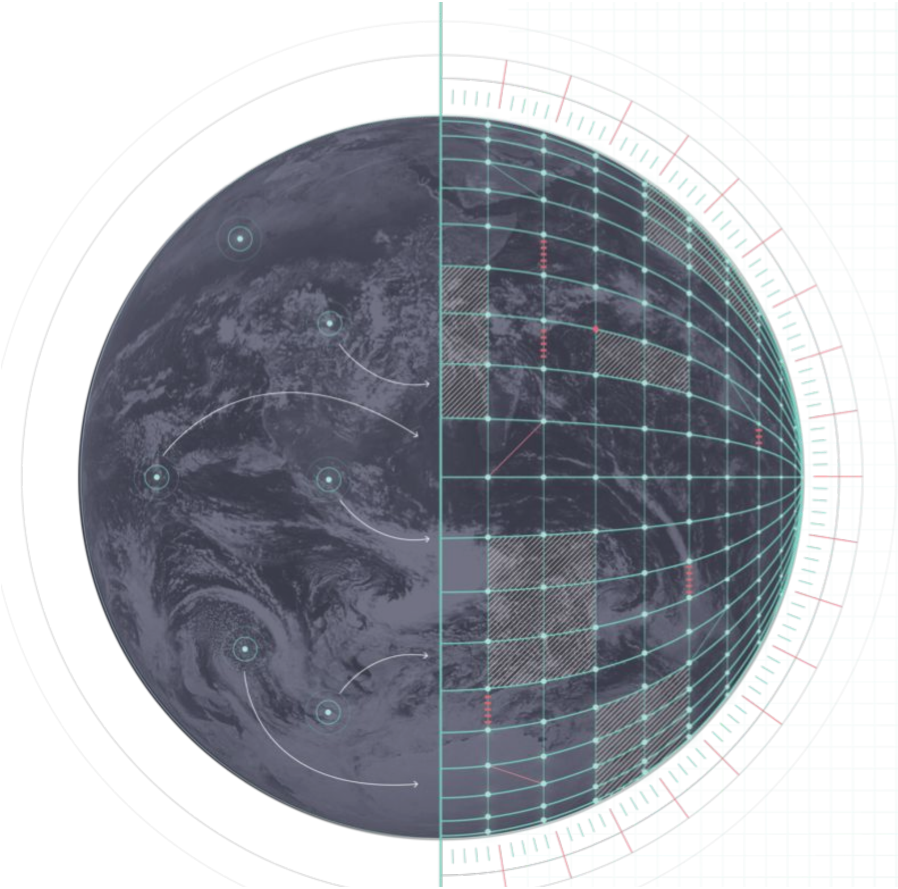


AQUA:

a novel quality assessment tool for km-scale simulations in the Destination Earth Climate Digital Twin – The Core Framework

Matteo Nurisso, J. von Hardenberg, S. Caprioli, S. Ghosh, N. Koldunov, B. Kinoshita, N. Nazarova, P. Ghinassi, P. Davini

EGU2024, Vienna 16 April 2024

Sharing is encouraged

Funded by the European Union

Destination Earth implemented by ECMWF esa EUMETSAT

The challenges of Destination Earth

The challenges of DestinE Climate DT

Three Earth System Models:

ECMWF
Integrated Forecasting System (IFS)

NEMO
Nucleus for European Modelling of the Ocean

FESOM
Finite Element Sea-ice Ocean Model

ICOsahedral Non-hydrostatic model (ICON)

Climate DT key features and challenges:

- **User-driven approach** focused on user interactivity.
- Global climate simulations at **unprecedented horizontal resolution** (5 km).
- Novel **streaming framework** of climate model output to applications (end to end workflow).
- **Quality assessment** and uncertainty quantification based on observations ⟶ **AQUA!**
- Deployment on three European **pre-exascale supercomputers**.

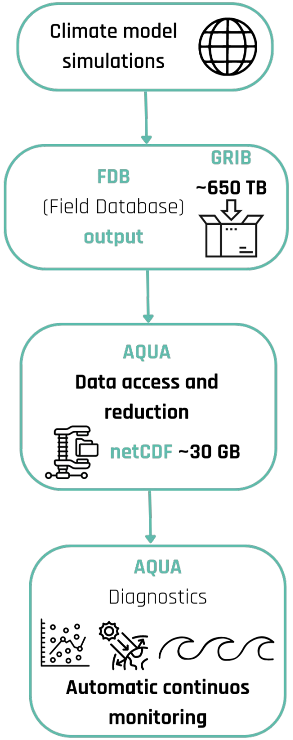
K. Keller's Talk on Destination Earth (*Wed, 17 Apr, 11:00 / Room 0.94/95*)

The challenges of AQUA

Many tools are available for model analysis (ESMValTool, E3SM, CMEC…), however:

- **Amount of data** for a single simulation is critically high
- **Data access** in not with simple netCDF files, but FDB software, with data both locally on HPC and remote on Data Bridge
- **Scalability** is crucial for even being able to open data
- Monitoring is continuous and automatic in a **operational framework**

AQUA experience inside DestinE is highly beneficial for many other projects


Climate model simulations
GRIB
FDB
~650 TB
(Field Database)
output
AQUA
Data access and reduction
netCDF ~30 GB
AQUA
Diagnostics
Automatic continuos monitoring

The Philosophy of AQUA

AQUA provides a framework to **access**, **process** and **analyze** large volumes of climate data (e.g. Destination Earth: 36 TB of output, per simulated year.)

Python3 library package based on xarray, dask, CDO and intake

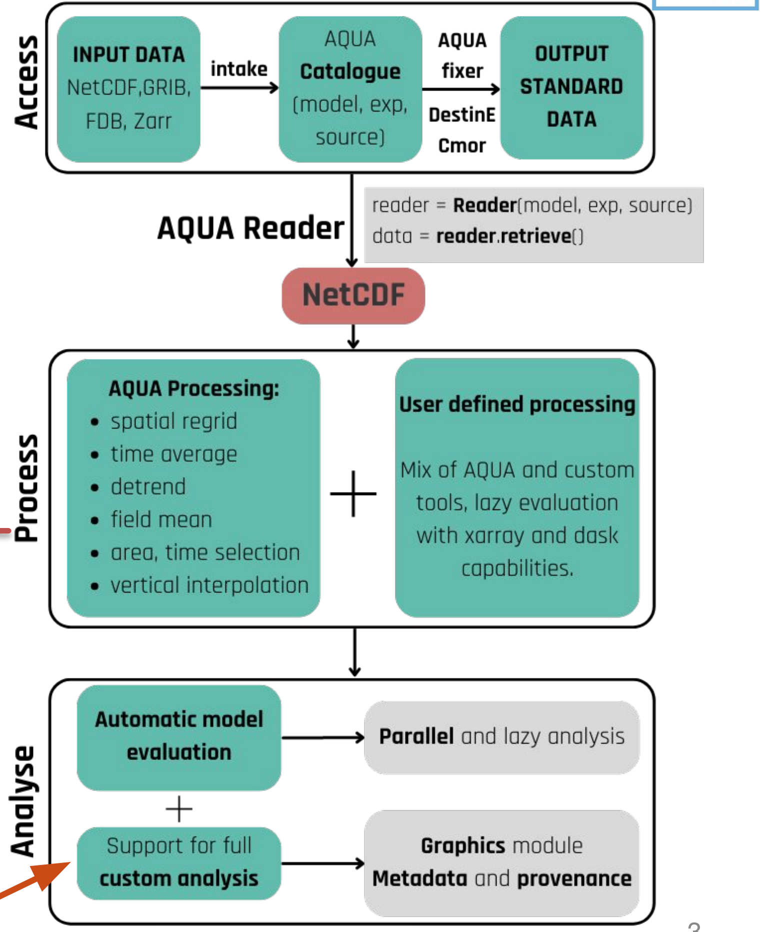
Modular structure seamless integration between core functions

Technical details are hidden simple user experience

Lazy access to all compatible data (parallel dask enabled)

Fast regrid capabilities based on precomputed weights

CI/CD on every new code addition with around 80% code coverage



```
reader = Reader(model, exp, source)
data = reader.retrieve()
```

S. Caprioli's following talk

Core concept: The Catalogue

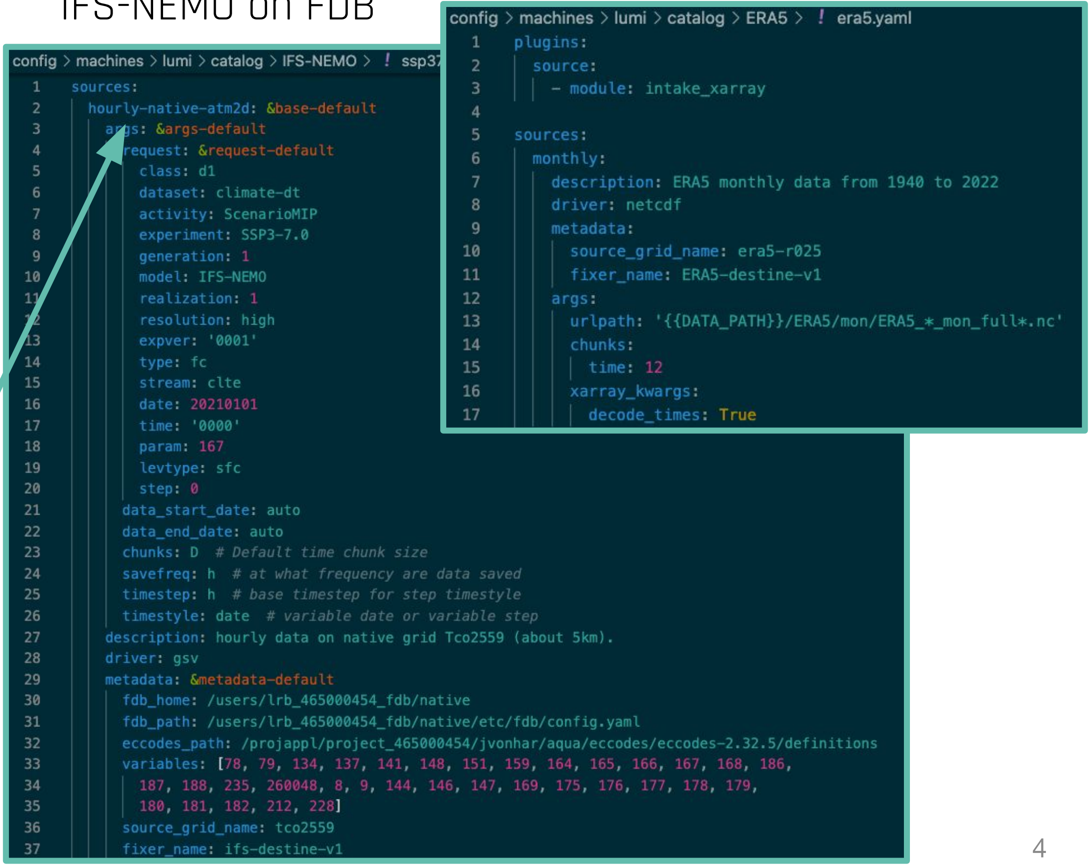
Catalogue: a machine dependent folder containing paths and technical details of all the sources that the user may want to access (models and observational datasets).

- Built on **intake** package
- **3 level hierarchy** for models, experiments, sources
- Ensembles support
- Dedicated intake driver for **lazy FDB access**
- **Templating** can be done for automatic catalogue generation!

Once a source is added it is available to every AQUA user, who will only ask for:

```
reader = Reader(model='IFS-NEMO', exp='ssp370',
                source='hourly-native-atm2d')
data = reader.retrieve()
```

This is a lazy (dask) xarray



IFS-NEMO on FDB

config > machines > lumi > catalog > IFS-NEMO > ! ssp37

```
sources:
  hourly-native-atm2d: &base-default
    args: &args-default
      request: &request-default
        class: d1
        dataset: climate-dt
        activity: ScenarioMIP
        experiment: SSP3-7.0
        generation: 1
        model: IFS-NEMO
        realization: 1
        resolution: high
        expver: '0001'
        type: fc
        stream: clte
        date: 20210101
        time: '0000'
        param: 167
        levtype: sfc
        step: 0
      data_start_date: auto
      data_end_date: auto
      chunks: D  # Default time chunk size
      savefreq: h  # at what frequency are data saved
      timestep: h  # base timestep for step timestyle
      timestyle: date  # variable date or variable step
    description: hourly data on native grid Tco2559 (about 5km).
    driver: gsv
    metadata: &metadata-default
      fdb_home: /users/lrb_465000454_fdb/native
      fdb_path: /users/lrb_465000454_fdb/native/etc/fdb/config.yaml
      eccodes_path: /projappl/project_465000454/jvonhar/aqua/eccodes/eccodes-2.32.5/definitions
      variables: [78, 79, 134, 137, 141, 148, 151, 159, 164, 165, 166, 167, 168, 186,
        187, 188, 235, 260048, 8, 9, 144, 146, 147, 169, 175, 176, 177, 178, 179,
        180, 181, 182, 212, 228]
      source_grid_name: tco2559
      fixer_name: ifs-destine-v1
```

ERA5 as netCDF

config > machines > lumi > catalog > ERA5 > ! era5.yaml

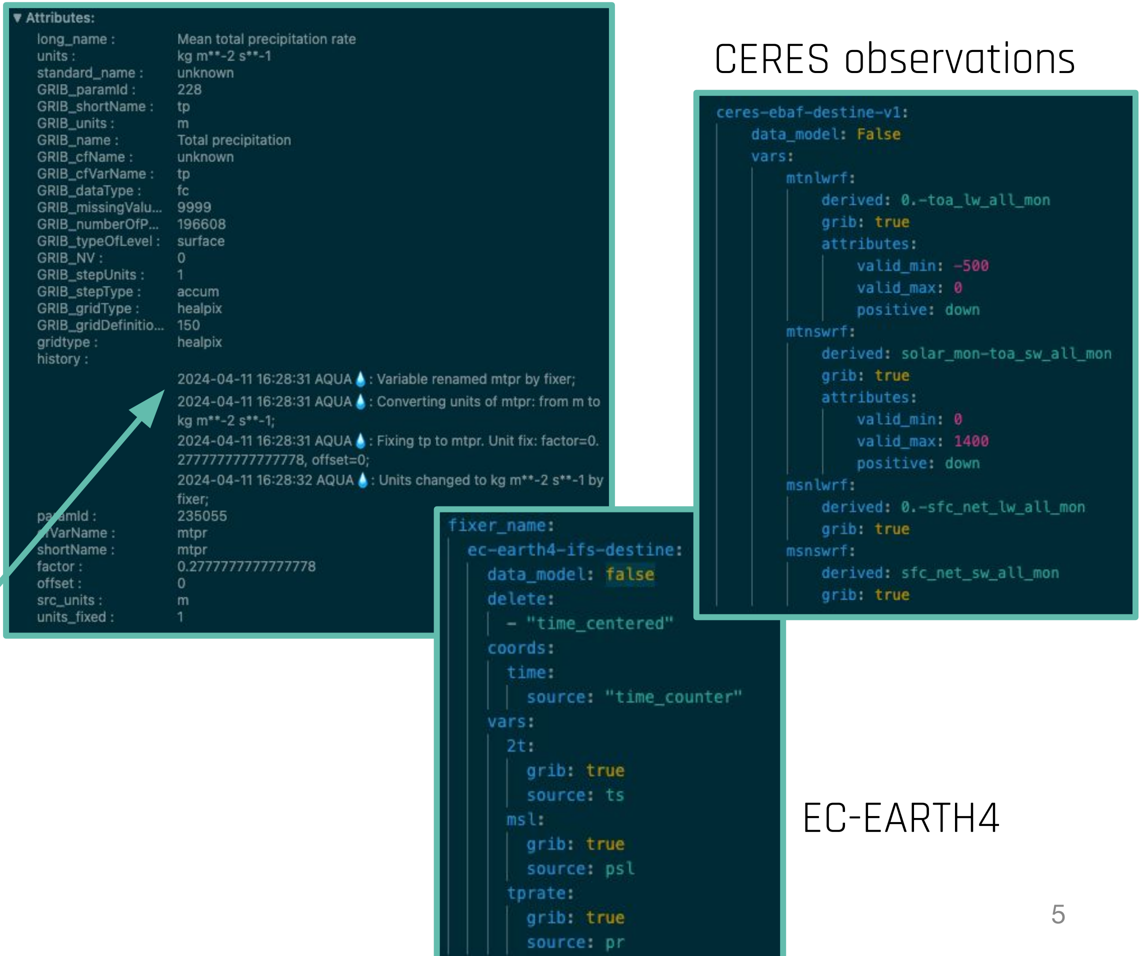
```
plugins:
  source:
    - module: intake_xarray

sources:
  monthly:
    description: ERA5 monthly data from 1940 to 2022
    driver: netcdf
    metadata:
      source_grid_name: era5-r025
      fixer_name: ERA5-destine-v1
    args:
      urlpath: '{{DATA_PATH}}/ERA5/mon/ERA5_*_mon_full*.nc'
      chunks:
        time: 12
      xarray_kwargs:
        decode_times: True
```

Core concept: The Fixer

AQUA intermodel and observations comparison requires a common metadata format.

- Yaml file with **fixer_name** that can be applied to multiple models/exps/sources
- In DestinE Climate DT a GRIB standard is used.
- Other standards can be defined for future applications (CMOR).
- Fixes are defined when creating a source and the **automatically enabled** when data are retrieved.
- For the user the process is totally **invisible**.
- GRIB and **history metadata** are updated for data provenance.



```
▼ Attributes:
    long_name :            Mean total precipitation rate
    units :                kg m**-2 s**-1
    standard_name :        unknown
    GRIB_paramId :         228
    GRIB_shortName :       tp
    GRIB_units :           m
    GRIB_name :            Total precipitation
    GRIB_cfName :          unknown
    GRIB_cfVarName :       tp
    GRIB_dataType :        fc
    GRIB_missingValu...    9999
    GRIB_numberOfP...      196608
    GRIB_typeOfLevel :     surface
    GRIB_NV :              0
    GRIB_stepUnits :       1
    GRIB_stepType :        accum
    GRIB_gridType :        healpix
    GRIB_gridDefinitio...  150
    gridtype :             healpix
    history :
                           2024-04-11 16:28:31 AQUA💧: Variable renamed mtpr by fixer;
                           2024-04-11 16:28:31 AQUA💧: Converting units of mtpr: from m to
                           kg m**-2 s**-1;
                           2024-04-11 16:28:31 AQUA💧: Fixing tp to mtpr. Unit fix: factor=0.
                           27777777777777778, offset=0;
                           2024-04-11 16:28:32 AQUA💧: Units changed to kg m**-2 s**-1 by
                           fixer;
    paramId :              235055
    cfVarName :            mtpr
    shortName :            mtpr
    factor :               0.2777777777777778
    offset :               0
    src_units :            m
    units_fixed :          1
```

CERES observations

```
ceres-ebaf-destine-v1:
    data_model: False
    vars:
        mtnlwrf:
            derived: 0.-toa_lw_all_mon
            grib: true
            attributes:
                valid_min: -500
                valid_max: 0
                positive: down
        mtnswrf:
            derived: solar_mon-toa_sw_all_mon
            grib: true
            attributes:
                valid_min: 0
                valid_max: 1400
                positive: down
        msnlwrf:
            derived: 0.-sfc_net_lw_all_mon
            grib: true
        msnswrf:
            derived: sfc_net_sw_all_mon
            grib: true
```

```
fixer_name:
  ec-earth4-ifs-destine:
    data_model: false
    delete:
      - "time_centered"
    coords:
      time:
        source: "time_counter"
    vars:
      2t:
        grib: true
        source: ts
      msl:
        grib: true
        source: psl
      tprate:
        grib: true
        source: pr
```

EC-EARTH4

Core concept: The Regridder

AQUA provides functions to interpolate and regrid data to match the spatial resolution of different datasets.

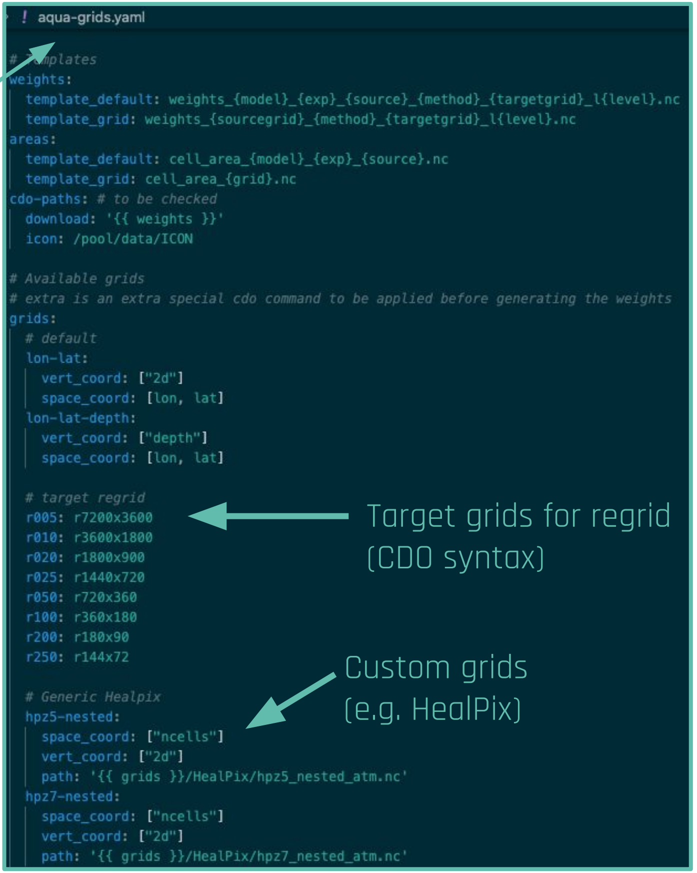
- Regridding functionalities based on smmregrid which operates **sparse matrix computation** based on **pre-computed weights**.
- Grids are defined in yaml file in AQUA and weights are evaluated with CDO only once.
- Grid files are already **synchronized** and have a backup on DKRZ Swift website.
- Each source can specify which grid is the correct one to use.
- Once the grid is set, regrid technicalities are transparent to user

```
reader = Reader(model='IFS-NEMO', exp='ssp370',
                          source='hourly-native-atm2d', regrid='r100')
data = reader.retrieve()
data_regridded = reader.regrid(data)
```

Prepare for regrid to 1° deg resolution

Data not regridded

Actual regrid



```
! aqua-grids.yaml

# Templates
weights:
  template_default: weights_{model}_{exp}_{source}_{method}_{targetgrid}_l{level}.nc
  template_grid: weights_{sourcegrid}_{method}_{targetgrid}_l{level}.nc
areas:
  template_default: cell_area_{model}_{exp}_{source}.nc
  template_grid: cell_area_{grid}.nc
cdo-paths: # to be checked
  download: '{{ weights }}'
  icon: /pool/data/ICON

# Available grids
# extra is an extra special cdo command to be applied before generating the weights
grids:
  # default
  lon-lat:
    vert_coord: ["2d"]
    space_coord: [lon, lat]
  lon-lat-depth:
    vert_coord: ["depth"]
    space_coord: [lon, lat]

  # target regrid
  r005: r7200x3600
  r010: r3600x1800
  r020: r1800x900
  r025: r1440x720
  r050: r720x360
  r100: r360x180
  r200: r180x90
  r250: r144x72

  # Generic Healpix
  hpz5-nested:
    space_coord: ["ncells"]
    vert_coord: ["2d"]
    path: '{{ grids }}/HealPix/hpz5_nested_atm.nc'
  hpz7-nested:
    space_coord: ["ncells"]
    vert_coord: ["2d"]
    path: '{{ grids }}/HealPix/hpz7_nested_atm.nc'
```

Target grids for regrid (CDO syntax)

Custom grids (e.g. HealPix)

Processing

AQUA core classes and functions can be integrated in any analysis

- Use **full AQUA diagnostics**
- Use AQUA framework as a **library** in your analysis
- Build **your own AQUA diagnostic**

Simple example of custom analysis

Custom processing for a seasonal cycle

```
from aqua import Reader

reader = Reader(model='IFS-FESOM', exp='ssp370-ng4', source='lra-r100-monthly')

data_20_30 = reader.retrieve(var='2t', startdate='2020', enddate='2030')

cycle_20_30 = data_20_30[‘2t’].aqua.fldmean().groupby(‘time.month’).mean(‘time’)
```

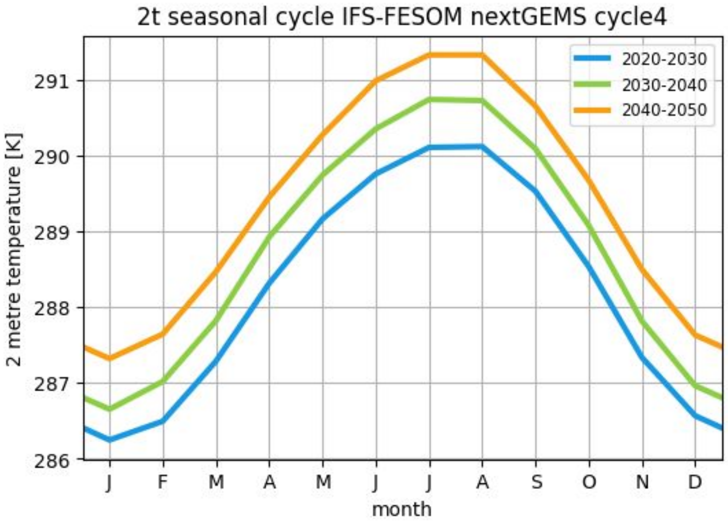
Graphics utilities for a custom plot

```
from aqua.graphics import plot_seasonalcycle

plot_seasonalcycle(data=[cycle_20_30['2t'], cycle_30_40['2t']], cycle_40_50[‘2t’]]

                   data_labels=['2020-2030', '2030-2040', '2040-2050'],

                   title='2t seasonal cycle IFS-FESOM nextGEMS cycle4')
```



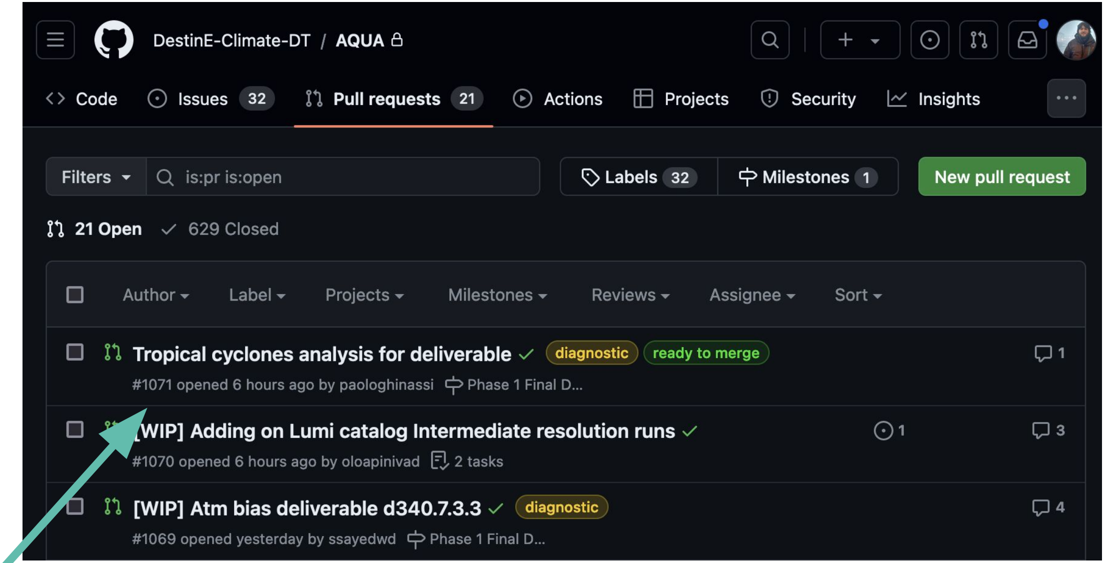
THANK YOU FOR YOUR ATTENTION!

The code is on GitHub and will be **Open Source** soon!

https://github.com/DestinE-Climate-DT/AQUA

Aqua core team Contacts:

- m.nurisso@isac.cnr.it
- silvia.caprioli@polito.it
- jost.hardenberg@polito.it
- p.davini@isac.cnr.it
- natalia.nazarova@polito.it



□ Tropical cyclones analysis for deliverable ✓ diagnostic ready to merge 1
#1071 opened 6 hours ago by paologhinassi Phase 1 Final D...

□ [WIP] Adding on Lumi catalog Intermediate resolution runs ✓ 1 3
#1070 opened 6 hours ago by oloapinivad 2 tasks

□ [WIP] Atm bias deliverable d340.7.3.3 ✓ diagnostic 4
#1069 opened yesterday by ssayedwd Phase 1 Final D...

Already over 1000 issues and pull requests with over 22 releases and automatic container creation!