

# Climate4Impact (C4I) and icclim Tools to ease access to climate data



Christian Pagé  
CERFACS, Toulouse, France



Royal Netherlands  
Meteorological Institute  
*Ministry of Infrastructure  
and the Environment*

Abel Aoun  
CERFACS Toulouse, France

Alessandro Spinuso  
KNMI, De Bilt, Netherlands



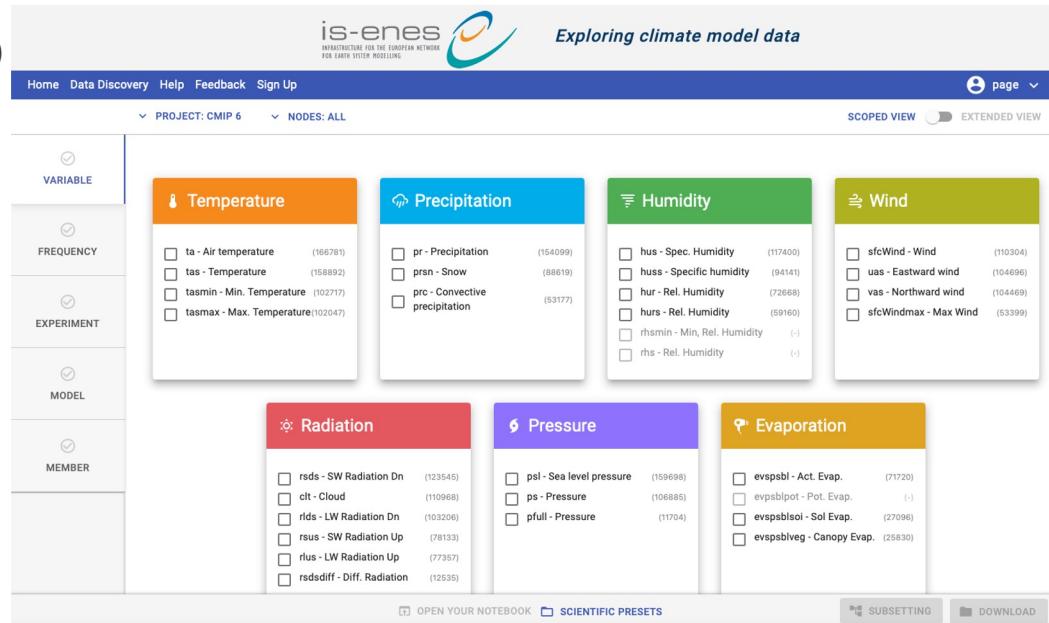
ESGF represents a **multinational** effort to securely **access, monitor, catalog, transport, and distribute** reference **data for climate** research experiments and observations.





## Platform for researchers to explore climate data and perform analysis

- Front-end to climate data infrastructure (ESGF)
- Tailored Search Interface with view modes
- Jupyter-Lab enhanced environment
- Notebooks gallery
- Flexible analysis features
  - Climate indices calculation: **icclim**
  - Data Staging/Reduction Workflows
  - Personal store for processing outcome
- Automated reproducibility mechanisms and documentation (Data/Analysis)
- Modular Deployment & Decoupled Architecture
- Pages for Models Performance Comparison (**ESMValTool**)



The screenshot shows the C4I portal's search results page. The left sidebar has filters for VARIABLE (selected), FREQUENCY, EXPERIMENT, MODEL, and MEMBER. The results are displayed in seven columns: Temperature (orange), Precipitation (blue), Humidity (green), Wind (yellow-green), Radiation (red), Pressure (purple), and Evaporation (gold). Each column lists variables like 'ta - Air temperature' or 'sfcWind - Wind' with their respective counts.

V2: Complete Redesign from V1  
Beta version available <https://dev.climate4impact.eu>





Copyright © MMXV Shep Hyken.

## Search Parametrisation made easier

**Exploring climate model data**

Home Data Discovery Help Feedback Sign Up

PROJECT: CMIP 6 NODES: ALL

VARIABLE

Frequency

EXPERIMENT

MODEL

MEMBER

variable:tasmax frequency:day experiment\_id:ssp585 source\_id:EC-Earth3 member\_id:r1l1p1f1 member\_id:r2l2p1f1

Temperature

- ta - Air temperature (9)
- tas - Temperature (6)
- tasmin - Min. Temperature (6)
- tasmax - Max. Temperature (6)

Precipitation

- pr - Precipitation (6)
- prsn - Snow (4)
- prc - Convective precipitation (1)

Radiation

- rsds - SW Radiation Dn (5)
- rlus - LW Radiation Up (4)
- rsus - SW Radiation Up (2)
- rlds - LW Radiation Dn (2)
- rsdssdiff - Diff. Radiation (-)
- clt - Cloud (2)

Pressure

- ps - Pressure (6)
- psi - Sea level (5)
- pfull - Pressure (2)

Temperature

- ta - Air temperature (9)
- tas - Temperature (6)
- tasmin - Min. Temperature (6)
- tasmax - Max. Temperature (6)

FREQUENCY

EXPERIMENT

MODEL

MEMBER

## Nodes Selection by Service

**Exploring climate model data**

Available ESGF Nodes

Select & enable Rook WPS subsetting

Node	Subsetting Mode
esgf1.dkrz.de	Rook WPS
esgf3.dkrz.de	Rook WPS
aims3.llnl.gov	Opendap
cmip.dess.tsinghua.edu.cn	Opendap
cmip.fio.org.cn	Opendap
cordexesg.dmi.dk	Opendap
crd-esgf-drc.ec.gc.ca	Opendap
data.meteo.unican.es	Opendap
dataserver.nccs.nasa.gov	Opendap
dpesgf03.nccs.nasa.gov	Opendap
espg-cccr.tropmet.res.in	Opendap

Wind

- uas - Eastward wind (6)
- vas - Northward wind (6)
- sfcWind - Wind (5)
- sfcWindmax - Max Wind (4)

Evaporation

- evpsbl - Act. Evap. (-)
- evsbjpot - Pot. Evap. (-)
- evspbsol - Sol Evap. (-)
- evpsbiveg - Canopy Evap. (-)

OK

## Climate4Impact Search for CMIP5/6 CORDEX Data

<https://dev.climate4impact.eu>

The screenshot shows a search interface for Climate4Impact. On the left, there are filters for PROJECT: CMIP 6 and NODES: ALL. The main area displays various climate variables grouped into categories: Temperature, Precipitation, Humidity, Wind, Radiation, Pressure, and Evaporation. Each category has a list of specific variables with their counts in parentheses. A large blue arrow points from this interface to the Jupyter notebook workspace.

Category	Variables	Count
Temperature	tasmin - Min. Temperature, tasmax - Max. Temperature, ta - Air temperature, tas - Temperature	(6), (6), (6), (4)
Precipitation	pr - Precipitation, prsn - Snow, prc - Convective precipitation	(6), (4), (1)
Humidity	hurs - Rel. Humidity, huss - Specific humidity, rhmin - Min. Rel. Humidity, rhmax - Max. Rel. Humidity, hrs - Rel. Humidity, has - Spec. Humidity, hur - Rel. Humidity	(6), (5), (1), (1), (5), (1)
Wind	sfcWind - Wind, sfcWind - Eastward wind, sfcWind - Northward wind, sfcWindmax - Max Wind	(6), (6), (6), (4)
Radiation	rds - SW Radiation Dn, rrsu - SW Radiation Up, rlds - LW Radiation Dn, rrls - LW Radiation Up, rsdiff - Diff. Radiation, cld - Cloud	(5), (1), (2), (2), (1), (2)
Pressure	ps - Pressure, cstl - Sea level pressure, oful - Pressure	(1), (6), (1)
Evaporation	evipob1 - Act. Evap., evipob2 - Pot. Evap., evipob3 - Sol Evap., evipobveg - Canopy Evap.	(1), (1), (1), (1)

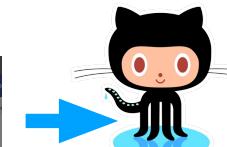
## Workflows for data staging & reduction onto icclim powered Notebooks

The screenshot shows a Jupyter notebook environment. A blue arrow points from the search interface to this workspace. The notebook contains Python code for calculating time averages and plotting summer days. It includes two heatmaps showing spatial patterns over time and latitude.

```

# Set spatial extent and centre
central_lat = 47.8
central_lon = 10.0
extent = [-138, 38, 38, 36] # Western Europe
# Calculate time average
icl_avg = iccl.calculate('Time', keep_attrs=True)
# Set plot projection

```

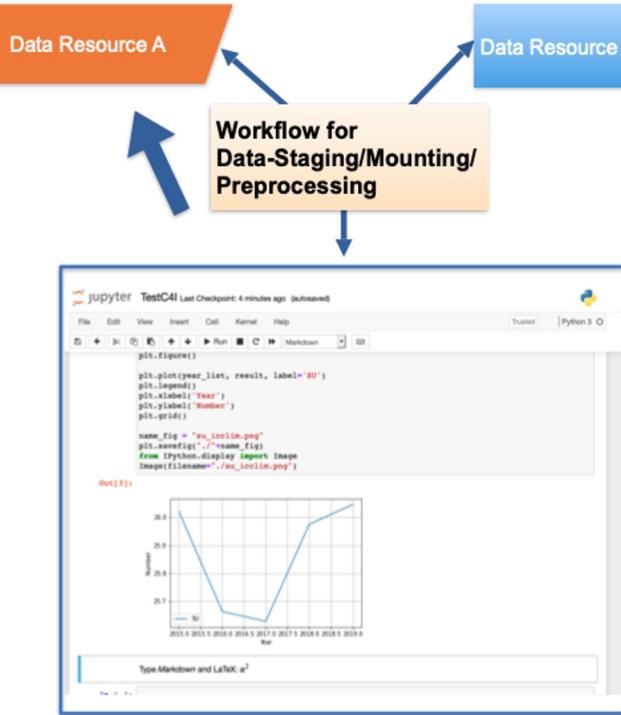


Save/Share Progress to Git

Reduced Data MyBinder Reproduce

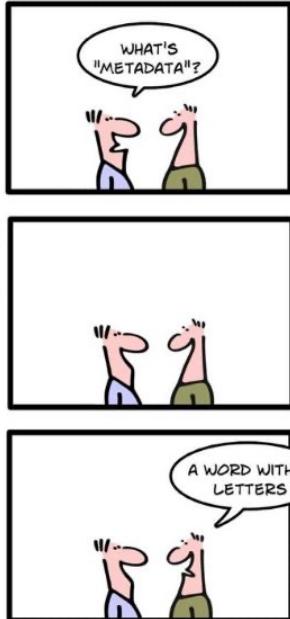
- Trace Changes to Restore, Recover Software and/or Data





- access distributed raw data
- develop, document and reuse methods for processing and visualisation.
- update/extend raw data and software
- Track changes and rollback  
(Traceability/Recovery)
- keep old versions of the data after updates  
(Reproducibility)
- snapshot and restore the state of a workspace software  
(Reproducibility)

SIMPLY EXPLAINED:  
METADATA



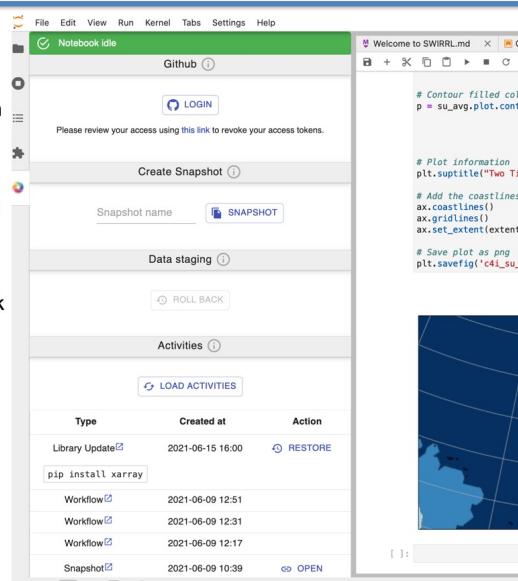
**Workflow Monitoring**

**GitHub Authentication**

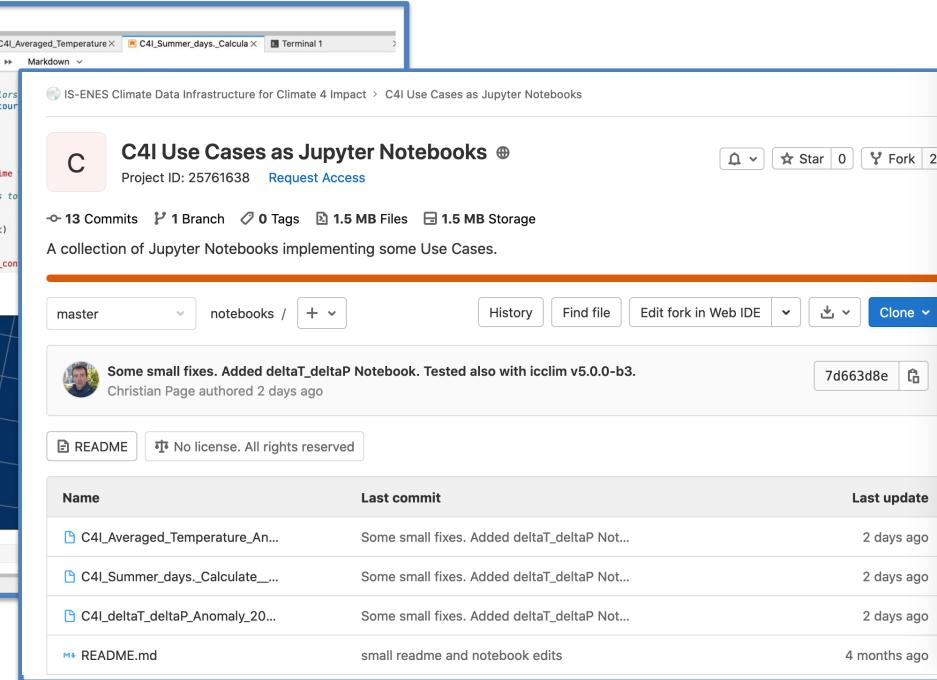
**Snapshot Controls**

**Data Staging Rollback**

**Activities History and Provenance**



A screenshot of the IS-ENES Climate Data Infrastructure for Climate 4 Impact interface. It shows a sidebar with sections for Workflow Monitoring, GitHub Authentication, Snapshot Controls, Data Staging Rollback, and Activities History and Provenance. A red curly brace on the left groups the Activities History and Provenance section with the others. To the right, there are two main panes: one showing a GitHub login dialog and another showing a Jupyter Notebook interface with code and a plot.



A screenshot of a GitHub repository page titled "C4I Use Cases as Jupyter Notebooks". The repository has a project ID of 25761638 and 13 commits. The description states: "A collection of Jupyter Notebooks implementing some Use Cases." The page includes a file browser, commit history, and a table of contents for notebooks.

Name	Last commit	Last update
C4I_Averaged_Temperature_An...	Some small fixes. Added deltaT_deltaP Not...	2 days ago
C4I_Summer_days_Calculate_...	Some small fixes. Added deltaT_deltaP Not...	2 days ago
C4I_deltaT_deltaP_Anomaly_20...	Some small fixes. Added deltaT_deltaP Not...	2 days ago
README.md	small readme and notebook edits	4 months ago

<https://gitlab.com/is-enes-cdi-c4i/notebooks>



**SWIRRL hides the complexity of orchestrating Workspaces in a target Cloud resource based on Kubernetes Cluster**

**Integrates Interactive Tools  
(Notebooks, Workflows, GIT, Binder)**



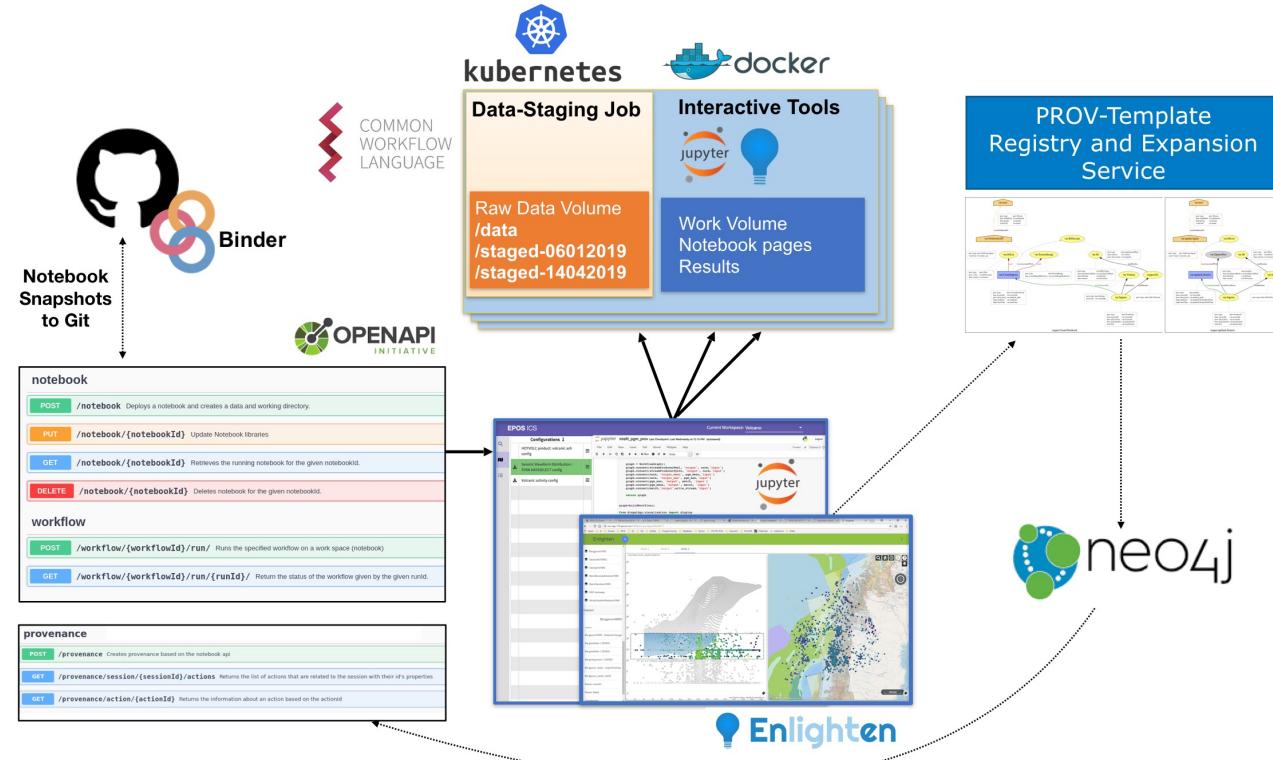
**Offers a REST Web API**

**Manages Metadata (Provenance)**

Findable  Accessible  Interoperable  Reusable 



**Supports Reproducibility**



Home Data Discovery Help Feedback Sign Up

PROJECT: CMIP 6 NODES: CUSTOM (2)

VARIABLE

FREQUENCY

EXPERIMENT

MODEL

MEMBER

variable:ta variable:prsn variable:huss frequency:day experiment\_id:ssp585

Model

- CanESM5 - CanESM5
- MPI-ESM1-2-LR - MPI-ESM1.2-LR
- UKESM1-0-LL - UKESM1.0-N96ORCA1
- CNRM-CM6-1 - CNRM-CM6-1
- MIROC6 - MIROC6
- CNRM-ESM2-1 - CNRM-ESM2-1
- MRI-ESM2-0 - MRI-ESM2.0
- GFDL-CM4 - GFDL-CM4
- HadGEM3-GC31-LL - HadGEM3-GC3.1-N96ORCA1
- MPI-ESM1-2-HR - MPI-ESM1.2-HR
- INM-CM4-8 - INM-CM4-8
- INM-CM5-0 - INM-CM5-0
- ACCESS-CM2 - Australian Community Climate and Earth System Model
- AWI-CM-1-1-MR - AWI-CM 1.1 MR
- NorESM2-LM - NorESM2-LM (low atmosphere-medium ocean resolution)
- BCC-CSM2-MR - BCC-CSM 2 MR
- CMCC-CM2-SR5 - CMCC-CM2-SR5
- FGOALS-g3 - FGOALS-g3

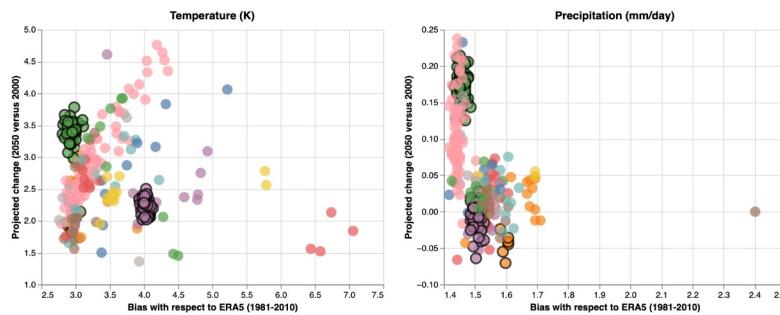
COMPARE MODEL PERFORMANCE

## Climate impact result viewer

This application shows results from CMIP5 and CMIP6 models, calculated with ESMValTool. It is intended to provide some guidance for climate impact researchers, to select one or more datasets that adequately sample the spread of the CMIP ensemble.

- Bias is calculated with respect to the ERA5 reanalysis dataset over the period 1981–2015.
- Future change is calculated for 2036–2065 as compared to 1986–2015.
- Area is set to Europe (lon 0–39; lat 30–76.25)
- All data are taken from the RCP/SSP 8.5 scenario

Hold ctrl to pan and zoom, hold alt to select a range (points will be highlighted in both graphs), then hold shift to select multiple points.



Project\_project CMIP6

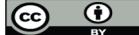
Selected datasets:

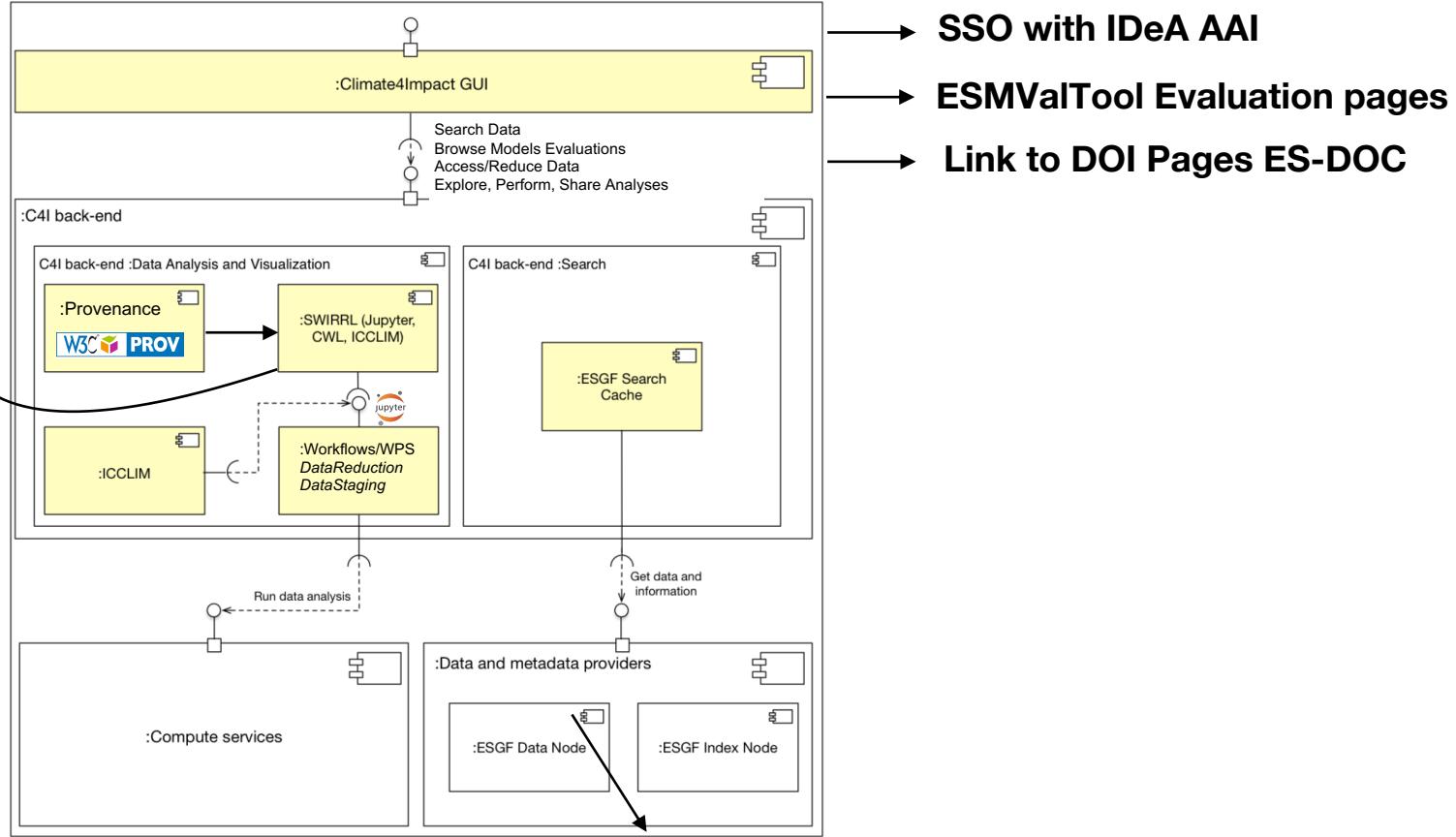
- CNRM-CM6-1
- MIROC6
- CanESM5

[View static recipe output](#)

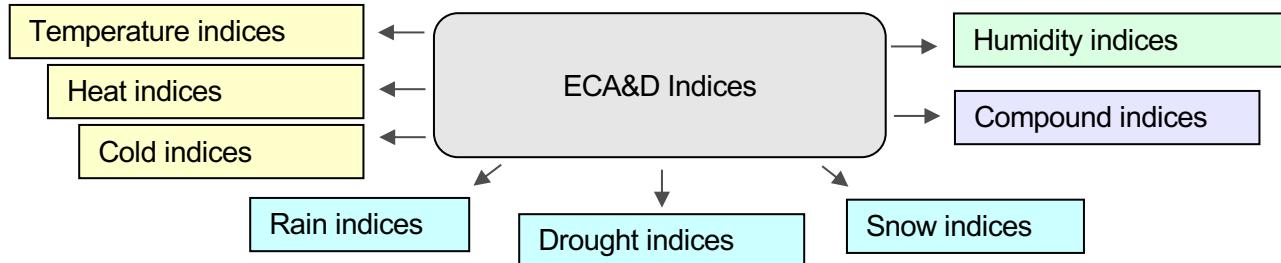
Netherlands eScience Center

IS-ENES3 has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 824084





## icclim: Index Calculation for Climate



- Intra-period extreme temperature range [° C] - **ETR**
- Warm days (days with mean temperature > 90th percentile of daily mean temperature) - **TG90p**
- Summer days (days with max temperature > 25 ° C) - **SU**
- ...

- Python code developed at Cefracs since September 2013
- Funded by EU FP7 IS-ENES2, FP7 CLIPC and H2020 IS-ENES3
- Generic and modular approach, can be reused in other environments
- New V5 completely rewritten and using underlying xclim functions, based on xarray and dask
- I/O interface is structured for optimal performance
- Implement the proper percentile indices calculations when calculation period overlaps reference period (called bootstrapping method)
- Available indices: [https://icclim.readthedocs.io/en/latest/explanation/climate\\_indices.html#icclim-capabilities](https://icclim.readthedocs.io/en/latest/explanation/climate_indices.html#icclim-capabilities)



Documentation: [https://icclim.readthedocs.io/en/latest/python\\_api.html](https://icclim.readthedocs.io/en/latest/python_api.html)

Source code: <https://github.com/cerfacs-globc/icclim>

Current Version 5.4.0: <https://github.com/cerfacs-globc/icclim/releases/tag/v5.4.0>

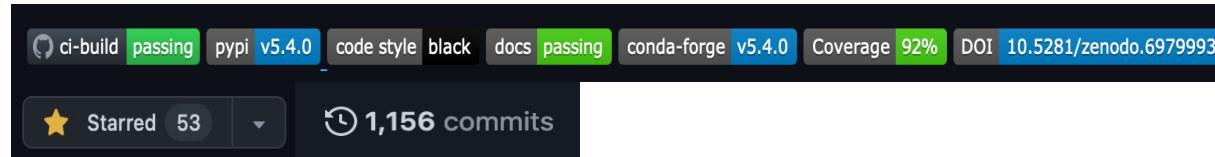
## icclim.index(\*\*kwargs)

- Parameters:**
- **in\_files** (*str / list[str] / Dataset / DataArray*) – Absolute path(s) to NetCDF dataset(s), including OPeNDAP URLs, or path to zarr store, or *xarray.Dataset* or *xarray.DataArray*.
  - **index\_name** (*str*) – Climate index name. For ECA&D index, case insensitive name used to lookup the index. For user index, it's the name of the output variable.
  - **var\_name** (*str / list[str] / None*) – *optional* Target variable name to process corresponding to **in\_files**. If *None* (default) on ECA&D index, the variable is guessed based on the climate index wanted. Mandatory for a user index.
  - **slice\_mode** (*str*) – Type of temporal aggregation: {"year", "month", "DJF", "MAM", "JJA", "SON", "ONDJFM" or "AMJJAS"}. Default is "year". See **slice\_mode** for details.
  - **time\_range** (*list[datetime.datetime] / None*) – *optional* Temporal range: upper and lower bounds for temporal subsetting. If *None*, whole period of input files will be processed. Default is *None*.
  - **out\_file** (*str / None*) – Output NetCDF file name (default: "icclim\_out.nc" in the current directory). Default is "icclim\_out.nc". If the input **in\_files** is a **Dataset**, **out\_file** field is ignored. Use the function returned value instead to retrieve the computed value. If **out\_file** already exists, icclim will overwrite it!
  - **threshold** (*float / list[float] / None*) – *optional* User defined threshold for certain indices. Default depend on the index, see their individual definition. When a list of threshold is provided, the index will be computed for each thresholds.



## icclim v5.4.0

<https://github.com/cerfacs-globc/icclim>



ci-build passing pypi v5.4.0 code style black docs passing conda-forge v5.4.0 Coverage 92% DOI 10.5281/zenodo.6979993

Starred 53 1,156 commits

```
import icclim
summer_days = icclim.su("netcdf_files/tasmax_1990-2100.nc", out_file="summer_days.nc")
```

# A F.A.I.R icclim



Findable:

- [Open source code](#), licensed under the permissive Apache 2 license.
- Try to follow [OpenSSF principles](#)
- Documentation freely available on [readthedocs](#)
- Dissemination by Christian in conferences (AGU, AMS, EGU...).
- Integrated in [C4I](#) and C3S.



Accessible

- Documentation on readthedocs.
- Github issues and pull requests.
- CERFACS support the development.



Interoperable

- Based on well knowns libraries: xarray, xclim, numpy.
- Support NetCDF and xarray.Dataset input formats.
- Indices are derivated from ECA&D and CF conventions standards.
- icclim custom parsing for non CF complying inputs.



Reusable

- NetCDF output, with enriched metadata.
- Easy to integrate within existing infrastructures.
- Easy to write small scripts with it.



## icclim: examples

notebooks: <https://gitlab.com/is-enes-cdi-c4i/notebooks/>

```
## KNMI TX
tx_files = glob.glob(f"netcdf_files/knmi/clean/*tx*.nc")
bp = [datetime.datetime(1901, 1, 1), datetime.datetime(1921, 12, 31)]
icclim.index(index_name='tx90p',
              in_files=tx_files,
              base_period_time_range=bp,
              slice_mode='YS',
              out_file="netcdf_files/output/out.nc")
```

## [new in 5.1.0] Rechunk how data is stored

Documentation: [https://icclim.readthedocs.io/en/latest/how\\_to/dask.html#create-an-optimized-chunking-on-disk](https://icclim.readthedocs.io/en/latest/how_to/dask.html#create-an-optimized-chunking-on-disk)

```
import icclim

ref_period = [datetime.datetime(1980, 1, 1), datetime.datetime(2009, 12, 31)]
with icclim.create_optimized_zarr_store(
    in_files="netcdf_files/tas.nc",
    var_names="tas",
    target_zarr_store_name="opti.zarr",
    keep_target_store=False,
    chunking={"time": -1, "lat": "auto", "lon": "auto"},
) as opti_tas:
    icclim.index(
        index_name="TG90p",
        in_files=opti_tas,
        slice_mode="YS",
        base_period_time_range=ref_period,
        out_file="netcdf_files/output/tg90p.nc",
    )
```

## [new in 5.1.0] Compute multiple indices at once

Documentation [https://icclim.readthedocs.io/en/latest/how\\_to/recipes\\_ecad.html#multi-index-computation](https://icclim.readthedocs.io/en/latest/how_to/recipes_ecad.html#multi-index-computation)

```
bp = [datetime.datetime(1991, 1, 1), datetime.datetime(1999, 12, 31)]
tr = [datetime.datetime(1991, 1, 1), datetime.datetime(2010, 12, 31)]
res = icclim.indices(
    index_group=IndexGroup.HEAT,
    in_files=".//netcdf_files/climpact.sampledata.gridded.1991-2010.nc",
    base_period_time_range=bp,
    time_range=tr,
    out_file="pouetpouet.nc",
)
```

## [new in 5.2.0] one function per index

Documentation: [https://icclim.readthedocs.io/en/latest/references/ecad\\_functions\\_api.html](https://icclim.readthedocs.io/en/latest/references/ecad_functions_api.html)

```
import glob
import icclim
summer_days = icclim.su(
    in_files=glob.glob("netcdf_files/tasmax*.nc"),
    out_file="summer_days.nc",
)
```

## [new in 5.3.0] custom season between exact dates



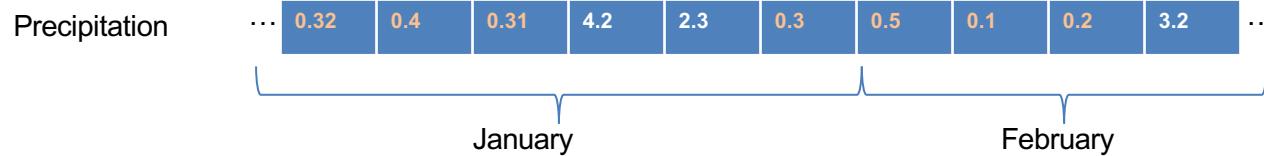
```
result = icclim.su(  
    in_files=xr.open_dataset(climp_file).tmax,  
    slice_mode=["season", ["19-07", "8 Aout"]],  
    ).compute()
```

# icclim v6

## Generic climate indices made easy

 CERFACS

## [new in 6.0] Spells starting before the season bounds are properly counted



Say we want to compute cdd on a monthly basis...

CDD: number of consecutive where  $\text{pr} < 1 \text{ mm/day}$

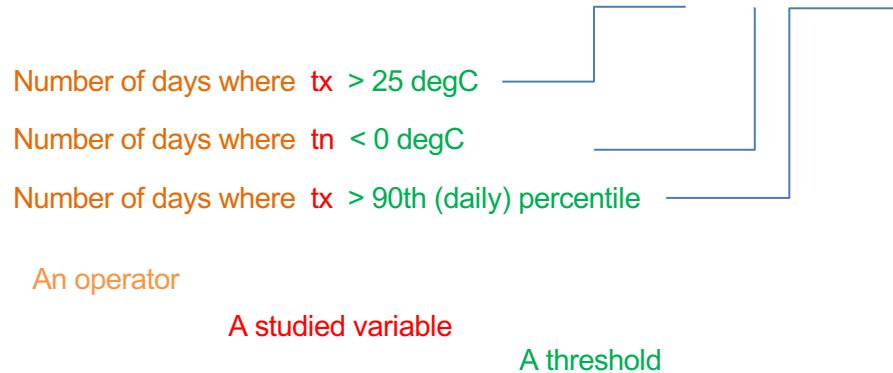
```
cdd = icclim.cdd(in_files=precipitation, slice_mode="month").CDD.compute()
```

Should February cdd be 3 days or 4 ?

With icclim 6 we count spells length before resampling into monthly values,  
So CDD in February would be 4 here

## [new in 6.0] Generic indices – What ?

What's the difference between SU, FD and Tx90p ?



For our output, we want the right computation to be executed and to have specific metadata to clearly identify :

- What was the computation configuration
- What was the initial data used

## [new in 6.0] Generic indices – How ?

```
icclim.index(in_files=climp_file,  
             Operator index_name="count_occurrences",  
             Studied variable var_name="tmax",  
             Threshold threshold="≥ 20 deg_C",  
             ).compute()
```

Not CF proof!

standard\_name: number\_of\_days\_with\_maximum\_air\_temperature\_above\_threshold

long\_name: Number of days when maximum air temperature is greater or equal  
to 293.15 K for each year.

units: d

cell\_methods: time: sum over days

```

thresh = Threshold("≥ 75 period_per",
                    threshold_min_value="1 mm/day",
                    reference_period=['1991-01-01', "31 décembre 2000"])

r75p = icclim.index(climp_file,
                     pr_thresh = Threshold("≥ 75 doy_per", threshold_min_value="1 mm/day")
                     temp_thresh = Threshold("≥ 75 doy_per"))                                CW (ECAD)

c_w = icclim.index(climp_file,
                     pr_thresh = Threshold("≥ 75 doy_per", threshold_min_value="1 mm/day")
                     temp_thresh = Threshold("≥ 75 doy_per")                                     Cold, wet and windy
                     wind_thresh = Threshold("≥ 95 doy_per")                                    (Bretagne?)

c_w_w = icclim.index(ds,
                      index_name="count_occurrences",
                      var_name=["precip", "tmax", "sfcWind"],
                      threshold=[pr_thresh, temp_thresh, wind_thresh])
  
```

## [new in 6.0] Generic indices – Ah...

Cold, wet and windy [long\\_name](#):

Number of days when precipitation is greater or equal to 75.0th day of year percentile (Day of year percentiles were computed per grid cell, on the ['1991-01-01', '2010-12-31'] period, with a 5 days centred window to aggregate values around each day of year. Only values 1.0 kg m-2 d-1 were considered). And maximum air temperature is greater or equal to 75.0th day of year percentile (Day of year percentiles were computed per grid cell, on the ['1991-01-01', '2010-12-31'] period, with a 5 days centred window to aggregate values around each day of year). And Near-Surface Wind Speed is greater or equal to 95.0th day of year percentile (Day of year percentiles were computed per grid cell, on the ['1991-01-01', '2010-12-31'] period, with a 5 days centred window to aggregate values around each day of year). for each year.

## [new in 6.0] Generic indices – Operators

count_occurrences	SU, TR, TG90p, CD, R95p SD50cm ...
max_consecutive_occurrence	CSU, CFD, CDD, CWD
sum_of_spell_lengths	WSDI, CSDI
excess	GD4
deficit	HD17
fraction_of_total	R75pTOT, R95pTOT, r99pTOT
maximum	RX1day, TXx, TNx, custom_maximum
minimum	TXn, TNn, user_index - minimum
average	TG, TN, TX, SDII, SD, custom_mean
sum	PRCPTOT, custom_sum
standard_deviation	**new**
max_of_rolling_sum	RX5day, user_indexrolling_sum
min_of_rolling_sum	custom_rolling_sum
min_of_rolling_average	custom_rolling_mean
max_of_rolling_average	custom_rolling_mean
mean_of_difference	DTR
difference_of_extremes	ETR
mean_of_absolute_one_time_step_difference	vDTR
difference_of_means	custom_anomaly

## [new in 6.0] Generic indices – Thresholds

Scalar	<code>Threshold(query="&gt;=", value=25, unit="degC")</code>
Day of year percentiles	<code>Threshold(query="&gt;=", value=99, unit="doy_per")</code>
Period percentiles	<code>Threshold(query="&gt;=", value=99, unit="period_per")</code>
Bounded period percentiles.	<code>Threshold("&gt;= 75 period_per", threshold_min_value="1 mm/day", reference_period=[1991-01-01, "31 décembre 2000"])</code>
Per-grid cell threshold	<code>Threshold(query="&gt;=", value="data.nc", threshold_var_name="tmin", unit="K")</code>
Sequence of scalars	<code>Threshold(query="&gt;=", value=[25, 29], unit="degC")</code>



# Thanks !

On behalf of the climate4impact and icclim teams

<https://dev.climate4impact.eu>

<https://icclim.readthedocs.io/en/stable/>

Christian Pagé

Alessandro Spinuso

Abel Aoun

christian.page@cerfacs.fr

alessandro.spinuso@knmi.nl

abel.aoun@cerfacs.fr



*This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°824084*



Our website

<https://is.enes.org/>



Follow us on Twitter !  
[@ISENES\\_RI](https://twitter.com/ISENES_RI)



Contact us at

[is-enes@ipsl.fr](mailto:is-enes@ipsl.fr)



Follow our channel  
**IS-ENES3 H2020**

