

**Climate modeling**  
**EC-Earth and CMIP6 datasets**

## **Introduction**

During the course, we will work with various climate model data. In addition to climate model data analysis tasks, we will go learn details of EC-Earth model: source code, compilation and installation, running environment, data workflows etc. This document contains information how to approach climate model data with tools for processing, analysis, and visualization. There are also instructions how to download latest climate model results from online resources.

# Downloading CMIP6 data

CMIP data can be downloaded from Earth System Grid Federation nodes (ESGF). One starting point for CMIP6 data is

<https://esgf-node.llnl.gov/search/cmip6/>

Choose a subset of data from graphical user interface. For example, to find monthly average near-surface air temperature in EC-Earth historical simulation (1850–2014), you can

- 1) Choose *Institution ID* : **EC-Earth-Consortium** (Click “Search”)
- 2) Choose *Experiment ID* : **historical** (Click “Search”)
- 3) Choose *Variable* : **tas** (Click “Search”)
- 4) Choose Frequency : **mon** (Click “Search”)

Now the system shows all different model realizations for your data (e.g. 19 simulations in total), Figure ?. In this case, each row is an individual simulation experiment, identified by identifier **r1i1p1f1, r2i1p1f1, ...** There are several ways to download the data. You can simply click on “List Files” → “HTTP Download” to get a single datafile. Or, use the WGET script in command line. Some forms of download require an OpenID account, which is easy to create.

The screenshot shows the ESGF search interface. At the top right, it says "You are at the ESGF@DOE/LLNL node" and "Technical Support". The navigation bar includes "Home", "Contact Us", and "Data Nodes Status".

On the left, there are several filter panels:

- MIP Era**: +
- Activity**: +
- Model Cohort**: +
- Product**: +
- Source ID**: +
- Institution ID**: -
  - EC-Earth-Consortium (19)
- Source Type**: +
- Nominal Resolution**: +
- Experiment ID**: -
  - historical (19)
- Sub-Experiment**: +
- Variant Label**: +
- Grid Label**: +
- Table ID**: +
- Frequency**: -
  - mon (19)
- Realm**: +
- Variable**: -
  - tas (19)
- CF Standard Name**: +
- Data Node**: +

The main search area has a search bar with "Enter Text:" and buttons for "Search", "Reset", "Display 10", and "results per page". There are also checkboxes for "Show All Replicas", "Show All Versions", and "Search Local Node Only (Including All Replicas)".

Search Constraints:  EC-Earth-Consortium |  historical |  tas |  mon

Total Number of Results: 19  
-1- 2 Next >>  
Please login to add search results to your Data Cart  
Expert Users: you may display the search URL and return results as XML or return results as JSON

The results list shows 6 entries:

1. CMIP6.CMIP.EC-Earth-Consortium.EC-Earth3.historical.r20i1p1f1.Amon.tas.gr  
Data Node: esgf.bsc.es  
Version: 20191017  
Total Number of Files (for all variables): 165  
Full Dataset Services: [ Show Metadata ] [ List Files ] [ WGET Script ] [ LAS ] [ Show Citation ] [ PID ] [ Further Info ]
2. CMIP6.CMIP.EC-Earth-Consortium.EC-Earth3.historical.r19i1p1f1.Amon.tas.gr  
Data Node: esgf.bsc.es  
Version: 20191017  
Total Number of Files (for all variables): 165  
Full Dataset Services: [ Show Metadata ] [ List Files ] [ WGET Script ] [ LAS ] [ Show Citation ] [ PID ] [ Further Info ]
3. CMIP6.CMIP.EC-Earth-Consortium.EC-Earth3.historical.r24i1p1f1.Amon.tas.gr  
Data Node: esgf.bsc.es  
Version: 20190725  
Total Number of Files (for all variables): 165  
Full Dataset Services: [ Show Metadata ] [ List Files ] [ WGET Script ] [ LAS ] [ Show Citation ] [ PID ] [ Further Info ]
4. CMIP6.CMIP.EC-Earth-Consortium.EC-Earth3-Veg.historical.r4i1p1f1.Amon.tas.gr  
Data Node: esgf-cnr.hpc.cineca.it  
Version: 20190728  
Total Number of Files (for all variables): 165  
Full Dataset Services: [ Show Metadata ] [ List Files ] [ WGET Script ] [ Show Citation ] [ PID ] [ Further Info ]
5. CMIP6.CMIP.EC-Earth-Consortium.EC-Earth3.historical.r4i1p1f1.Amon.tas.gr  
Data Node: esgf-cnr.hpc.cineca.it  
Version: 20190721  
Total Number of Files (for all variables): 165  
Full Dataset Services: [ Show Metadata ] [ List Files ] [ WGET Script ] [ Show Citation ] [ PID ] [ Further Info ]
6. CMIP6.CMIP.EC-Earth-Consortium.EC-Earth3.historical.r16i1p1f1.Amon.tas.gr  
Data Node: esgf.bsc.es  
Version: 20191020  
Total Number of Files (for all variables): 165  
Full Dataset Services: [ Show Metadata ] [ List Files ] [ WGET Script ] [ LAS ] [ Show Citation ] [ PID ] [ Further Info ]

## Processing climate model data

Climate models produce data in various formats, mostly in GRIB and NetCDF. Here, we focus on [NetCDF format](#).

NetCDF data is (from <https://www.unidata.ucar.edu/software/netcdf>)

- **Self-Describing.** A netCDF file includes information about the data it contains.
- **Portable.** A netCDF file can be accessed by computers with different ways of storing integers, characters, and floating-point numbers.
- **Scalable.** Small subsets of large datasets in various formats may be accessed efficiently through netCDF interfaces, even from remote servers.
- **Appendable.** Data may be appended to a properly structured netCDF file without copying the dataset or redefining its structure.
- **Sharable.** One writer and multiple readers may simultaneously access the same netCDF file.
- **Archivable.** Access to all earlier forms of netCDF data will be supported by current and future versions of the software.

CMIP data downloaded from ESGF is in NetCDF format, and you can find a lot of metadata in the files.

There are several options to process, analyze and visualize NetCDF format data. Here, our main tool for processing is CDO ([Climate Data Operators](#)). CDO provides easy command-line interface for processing of model output. For example

Calculate annual average (assuming original data as monthly averages):

```
cdo yearmean input_data.nc annual_averages.nc
```

Extract a smaller region from data (LON: 20°E–40°E, LAT: 50°N–60°N):

```
cdo sellonlatbox,20,40,50,60 input_data.nc regional_box.nc
```

Full CDO manual: <https://code.mpimet.mpg.de/projects/cdo/embedded/index.html>

We also encourage to process datasets in **Python** (see details later in this document).

## Visualization of climate model data

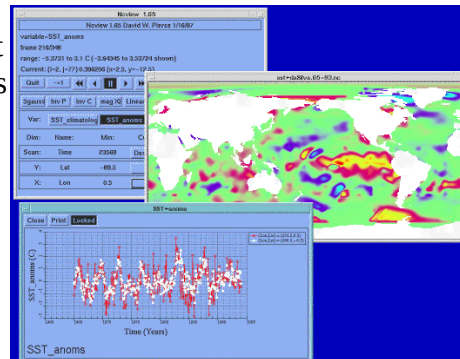
For visualization of NetCDF, several options [are available](#). We recommend to use

- 1) ncview for quick looking at the data
- 2) Panoply for polishing final graphics
- 3) Python (with Matplotlib + Cartopy)

[Ncview](#) is a visual browser, for quickly looking into data. It is possible to save plots, but making nice polished figures is difficult. It is very useful when you want to understand what is inside the files, and if your results look realistic, before polishing a figure with other software.

### Download:

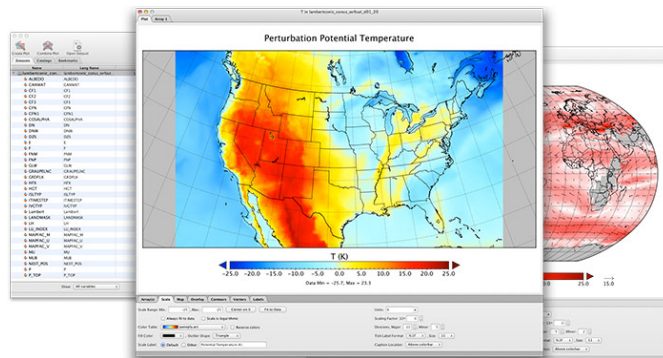
[http://meteora.ucsd.edu/~pierce/ncview\\_home\\_page.html](http://meteora.ucsd.edu/~pierce/ncview_home_page.html)



[Panoply](#) allows to create publication-quality figures from NetCDF data, as well as animations. There are a lot of options to tune your figure (colormaps, projections).

### Download:

<https://www.giss.nasa.gov/tools/panoply/>



## Python

Although most of analysis and visualization can be done as a combination of CDO + Ncview + Panoply, we recommend to use Python for creating analysis and visualization scripts. To get started, you need to install

- 1) Python 3 (<https://www.python.org/downloads>)
- 2) Matplotlib (<https://matplotlib.org>)
- 3) Cartopy (<https://scitools.org.uk/cartopy/docs/latest/>)
- 4) NetCDF4 (<https://unidata.github.io/netcdf4-python/netCDF4/index.html>)
- 5) Numpy (<https://numpy.org/>)

If you're using linux, installing the packages is easy through PIP ("pip install cartopy", "pip install matplotlib", "pip install netcdf4", "pip install numpy").

See next page for example on plotting climate data in Python.

**Example in Python:** plotting a datafile which has been downloaded from ESGF (EC-Earth, historical experiment, precipitation, year 1968).

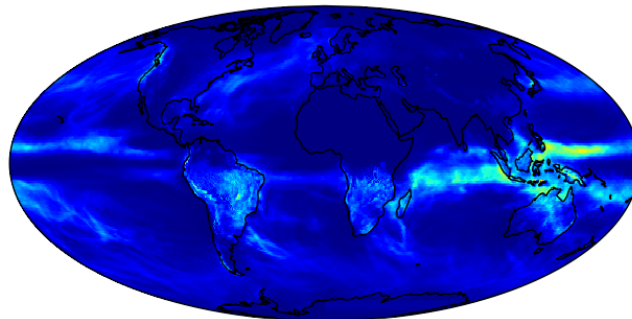
```
import netCDF4
import cartopy.crs as ccrs
import matplotlib.pyplot as plt
import numpy as np

# Load example CMIP6 data
file = netCDF4.Dataset("pr_Amon_EC-Earth3_historical_r1i1p1f1_gr_196801-196812.nc", 'r')

# Read variable data from file (only 1st timestep, i.e. January)
lon = file.variables['lon'][:]
lat = file.variables['lat'][:]
pr = file.variables['pr'][0][:, :]
file.close()

# Plot data
fig = plt.figure(figsize=(12,7))
m = plt.axes(projection=ccrs.Mollweide(central_longitude=0))
m.coastlines(linewidth=1)
cs = m.pcolormesh(lon, lat, pr, cmap='jet', transform=ccrs.PlateCarree(), linewidth=0)

# Show plot
plt.show()
```

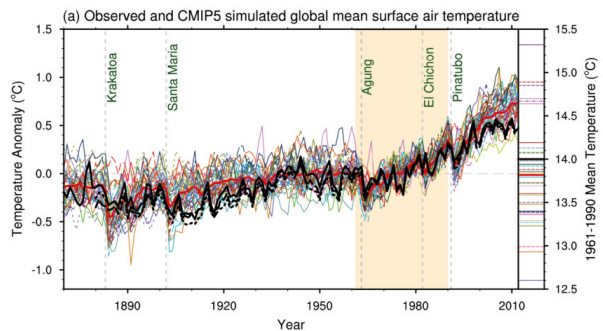


# Tasks

During the course we will analyse latest CMIP6 data from EC-Earth, complemented with data from various other climate models. We will reproduce certain analysis from IPCC AR5 report, and compare newest model results against earlier CMIP5 experiments. For EC-Earth, several configurations of the model are compared, e.g. EC-Earth (atmosphere-ocean) and EC-Earth-Veg (atmosphere-ocean-vegetation). You will learn how to set up, modify and run climate models, and deep knowledge on how to analyze climate model output.

Below are examples of tasks to be covered during the course:

Reproduce the IPCC AR5 figure 9.8 (right) for CMIP6 EC-Earth model: temperature anomalies during the historical simulation (1850–2014). How does EC-Earth compare to reanalyzed time-series (black) or older CMIP5 models?



Analyze future scenarios in CMIP6: how do new SSP scenarios compare to RCP-scenarios in CMIP5? Reproduce IPCC AR5 figure SPM.7 (right) for global average surface temperature change and Northern Hemisphere September sea ice extent.

