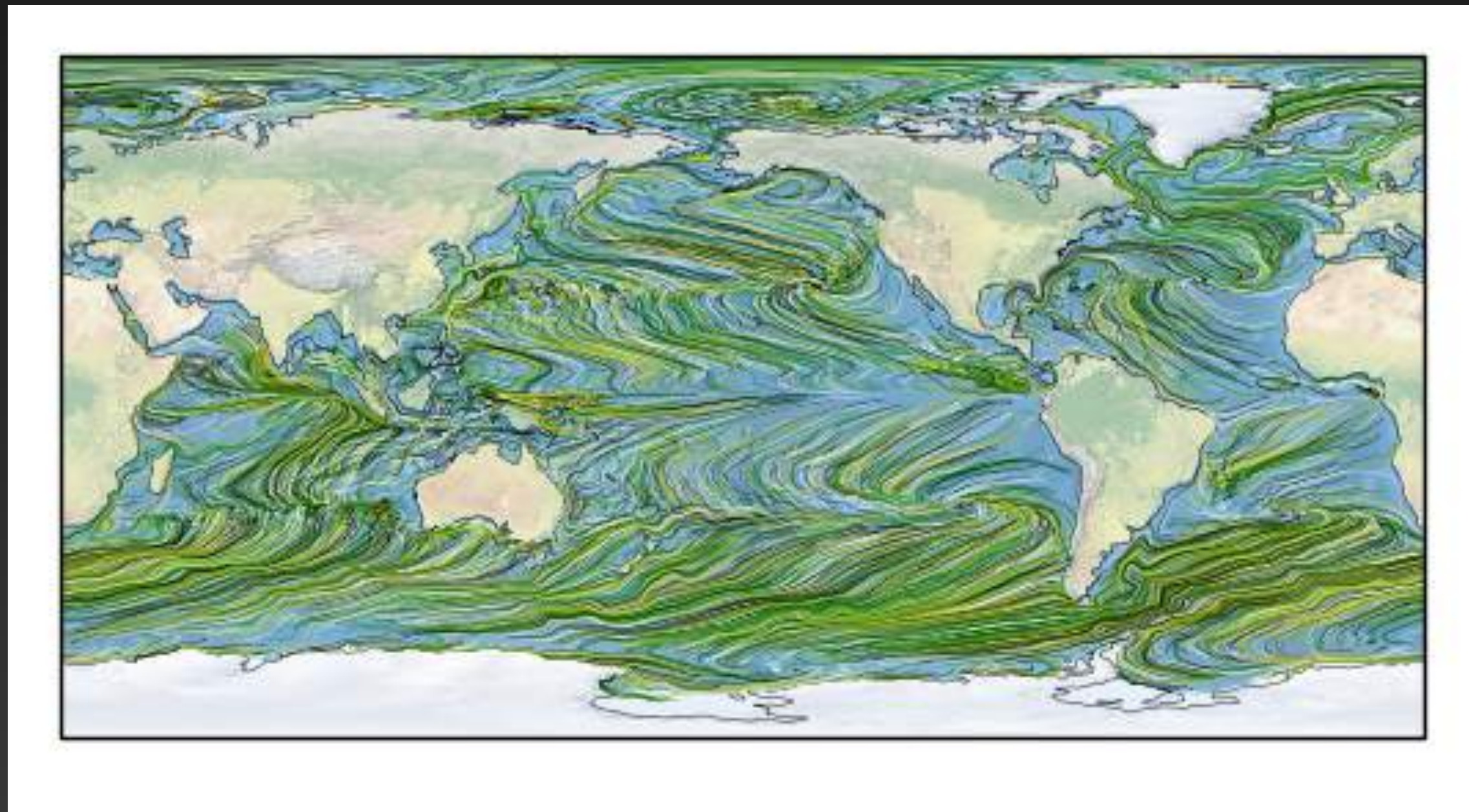
- optics, chemistry, biomass, EKE, spectral components, hyperspectral, ...

- Chemostat, local NPZD, [PlanktonIndividuals.jl](#) for aquacoscms for generally

- Trained classifiers & AI, MCMC, statistical models, optimal control, ...

Use Case Example

- aquacosm on drifters or floats = growth rates
- Global CBIOMES model ▶▶ currents ▶▶ vessel path simulation (here ~ drifter) ▶▶ sample environmental conditions ▶▶ agent based model
- Research : rate parameters, diel cycle vs internal waves effects, etc



Digital Twin Infrastructure

👉 <https://github.com/gaelforget> 👈

Digital Twin Infrastructure

- ✓ Ability to run, combine, perturb, and observe models (ClimateModels.jl, ...)
 - all but global km-scale model are fairly cheap to run
- ✓ Output files & archiving (Dataverse.jl, NCDatasets.jl, ...)
- ✓ Analysis, visualization (Makie.jl, Pluto.jl, ...)
- ✓ Interactivity (Pluto.jl, Observables.jl, ...)
- ✓ Cloud / On-Premise computing / On-Observing Vessel (Docker, GitHub, ...)
- x Web apps and APIs (Dash.jl, JSServe.jl, ...)

👉 <https://github.com/gaelforget> 👈

Models (2)

Eulerian, Lagrangian, Physics, Optics, Chemistry, Biology, AI

▼ Modeling (Other)

- [MITgcm](#) : *M.I.T. general circulation model* master code and documentation stars 265
- [ECCOv4](#) : Ocean state estimation framework, and model configuration stars 17
- [ECCO-Docker](#) : virtual machine to analyze and run ECCO solutions stars 3
- [gcmfaces](#) : *Matlab / Octave* toolbox that handles gridded Earth variables in generic fashion stars 20

- Packages (e.g., using ClimateModels.jl)
- Notebooks (see docs & example folders)

ECCO-Docker

Already on EIS, deploy in cloud, or use locally anywhere to run models

The screenshot displays the ECCO-Docker web interface. On the left is a file explorer with a search bar and a table of files and folders. The main area on the right is titled 'Launcher' and contains two sections: 'Notebook' and 'Console'. Each section offers four environment options: Python 3 (ipykernel), Julia 1.7.2, Octave, and R. The 'Pluto.jl' option is also visible in the Notebook section.

| Name | Last Modified |
|---------------|----------------|
| build | 35 minutes ago |
| MITgcm | 27 minutes ago |
| plutoserver | an hour ago |
| plutoserve... | 35 minutes ago |
| src | an hour ago |
| work | 7 days ago |
| Manifest.t... | 40 minutes ago |
| Project.toml | an hour ago |
| runpluto.sh | an hour ago |
| setup.py | an hour ago |

JuliaOcean and JuliaClimate

- GitHub organizations open to community at large
- Always looking for feedback : open GitHub issues (*please do complain!*)
- Always looking for contributors to existing packages (*missing features?*)
- New packages are very much welcome (*visibility, modularity, maintain*)
- New members also (*e.g., ping us on GitHub or Julia Slack*)

