

Data and Analyse

Table of Content

Data and Analyse	1
1. Data	2
1.1. CMIP archive	2
1.1.1. Start with CMIP archives	2
1.1.2. Access CMIP archives	2
1.1.3. Dive into CMIP archives at IPSL	2
1.1.4. What's a dataset?	2
1.2. THREDDS access (aka DODS service)	2
1.3. Observations / Reanalyses	2
1.4. Data management Tools	3
2. Analyse	3
2.1. CliMAF: a Climate Model Assessment Framework	3
2.2. The C-ESM-EP: the CliMAF Earth System Model Evaluation Platform	3
2.2.1. Use C-ESM-EP in libGCM	3
2.3. Using the fast graphical display remote system at TGCC	4
2.3.1. From a TGCC partner network	4
2.3.2. From any network, using ssh1, Spirit or any gateway known by TGCC	4
2.3.3. GNOME configuration	4
2.3.4. Finally	4
2.4. ESMValTool	5

1. Data

1.1. CMIP archive

1.1.1. Start with CMIP archives

We highly recommend to read the user's guide of CMIP projects available at: <https://pcmdi.llnl.gov/mips/>

1.1.2. Access CMIP archives

CMIP archives are available through the ESPRI platform including CICLAD and ClimServ machines. You can create an account on ESPRI mesocentre to transparently log to CICLAD or ClimServ at <https://meso-account.ipsl.fr/>.

1.1.3. Dive into CMIP archives at IPSL

The IPSL-CM model produces CMIP-compliant files. The CMIP convention include the CF convention with additional rules from the WIP panel (ex. a CMIP file must describe only one variable). Once the IPSL-CM data are quality checked, the CMIP-compliant files are migrated to a dedicated filesystem at TGCC following the CMIP directory structure. This part of the TGCC filesystem is read-only mounted on the ESPRI mesocentre. On CICLAD and ClimServ, the CMIP data archives are then available under :

```
/bdd/CMIP6
```

```
/bdd/CMIP5
```

The CMIP data from the other GCMs are downloaded at the IDRIS and IPSL. They are organized in the same CMIP directory structure and transparently available in the same `/bdd` root.

Example: `/bdd/CMIP5/output/IPSL` provides access to the **whole** IPSL-CM data production for CMIP5 exercice. `/bdd/CMIP5/output/BCC` provides access to BCC data requested by the IPSL community.

We adopted an "on-demand" process to add CMIP data from other GCM. Please send your request at `glipsl@ipsl.fr` with the following template:

```
cat < EOF > my_template.txt
project=CMIP5
experiments=historical amip
models=IPSL-CM5A-LR CNRM-CM5
ensembles=all
variables[atmos][3hr]=cltc tas
variables[land][fx]=sftgif
variables[seaIce][mon]=sic evap
EOF
```

1.1.4. What's a dataset?

A "dataset" (as defined by ESGF) is **one** version of a data set resulting from a single simulation (i.e., characterized by a unique option of each CMIP facet before the version such as the institute, the model, the domain, the experiment, the frequency, the ensemble, etc.).

Examples: CMIP5 dataset: `CMIP5/output1/IPSL/IPSL-CM5A-LR/ 1pctCO2/mon/atmos/Amon/r1i1p1/v20110427`

A dataset is the finest granularity for ESGF publication.

1.2. THREDDS access (aka DODS service)

The IPSL ESGF node includes a THREDDS data server which replaces the deprecated DODS server. This service automatically provides free access to parts of filesystems or remotely-mounted filesystems. From TGCC, the shared space `$CCCWORKDIR/./././thredds/$LOGIN` is expose at <https://vesg.ipsl.upmc.fr/thredds/catalog/work/catalog.html>.

Be careful that the THREDDS spaces at TGCC should only includes hard-links of data from your project spaces on WORK. The THREDDS folders were not designed to host deep copies of your files because of high constraints on storage and inodes.

1.3. Observations / Reanalyses

Many observational datasets and reanalyses are available thanks to Climserv on /bdd (ERAInterim, ERA5, EOBS, and others).

Have a look at it before downloading data to both save your time and make multiple copies of the same dataset on the disks!

1.4. Data management Tools

Informations are available [here](#)

2. Analyse

Many toolboxes/software are available on IPSL servers to do analyses (like CDO, nco, Ferret, Python, R). Additionally, the CliMAF python library and the C-ESM-EP evaluation package are in-house developments available for you to ease your analyses.

2.1. CliMAF: a Climate Model Assessment Framework

CliMAF (<https://climaf.readthedocs.io/en/master/>) is a python library to help you:

- browse and find data in your archives (like CMIP, CORDEX, IPSL model outputs, or observation/reanalyses)
- easily do pretreatments like period or geographical domain selection, regridding, computing climatologies
- either on one dataset or on an ensemble
- plot your results
- gather the plots in a html page
- all this taking advantage of a smart cache that automatically avoids recomputing an existing result

The CliMAF documentation (<https://climaf.readthedocs.io/en/master/>) has many examples in the form of html versions of jupyter notebooks. See here:

- <https://climaf.readthedocs.io/en/master/#can-climaf-make-my-scientific-life-easier>
- <https://climaf.readthedocs.io/en/master/#cmip6-cmip5-climeri-convergence-training-session-november-19-2018>

If you are interested in following the CliMAF activity and ask users questions, subscribe to the mailing list:

<https://climaf.readthedocs.io/en/master/community.html>

2.2. The C-ESM-EP: the CliMAF Earth System Model Evaluation Platform

The C-ESM-EP (<https://github.com/jservonnat/C-ESM-EP/wiki>) is an evaluation package based on CliMAF and developed between IPSL, CNRM and CERFACS to apply evaluation diagnostics, routinely or on demand, to a list of simulations/models and compare them easily. The result is an html frontpage with links to the atlases (html pages gathering the results of the evaluation diagnostics), covering the scientific demand of the scientists working on the development of the coupled models and on the individual components (atmosphere, ocean, land surfaces, sea ice, biogeochemistry).

The C-ESM-EP documentation is here: <https://github.com/jservonnat/C-ESM-EP/wiki> Follow the main page step by step and you will find the elements to use it. If not, do not hesitate to contact J. Servonnat or post an issue on github.

2.2.1. Use C-ESM-EP in libGCM

- read complete [documentation](#) in line
- Quick documentation :
 - Environment
 - On JeanZay launch commands

```
module load singularity
container=/gpfswork/rech/psl/commun/Tools/cesmep_environment/20230611_V3.0_IPSL8.sif
idrcontmgr cp $container
```

- On Irene : nothing to do
- activate cesmep for default atlas : in config.card add the following line :

```
Cesmep=TRUE
```

2.3. Using the fast graphical display remote system at TGCC

(known as Remote Desktop System service)

Provides a fast display for graphical use : Ferret, Matplotlib, Jupyter, notebooks, etc ...

Reference :

https://www-ccrt.ccc.cea.fr/docs/irene/fr/html/toc/fulldoc/Interactive_access.html?highlight=ccc_visu#remote-desktop-system-service-nicedcv

This involves opening a GNOME session on Irene's graphical node. niceDCV will use the graphical cards on this node and on the local PC/Mac to allow a fast display. We have a fairly comfortable accelerated graphics display. We can then use Python notebooks by launching Firefox from Irene.

niceDCV can be used either through a browser (thin client) or with the DCV Viewer application (thick client)

2.3.1. From a TGCC partner network

From Irene, run the command :

```
`bash ccc_visu virtual -p v100l -A <project> -M store,work,scratch `
```

- Replace `<project>` with a project where you have computing time (gencmip6, gen2212, gen12006, ...)
- As an option to `-M`, the file systems you will access.

Normally the rest is explained by the system:

- It logs on to the graphical node
- It displays a web link that you have to click
- It opens a page that proposes to use niceDCV either through the browser (thin client) or with an application to download (thick client).

2.3.2. From any network, using ssh1, Spirit or any gateway known by TGCC

From your local terminal (Mac or PC), open a connection to ssh1 by creating a SOCKS proxy:

```
ssh -D 3128 <login>@ssh1.lsce.ipsl.fr
```

- The port number 3128 is arbitrary

Then launch a browser and ask it to go through this SOCKS proxy. According to the machine :

- `chrome --proxy-server="socks://localhost:3128"`
- `/Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome --proxy-server="socks://localhost:3128"`

It should be possible to do this with Firefox, Safari, ...

Chrome connections are now seen by Irene as coming from ssh1: we are on the TGCC partner network, and we can use niceDCV as a web client.

To use the thick client :

- Open the DCV Viewer application.
- Configure it to go through the SOCKS proxy connection `settings:localhost:1357`.
- You can then open the connection file.

2.3.3. GNOME configuration

- Don't touch the language and keyboard settings: nice DCV does that very well on its own, and you'll soon get the hang of it!
On MAC: once nice DCV is active, go to the `Connection:Keyboards settings` menu and set :
 - Use Option (■) as local modifier
 - Use Command (■) as remote meta key
- Copy/paste: as in GNOME: `Shift-Ctrl-C/Shift-Ctrl-V` in a terminal, and `Ctrl-C/Ctrl-V` in Firefox.

2.3.4. Finally

Remember to close your sessions properly (see TGCC documentation).

2.4. ESMValTool

ESMValTool (<https://esmvaltool.org/>) is a community diagnostic and performance metrics tool for routine evaluation of Earth System Models in CMIP. The ESMValTool documentation is available here: <https://docs.esmvaltool.org/> On IPSL servers, it is available with: `module load esmvaltool` An online tutorial is available to learn ESMValTool independently: <https://tutorial.esmvaltool.org/>. It contains a quickstart guide for users of the module: <https://tutorial.esmvaltool.org/01-quickstart/>. The ESMValTool documentation provides several examples in the form of Jupyter notebooks, see here:

- <https://github.com/ESMValGroup/EGU22-short-course>
- <https://docs.esmvaltool.org/projects/ESMValCore/en/latest/example-notebooks.html>

Warning: local support for ESMValTool is not provided. Instead, support is available via the ESMValTool development team on GitHub, see here: <https://github.com/ESMValGroup/ESMValTool/discussions>