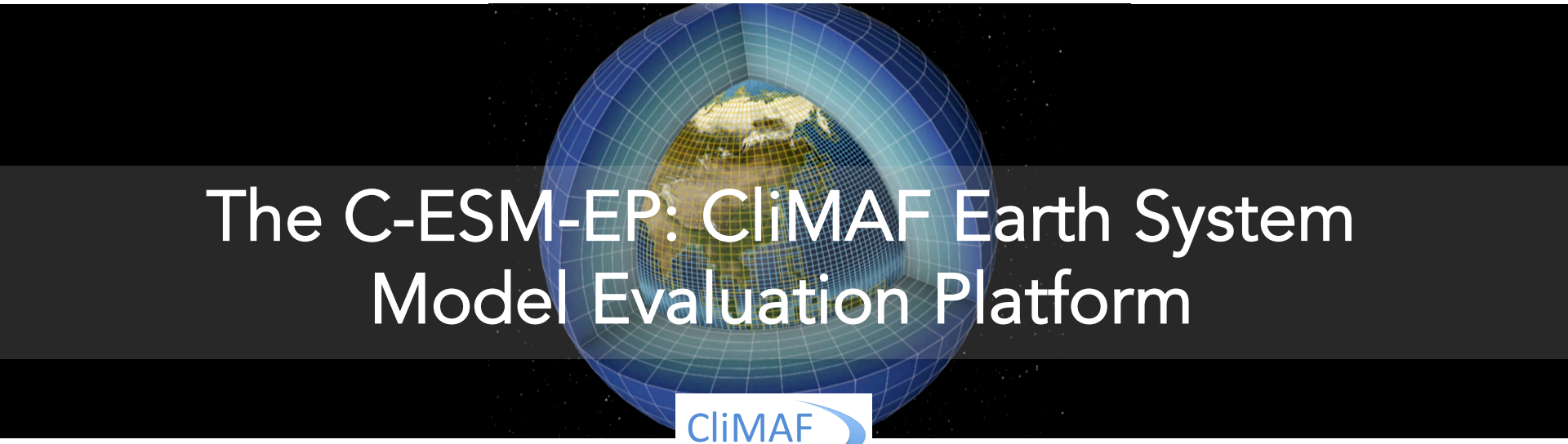
CliMAF

Sharing – Simplifying – Optimizing

and



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

CliMAF

Sharing – Simplifying – Optimizing

CliMAF / C-ESM-EP in brief

CliMAF: a python library to help you do your daily analyses and build your own routine tool

C-ESM-EP: a CliMAF-based package to evaluate/compare models/simulations (pre-defined sets of diagnostics and user diagnostics) and display the results in html pages

CliMAF

Sharing – Simplifying – Optimizing

CliMAF core team: Stéphane Sénési, Jérôme Servonnat, Ludivine Vignon
+ contributors: Marie-Pierre Moine, Emilia Sanchez-Gomez, Olivier Marti, Patrick Brockmann, Sébastien Denvil

CNRM-GAME

Sciences de l'environnement

Institut Pierre Simon Laplace

CERFACS

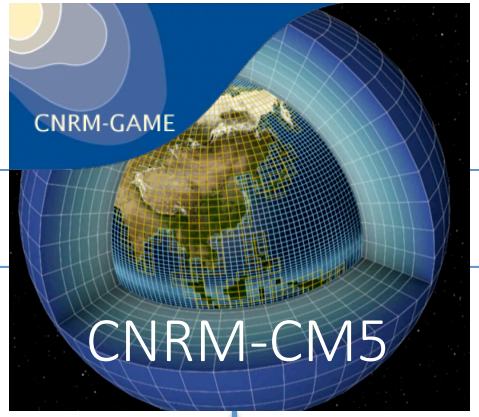
Agence Nationale de la Recherche ANR - MODÈLES NUMÉRIQUES - CONVERGENCE

WP5

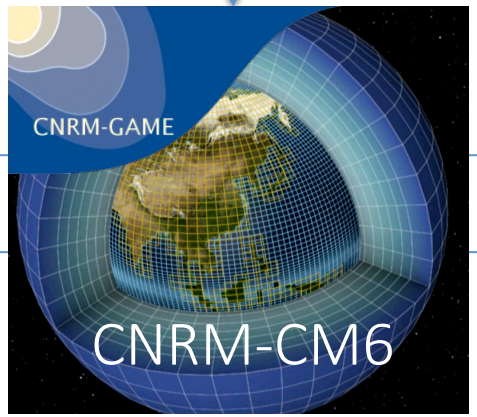
CliMAF
Sharing – Simplifying – Optimizing

Starting point: on the road to CMIP6

CMIP5



CMIP6



Starting point: on the road to CMIP6

CliMAF
Sharing – Simplifying – Optimizing

CMIP5

CNRM-GAME
CNRM-CM5

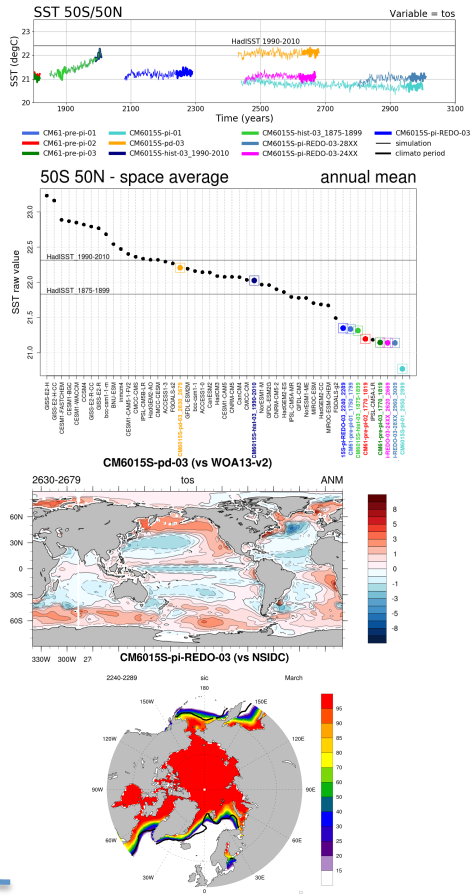
Sciences de l'environnement
Institut Pierre Simon Laplace
IPSL-CM5

Tens of simulations to analyse in a short period of time
With a lot of diagnostics

CMIP6

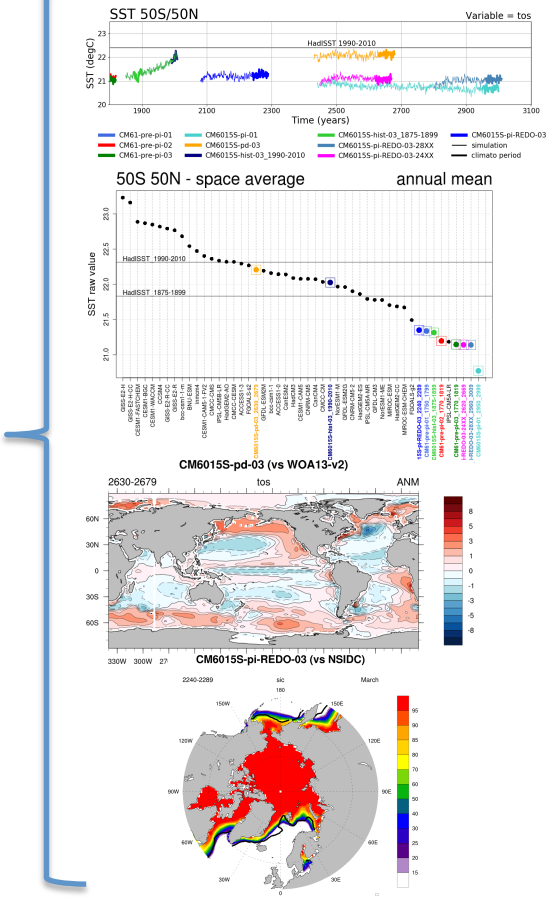
CNRM-GAME
CNRM-CM6

Sciences de l'environnement
Institut Pierre Simon Laplace
IPSL-CM6



Starting point: on the road to CMIP6

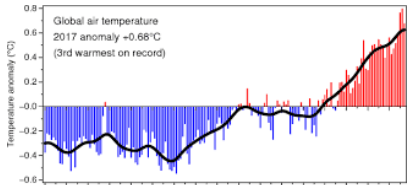
WCRP CMIP5
World Climate Research Programme



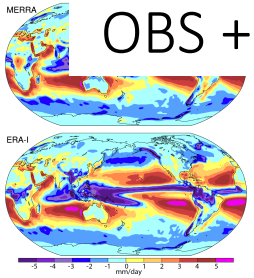
Starting point: on the road to CMIP6

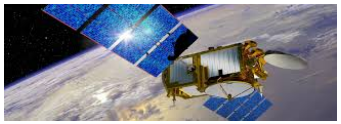
CliMAF
Sharing – Simplifying – Optimizing

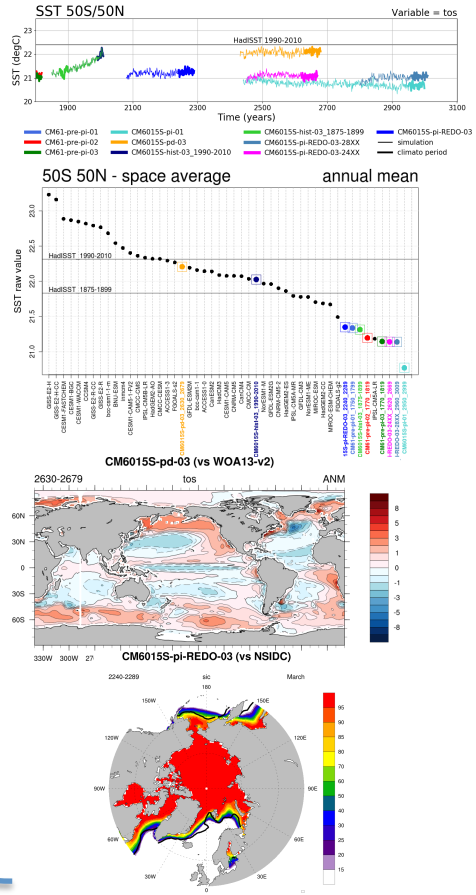
WCRP CMIP5
World Climate Research Programme



OBS + REANALYSES



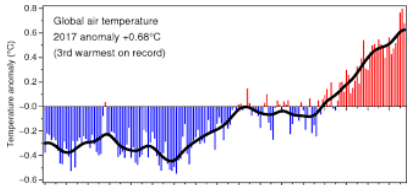




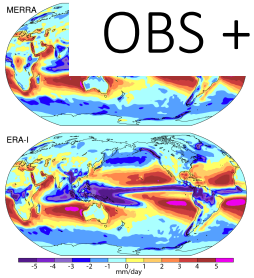
Starting point: on the road to CMIP6

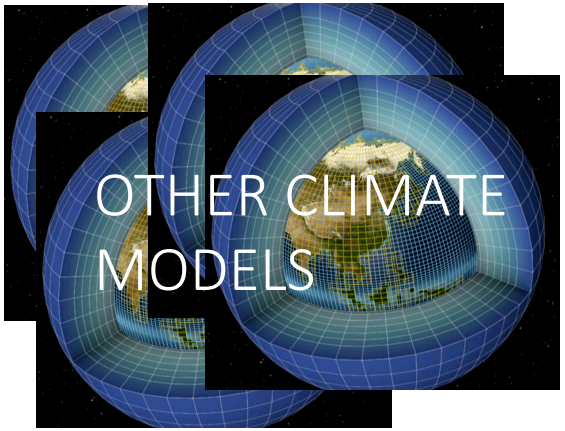
CliMAF
Sharing – Simplifying – Optimizing

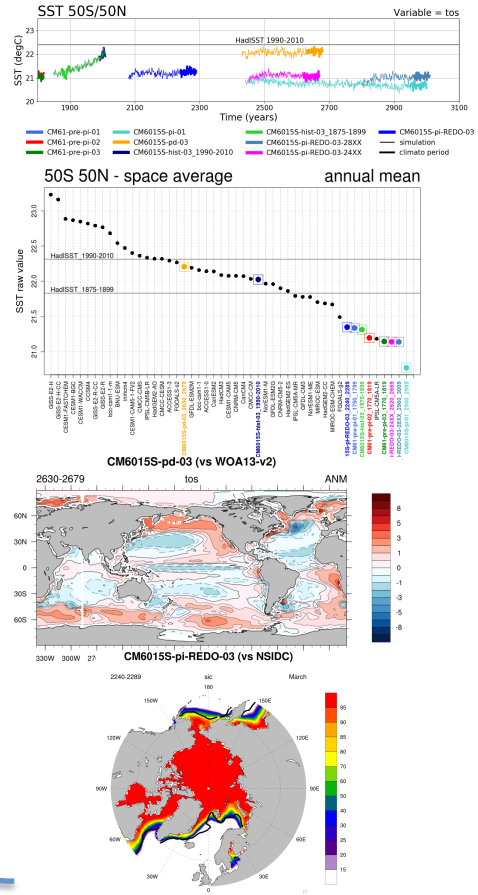
WCRP CMIP5
World Climate Research Programme



OBS + REANALYSES



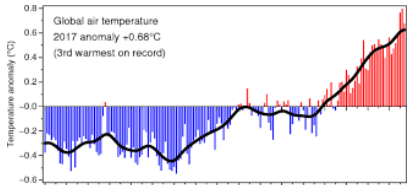




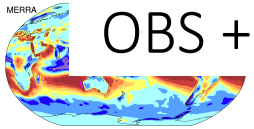
CliMAF
Sharing – Simplifying – Optimizing

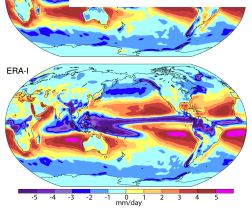
Starting point: on the road to CMIP6

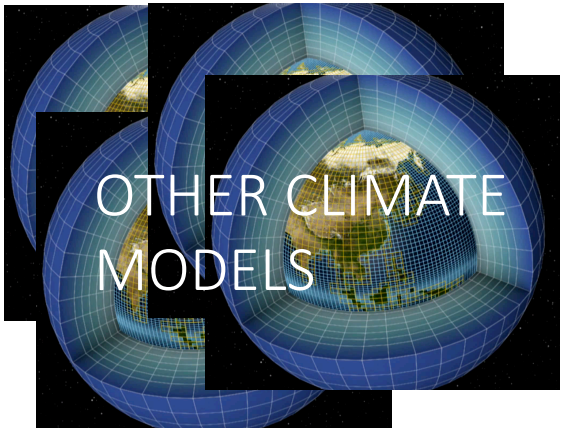
WCRP CMIP5
World Climate Research Programme

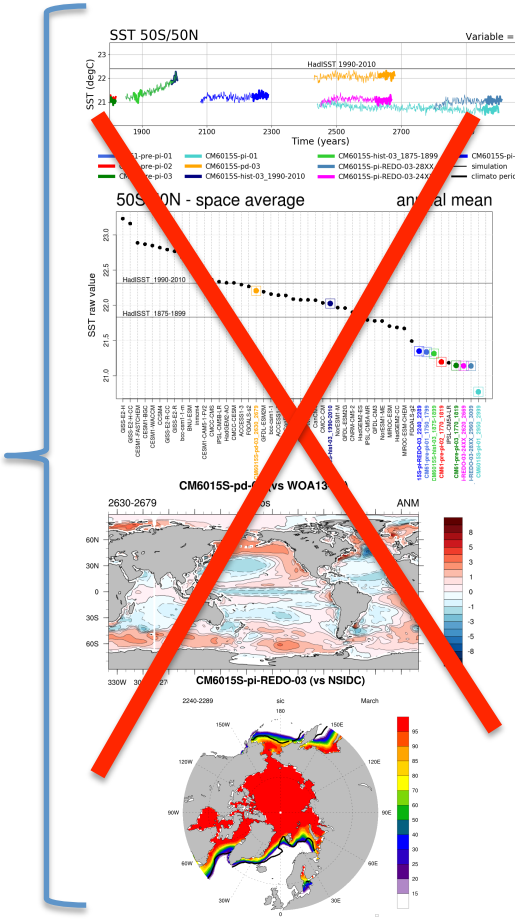


OBS + REANALYSES



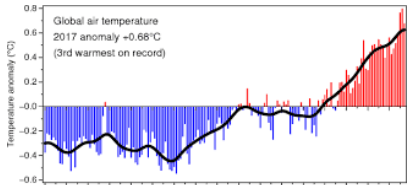


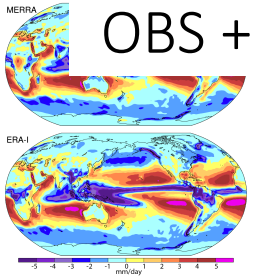




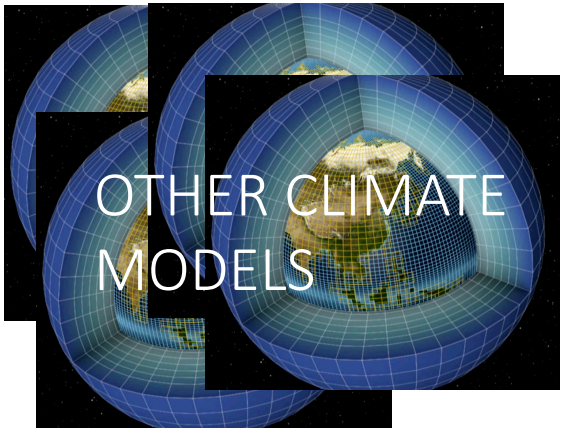
Starting point: on the road to CMIP6

WCRP CMIP5
World Climate Research Programme





OBS + REANALYSES



MY DIAGS
FOR WHAT
I WANT

CliMAF
Sharing – Simplifying – Optimizing

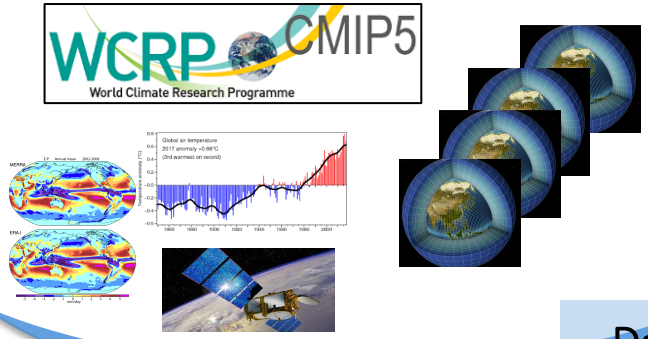
CliMAF specifications

- We have several data organizations (path/filename syntax, variable names, units... CMIP5 and CMIP6 among others) to deal with and we don't want to rewrite all of them
- We want to share diagnostics: one common library with a vocabulary that follows standards (inspired by the CMIP standards)
- We don't want to exclude existing scripts: possibility to plug any script that can be run with a command line
- We want to ease all those classic/basic treatments we do everyday
- We don't want to recompute what has already been computed once
- We want to use this library to build a big evaluation package as well as a custom set of diagnostics

In 2013, no existing package/library met all those specifications

CliMAF
Sharing – Simplifying – Optimizing

This is CliMAF



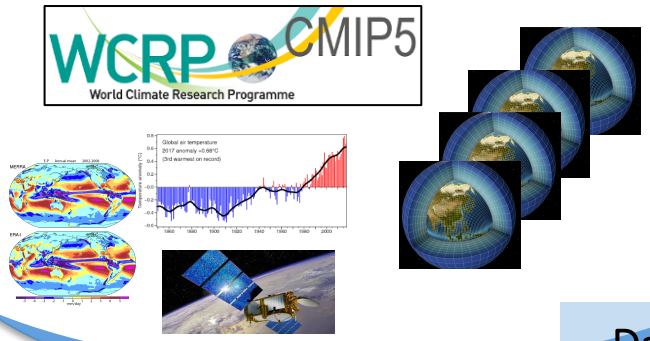
CNRM-CM, IPSL-CM, CMIP5, CMIP6, CORDEX, references (obs+reanalyses), other models….

Data access that standardizes your data on-the-fly = share diagnostics/code with other modelling groups

CliMAF dataset / ensembles = Python object with standard attributes

CliMAF
Sharing – Simplifying – Optimizing

This is CliMAF



CNRM-CM, IPSL-CM, CMIP5, CMIP6, CORDEX, references (obs+reanalyses), other models….

Data access that standardizes your data on-the-fly = share diagnostics/code with other modelling groups

CliMAF dataset / ensembles = Python object with standard attributes

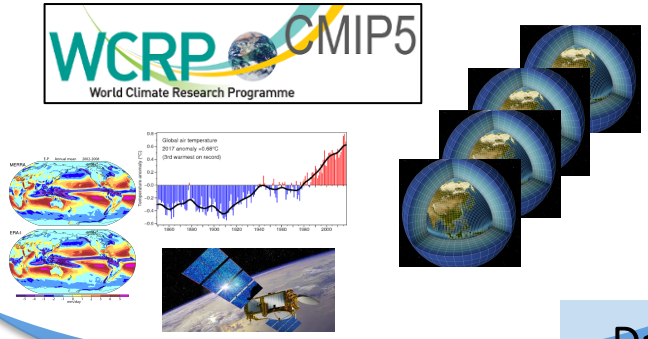
Easy solution for common data treatments: period and geographical selection, averages, regridding, plot…

Plug my own script = make it a CliMAF operator

A must for postdocs and PhD students: no need to reinvent the wheel!

This is CliMAF

CliMAF
Sharing – Simplifying – Optimizing



CNRM-CM, IPSL-CM, CMIP5, CMIP6, CORDEX, references (obs+reanalyses), other models….

Data access that standardizes your data on-the-fly = share diagnostics/code with other modelling groups

CliMAF dataset / ensembles = Python object with standard attributes

Easy solution for common data treatments: period and geographical selection, averages, regridding, plot…

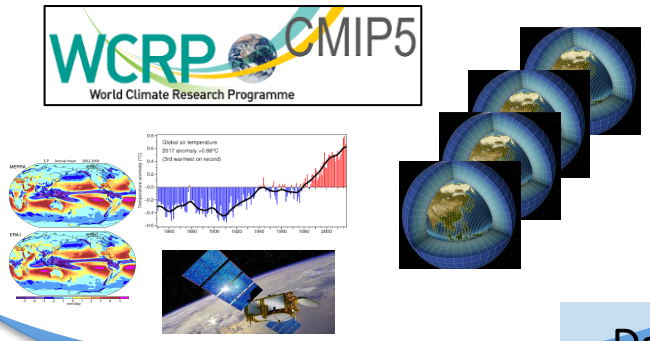
Plug my own script = make it a CliMAF operator

Netcdf file or figure

CliMAF automatically handles outputs and the existing results ; smart cache, no recomputing!

This is CliMAF

CliMAF
Sharing – Simplifying – Optimizing



CNRM-CM, IPSL-CM, CMIP5, CMIP6, CORDEX, references (obs+reanalyses), other models….

Data access that standardizes your data on-the-fly = share diagnostics/code with other modelling groups

CliMAF dataset / ensembles = Python object with standard attributes

Easy solution for common data treatments: period and geographical selection, averages, regridding, plot…

Plug my own script = make it a CliMAF operator

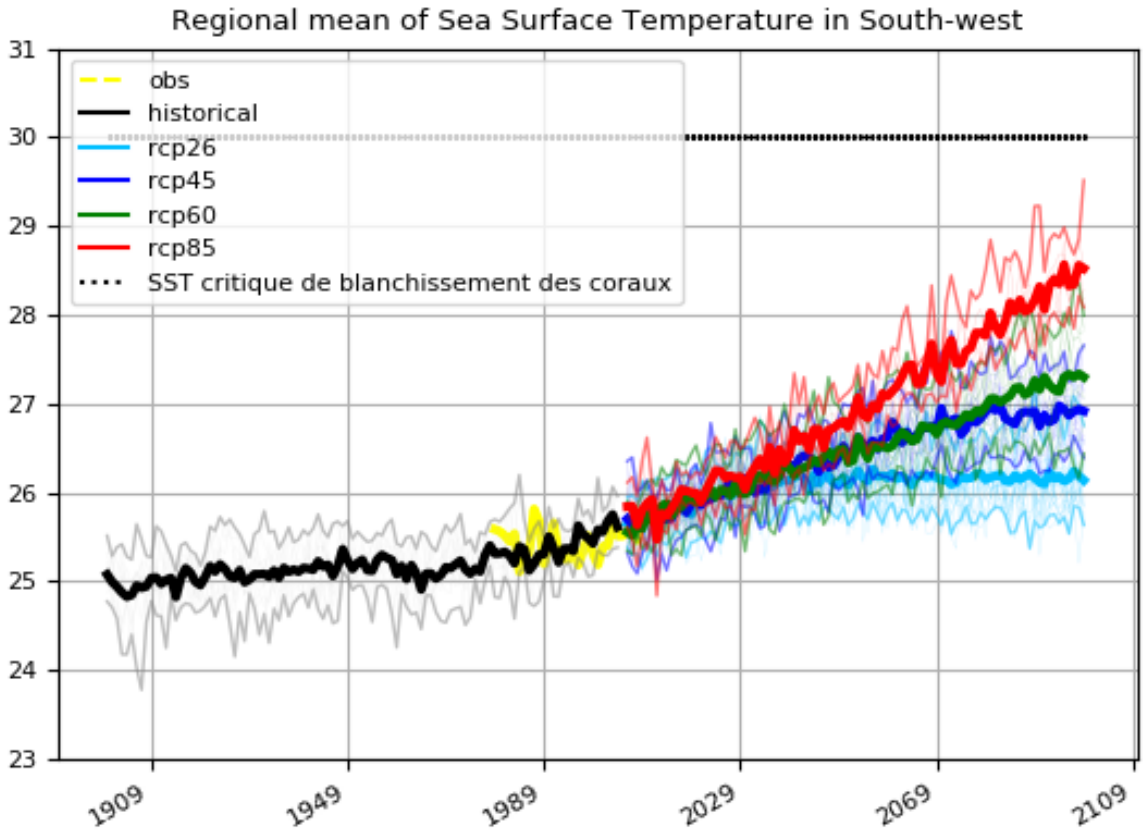
Netcdf file or figure

CliMAF automatically handles outputs and the existing results ; smart cache, no recomputing!

Html pages (share results with collaborators)

CliMAF
Sharing – Simplifying – Optimizing

Example



SST in **CMIP5** historical and scenario simulations for an impact study (H. Dayan)
Data access and pre-processing with CliMAF, and plot with a personal python script

CliMAF
Sharing – Simplifying – Optimizing

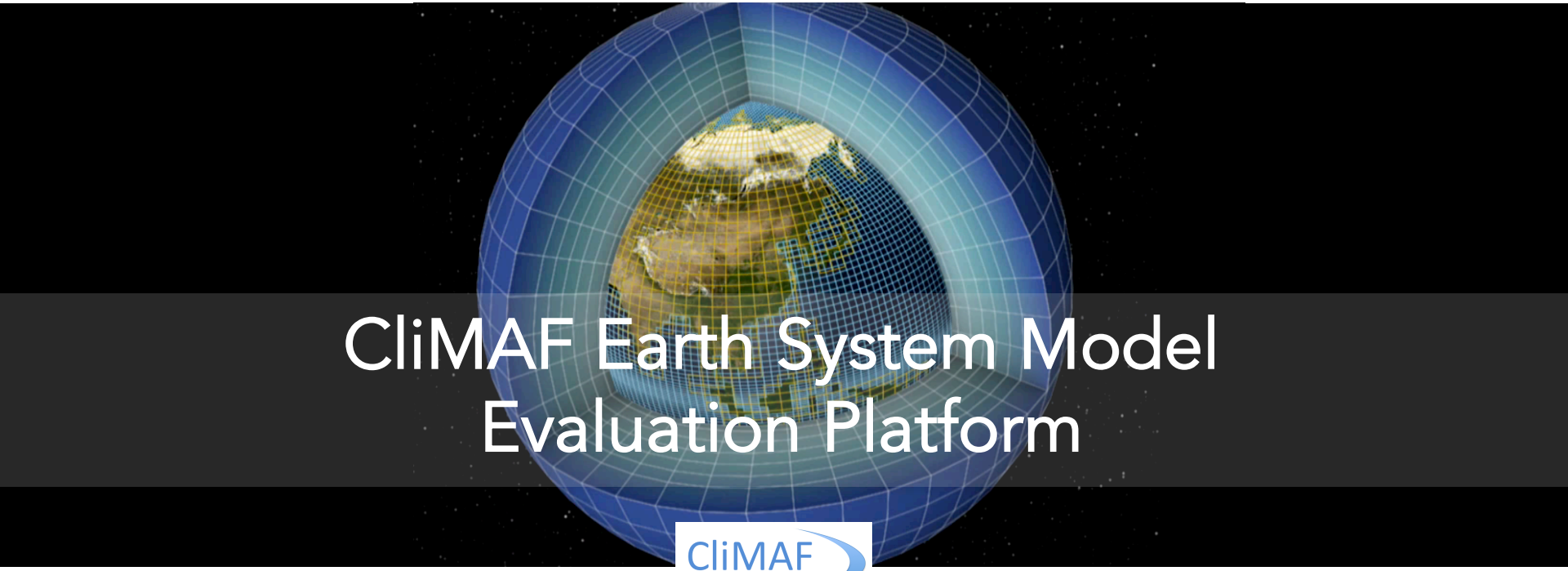
Technical points in short

CliMAF is a 'command line' manager : a scripting facility which allows to launch and pipe user-provided diagnostic scripts (any language) and binaries

- Works on CF-compliant netcdf files
- Provides all CDO operators for the treatments
- And uses NCL scripts for the plots
- Easy to install (clone of git repo, no compilation, and add to your PYTHONPATH):
  https://github.com/senesis/climaf
- Documentation: http://climaf.readthedocs.io/

CliMAF
Sharing – Simplifying – Optimizing

The C-ESM-EP

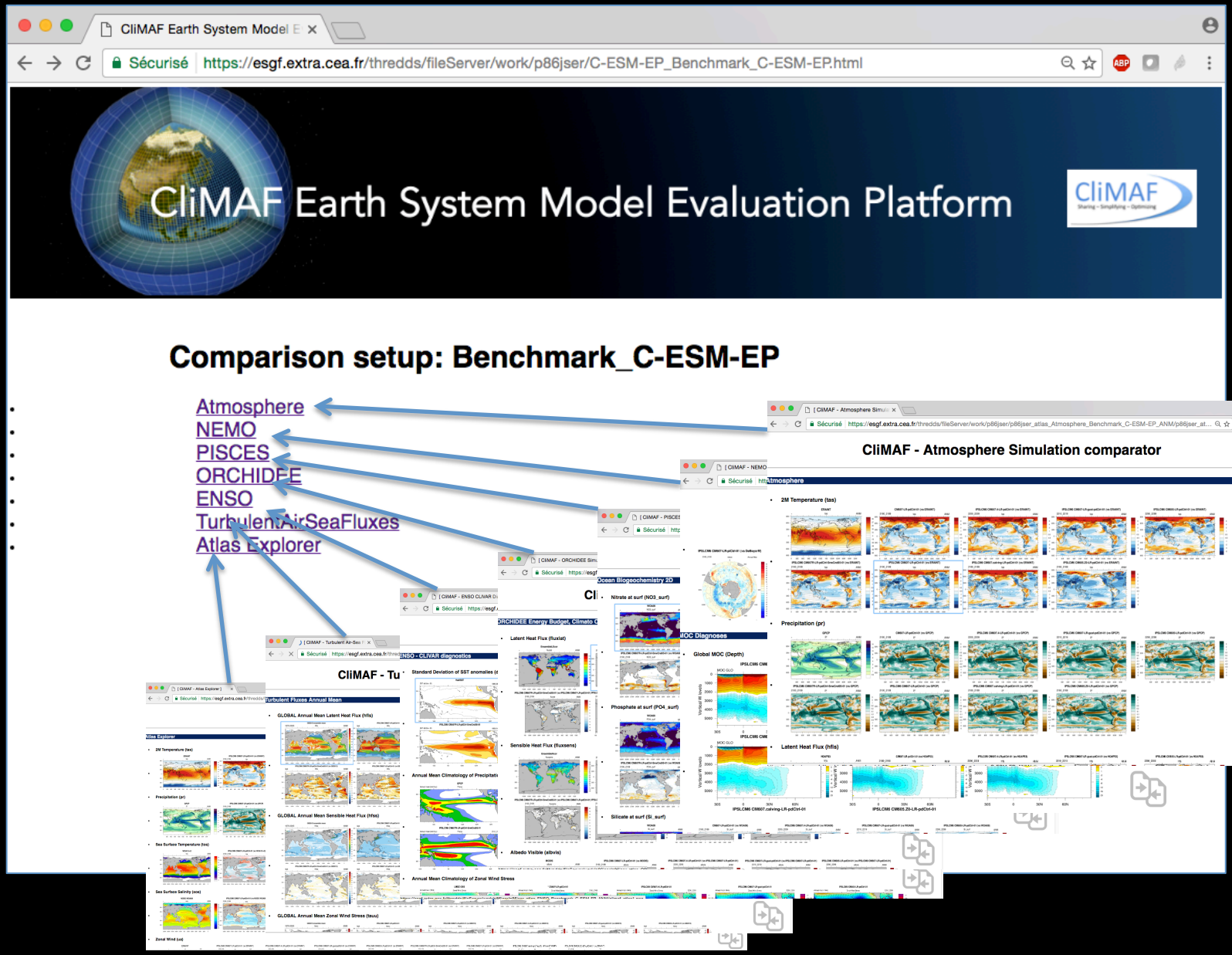


CliMAF Earth System Model Evaluation Platform

CliMAF
Sharing – Simplifying – Optimizing

Evaluating/comparing a set of simulations/models at IPSL/CNRM

Agence Nationale de la Recherche
ANR - CONVERGENCE
MODÈLES NUMÉRIQUES


CliMAF Earth System Model E
Sécurisé https://esgf.extra.cea.fr/thredds/fileServer/work/p86jser/C-ESM-EP_Benchmark_C-ESM-EP.html
CliMAF Earth System Model Evaluation Platform
CliMAF
Comparison setup: Benchmark_C-ESM-EP
Atmosphere
NEMO
PISCES
ORCHIDEE
ENSO
TurbulentAirSeaFluxes
Atlas Explorer
CliMAF - Atmosphere Simulation comparator

The C-ESM-EP in a nutshell

The quick way to use the C-ESM-EP

1. Copy the sources in a working directory:

```
cd my_working_directory
git clone https://github.com/jservonnat/C-ESM-EP.git
or
git clone jservon@ciclad.ipsl.jussieu.fr:~/C-ESM-EP/git C-ESM-EP
```

2. Setup your comparison:

```
cd C-ESM-EP
cp -r standard_comparison/ my_comparison/
```

3. Run all the components together or just a subset:

```
python run_C-ESM-EP.py my_comparison [NEMO_main,ORCHIDEE]
```

5. See the results on the URL returned by run_C-ESM-EP.py

```
-- The CliMAF ESM Evaluation Platform will be available here:
--
--   https://vesg.ipsl.upmc.fr/thredds/fileServer/IPSLFS/jservon/C-ESM-EP/all_components_demo_jservon/C-ESM-EP_all_components_demo.html
--
--
-- html file can be seen here:
-- /prodigfs/ipslfs/dods/jservon/C-ESM-EP/all_components_demo_jservon/C-ESM-EP_all_components_demo.html
(PMP_nightly-nox) jservon@ciclad-ng:~/C-ESM-EP/work>
```

Adding your simulations

the python list 'models' in datasets_setup.py

```
models = [

      dict(project='IGCM_OUT', login='p86fair', simulation='CM6014-pd-splith-01', color='green' ),
      dict(project='IGCM_OUT', login='p86maf', simulation='CM6014-pd-split-D-01', color='red'),
      dict(project='IGCM_OUT', login='p86maf', simulation='CM6014-pd-ttop-01', color='blue'),

      dict(project='IGCM_OUT', login='p86ghatt', model='LMDZOR', status='PROD',
           experiment='ref4438', simulation='CL5.4438.L6010.ref'),
      dict(project='IGCM_OUT', login='p86ghatt', model='LMDZOR', status='PROD',
           experiment='ref4438', simulation='CL5.4438.L6010.alt1'),

      dict(project='IGCM_OUT', login='p529bast', model='OL2', status='PROD',
           experiment='ref4783', simulation='FG2.4783.v3'),

]

# -- Provide a set of common keys to the elements of models
# ------------------------------------------------------------------------ >
common_keys = dict(
           root='/ccc/store/cont003/thredds', login='*',
           model='IPSLCM6',
           frequency='monthly',
           clim_period='last_10Y',
           ts_period='full',
           )
```

Adding your simulations

the python list 'models'

```
models = [

      dict(project='IGCM_OUT', login='p86fair', simulation='CM6014-pd-splith-01', color='green' ),
      dict(project='IGCM_OUT', login='p86maf', simulation='CM6014-pd-split-D-01', color='red'),
      dict(project='IGCM_OUT', login='p86maf', simulation='CM6014-pd-ttop-01', color='blue'),

      dict(proje
           exper
      dict(proje
           exper

      dict(proje
           exper

]

# -- Provide a s
# --------------
common_keys = di
           root=
           model
           frequ
           clim_
           ts_pe
           )
```

```
# -- CliMAF data access

dat1 = ds(   project = 'CMIP5',
             model = 'IPSL-CM5A-LR', experiment = 'historical',
             simulation = 'r1i1p1', variable = 'tas',
             frequency = 'monthly', period = '1980-2005' )

dat2 = ds(   project = 'IGCM_OUT',
             root = '/ccc/store/cont003/thredds', login = 'p86caub',
             model = 'IPSLCM6', simulation = 'CM605-LR-pdCtrl01',
             frequency = 'seasonal', clim_period = '2020_2029',
             variable = 'tas' )

dat3 = ds(   project = 'ref_climatos',
             variable = 'tas', product = 'ERAInterim' )
```

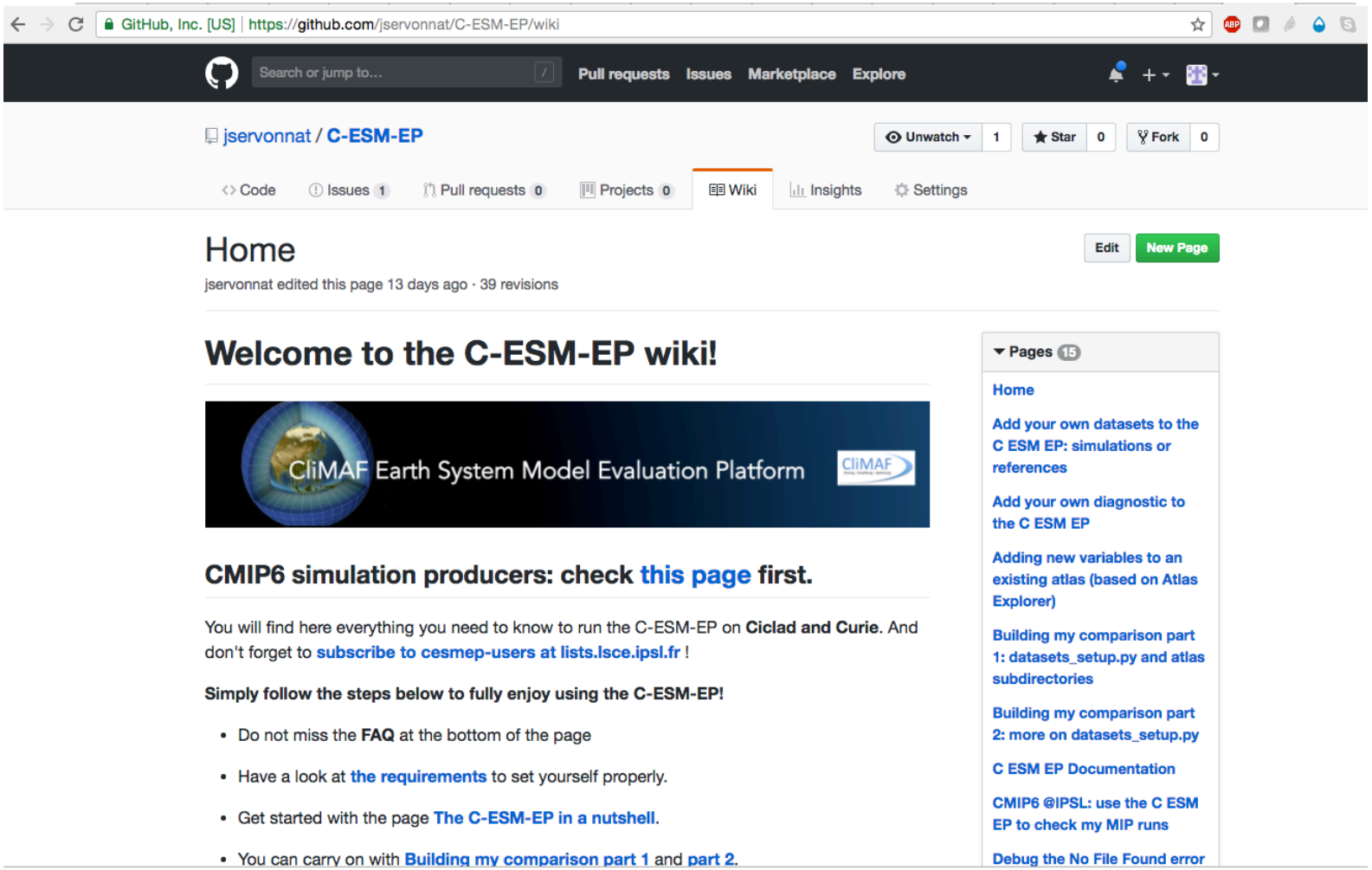
Scientific content

Community-defined sets of diagnostics

- **Atmosphere:** 6 atlases (LMD)
  - bias maps for ANM, DJF and JJA:
    - Global or polar stereographic projections
    - Surface, standard pressure levels, and zonal means
- **Ocean:** 4 atlases with the NEMO group
  - Bias maps (surface and at depth, physics and biogeochemistry)
  - Zonal averages
  - MOC diagnostics
  - Sea ice, mixed layer depth
- **Land Surfaces**:
  - climatologies, bias maps and model-model differences
  - Water, energy and carbon budget
- **ENSO:** CLIVAR diagnostics
- **Turbulent Air-Sea Fluxes:** 2 atlases Gainusa-Bogdan et al 2016, Hotelling Test
- **Large scale Metrics:** parallel coordinates with PCMDI metrics package
- **Tuning Metrics:** metrics on SST 50°S/50°N
- **Atlas Explorer**

The C-ESM-EP on github

Code, Wiki, issues



CliMAF
Sharing – Simplifying – Optimizing

Conclusion

- Should I use CliMAF or C-ESM-EP? Simple answer is:
  - The C-ESM-EP if I want the collection (or a subset) of diagnostics
  - CliMAF for anything else
  - Use examples to see how to do your own atlas
- And an other answer (that involves a bit of investment to learn both CliMAF and the C-ESM-EP): adding new diagnostics to the C-ESM-EP

=> If your diagnostic is about climatologies, difference maps, there are easy ways to do them with the C-ESM-EP = Atlas Explorer

- Documentation: http://climaf.readthedocs.io/ and https://github.com/jservonnat/C-ESM-EP/wiki

- Mailing lists: climaf.users@meteo.fr and cesmep-users@lists.lsce.ipsl.fr

Final word

- CliMAF and the C-ESM-EP are not magic!

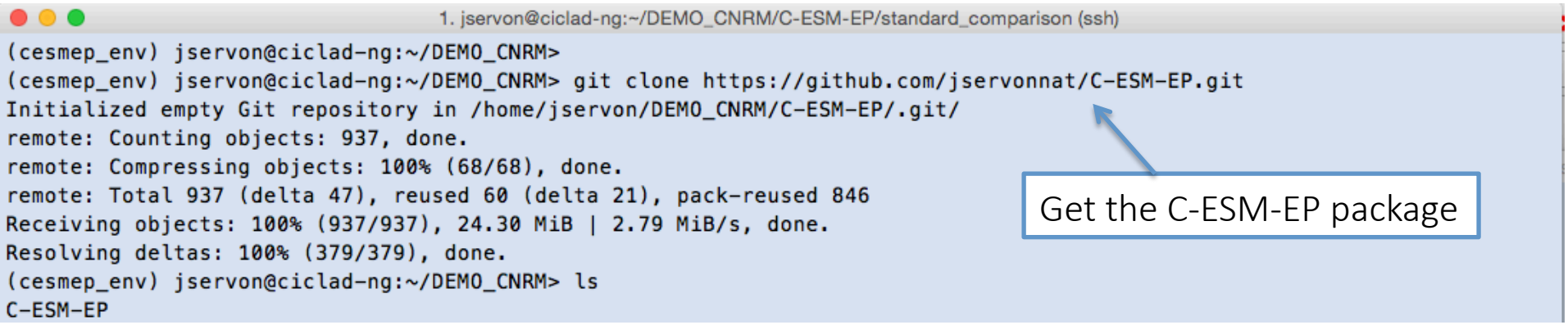
It's been developed by humans. And it does what you tell it to do.

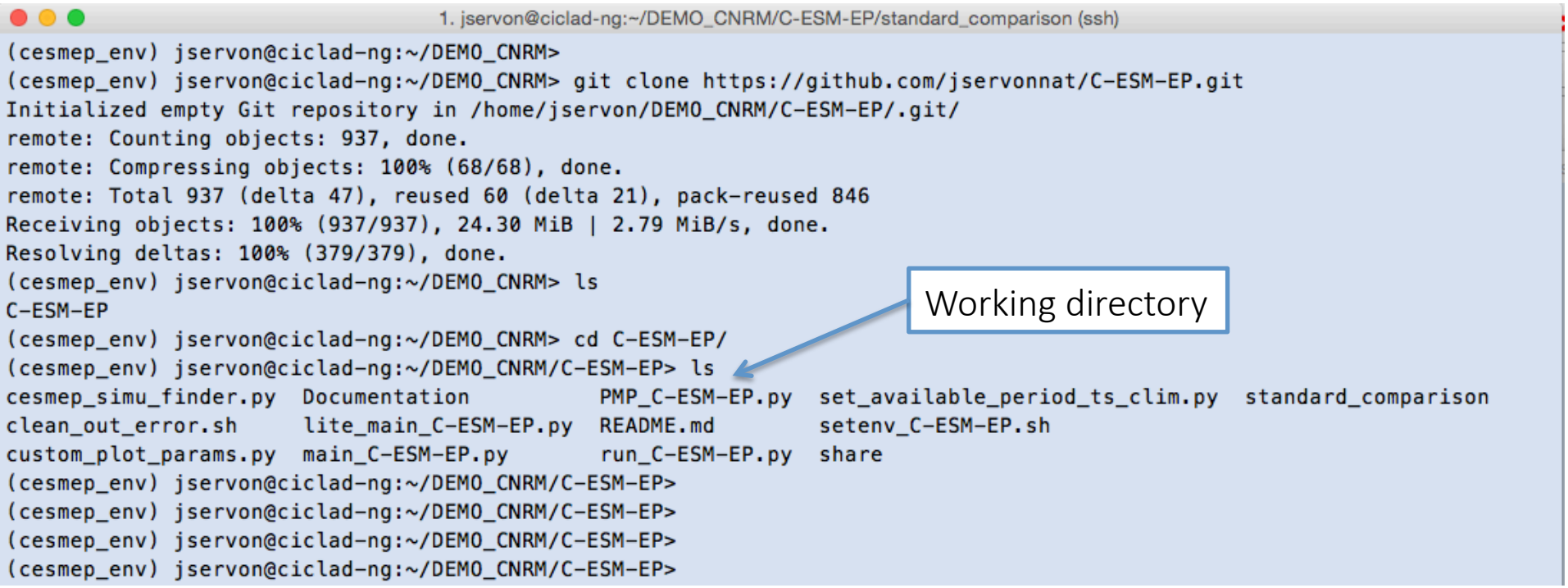
⇒ It's not because it's smart that you do not have to specify the name of your simulation correctly

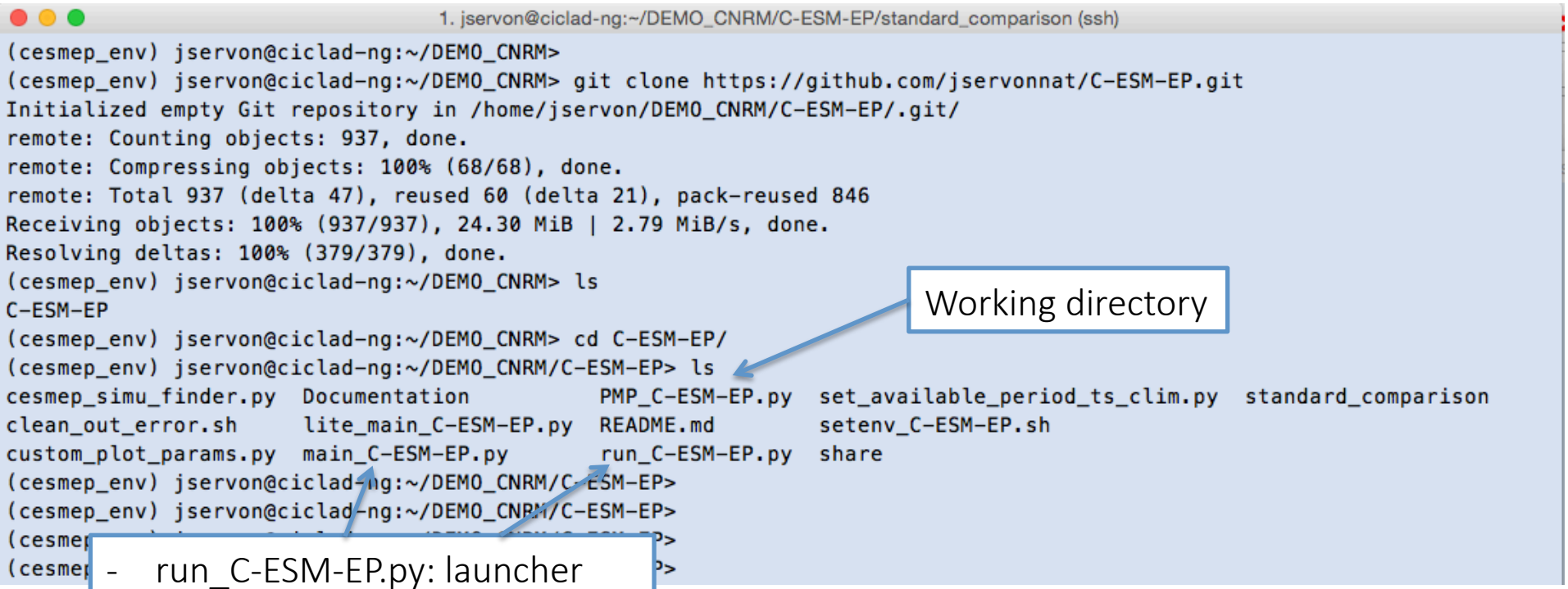
⇒ You need to learn it

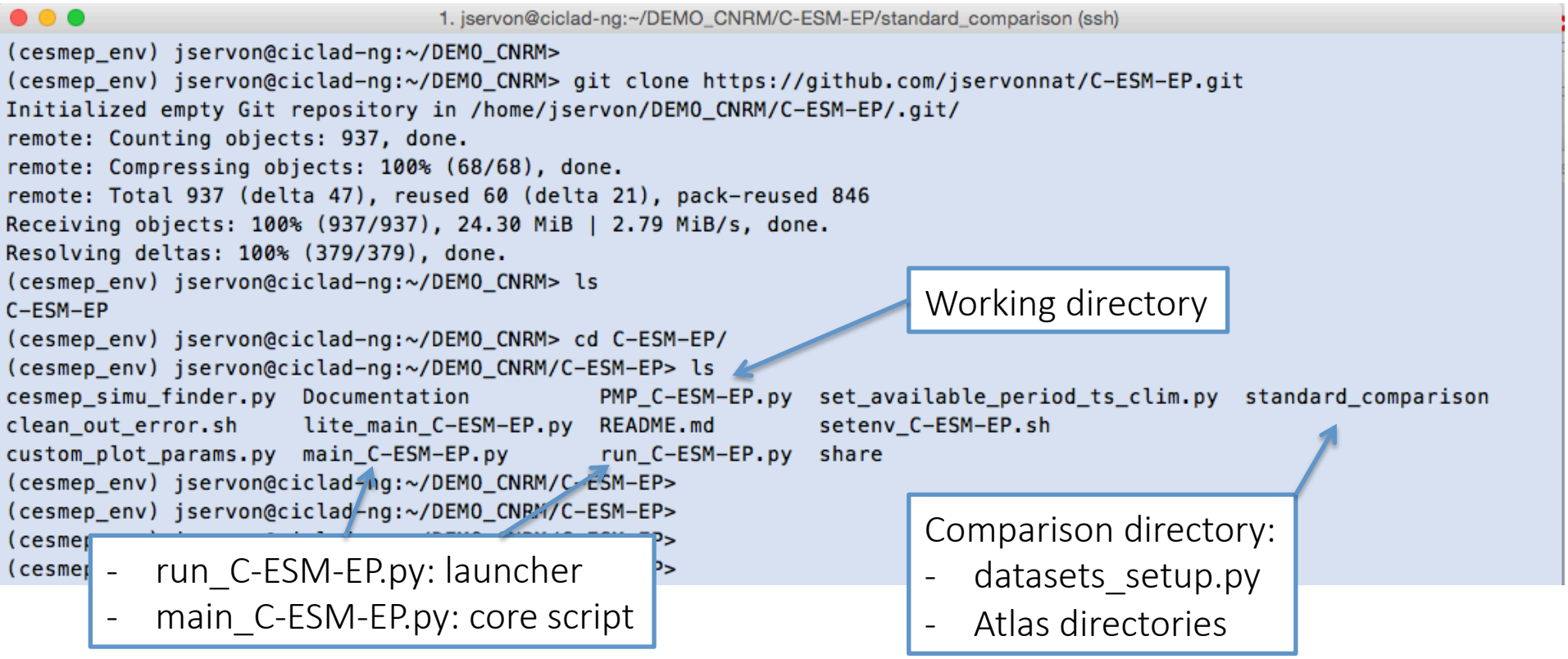
⇒ Upcoming weekly hands-on/practicals sessions! (on Thursday afternoon)

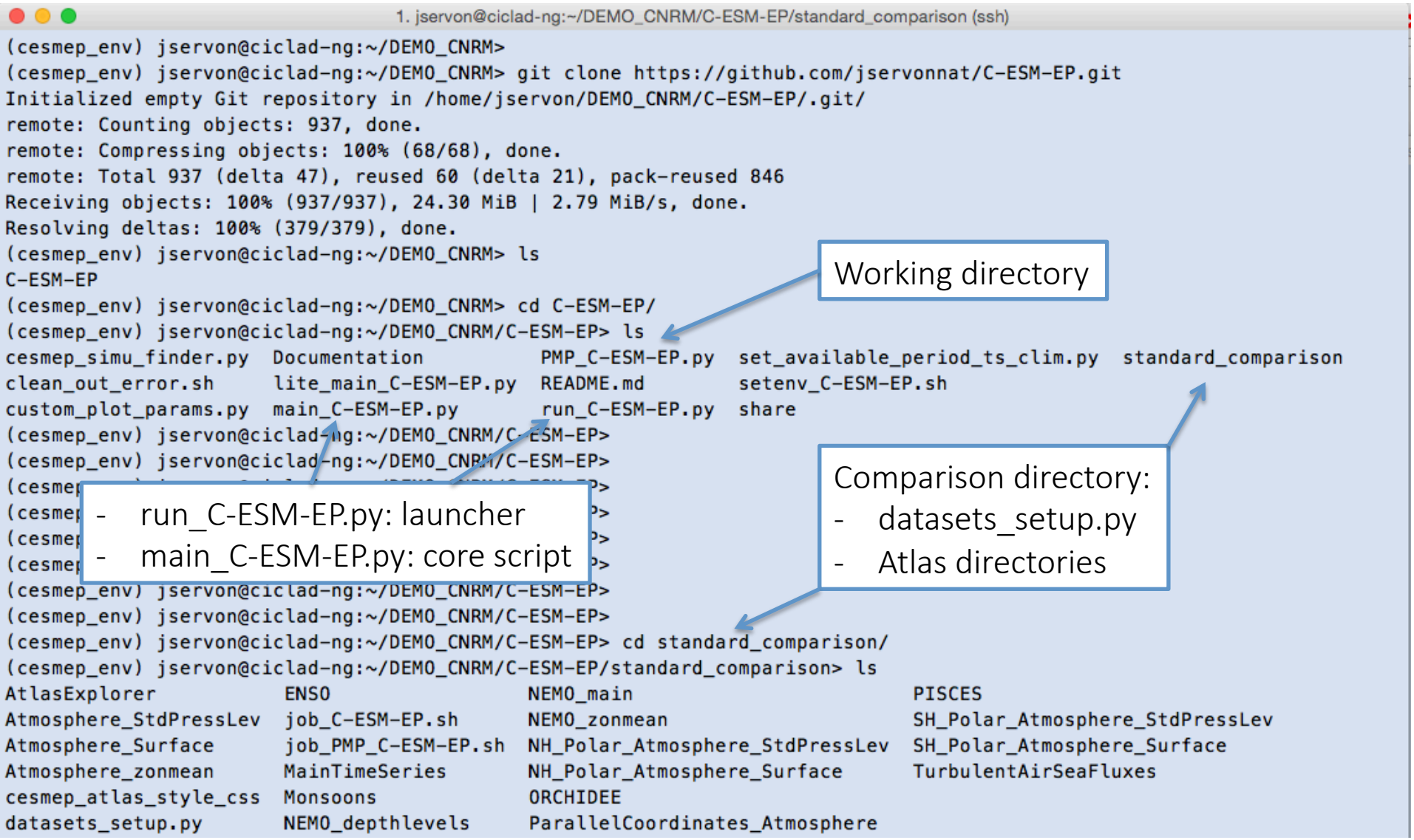
Questions?


1. jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP/standard_comparison (ssh)
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> git clone https://github.com/jservonnat/C-ESM-EP.git
Initialized empty Git repository in /home/jservon/DEMO_CNRM/C-ESM-EP/.git/
remote: Counting objects: 937, done.
remote: Compressing objects: 100% (68/68), done.
remote: Total 937 (delta 47), reused 60 (delta 21), pack-reused 846
Receiving objects: 100% (937/937), 24.30 MiB | 2.79 MiB/s, done.
Resolving deltas: 100% (379/379), done.
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> ls
C-ESM-EP
Get the C-ESM-EP package


1. jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP/standard_comparison (ssh)
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> git clone https://github.com/jservonnat/C-ESM-EP.git
Initialized empty Git repository in /home/jservon/DEMO_CNRM/C-ESM-EP/.git/
remote: Counting objects: 937, done.
remote: Compressing objects: 100% (68/68), done.
remote: Total 937 (delta 47), reused 60 (delta 21), pack-reused 846
Receiving objects: 100% (937/937), 24.30 MiB | 2.79 MiB/s, done.
Resolving deltas: 100% (379/379), done.
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> ls
C-ESM-EP
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> cd C-ESM-EP/
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP> ls
cesmep_simu_finder.py Documentation   PMP_C-ESM-EP.py set_available_period_ts_clim.py standard_comparison
clean_out_error.sh   lite_main_C-ESM-EP.py README.md   setenv_C-ESM-EP.sh
custom_plot_params.py main_C-ESM-EP.py   run_C-ESM-EP.py share
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
Working directory


1. jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP/standard_comparison (ssh)
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> git clone https://github.com/jservonnat/C-ESM-EP.git
Initialized empty Git repository in /home/jservon/DEMO_CNRM/C-ESM-EP/.git/
remote: Counting objects: 937, done.
remote: Compressing objects: 100% (68/68), done.
remote: Total 937 (delta 47), reused 60 (delta 21), pack-reused 846
Receiving objects: 100% (937/937), 24.30 MiB | 2.79 MiB/s, done.
Resolving deltas: 100% (379/379), done.
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> ls
C-ESM-EP
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> cd C-ESM-EP/
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP> ls
cesmep_simu_finder.py Documentation PMP_C-ESM-EP.py set_available_period_ts_clim.py standard_comparison
clean_out_error.sh lite_main_C-ESM-EP.py README.md setenv_C-ESM-EP.sh
custom_plot_params.py main_C-ESM-EP.py run_C-ESM-EP.py share
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
Working directory
- run_C-ESM-EP.py: launcher
- main_C-ESM-EP.py: core script


1. jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP/standard_comparison (ssh)
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> git clone https://github.com/jservonnat/C-ESM-EP.git
Initialized empty Git repository in /home/jservon/DEMO_CNRM/C-ESM-EP/.git/
remote: Counting objects: 937, done.
remote: Compressing objects: 100% (68/68), done.
remote: Total 937 (delta 47), reused 60 (delta 21), pack-reused 846
Receiving objects: 100% (937/937), 24.30 MiB | 2.79 MiB/s, done.
Resolving deltas: 100% (379/379), done.
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> ls
C-ESM-EP
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> cd C-ESM-EP/
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP> ls
cesmep_simu_finder.py Documentation PMP_C-ESM-EP.py set_available_period_ts_clim.py standard_comparison
clean_out_error.sh lite_main_C-ESM-EP.py README.md setenv_C-ESM-EP.sh
custom_plot_params.py main_C-ESM-EP.py run_C-ESM-EP.py share
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
Working directory
- run_C-ESM-EP.py: launcher
- main_C-ESM-EP.py: core script
Comparison directory:
- datasets_setup.py
- Atlas directories


1. jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP/standard_comparison (ssh)
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> git clone https://github.com/jservonnat/C-ESM-EP.git
Initialized empty Git repository in /home/jservon/DEMO_CNRM/C-ESM-EP/.git/
remote: Counting objects: 937, done.
remote: Compressing objects: 100% (68/68), done.
remote: Total 937 (delta 47), reused 60 (delta 21), pack-reused 846
Receiving objects: 100% (937/937), 24.30 MiB | 2.79 MiB/s, done.
Resolving deltas: 100% (379/379), done.
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> ls
C-ESM-EP
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> cd C-ESM-EP/
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP> ls
cesmep_simu_finder.py Documentation PMP_C-ESM-EP.py set_available_period_ts_clim.py standard_comparison
clean_out_error.sh lite_main_C-ESM-EP.py README.md setenv_C-ESM-EP.sh
custom_plot_params.py main_C-ESM-EP.py run_C-ESM-EP.py share
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP> cd standard_comparison/
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP/standard_comparison> ls
AtlasExplorer ENSO NEMO_main PISCES
Atmosphere_StdPressLev job_C-ESM-EP.sh NEMO_zonmean SH_Polar_Atmosphere_StdPressLev
Atmosphere_Surface job_PMP_C-ESM-EP.sh NH_Polar_Atmosphere_StdPressLev SH_Polar_Atmosphere_Surface
Atmosphere_zonmean MainTimeSeries NH_Polar_Atmosphere_Surface TurbulentAirSeaFluxes
cesmep_atlas_style_css Monsoons ORCHIDEE
datasets_setup.py NEMO_depthlevels ParallelCoordinates_Atmosphere
Working directory
- run_C-ESM-EP.py: launcher
- main_C-ESM-EP.py: core script
Comparison directory:
- datasets_setup.py
- Atlas directories

1. jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP/standard_comparison (ssh)

```
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> git clone https://github.com/jservonnat/C-ESM-EP.git
Initialized empty Git repository in /home/jservon/DEMO_CNRM/C-ESM-EP/.git/
remote: Counting objects: 937, done.
remote: Compressing objects: 100% (68/68), done.
remote: Total 937 (delta 47), reused 60 (delta 21), pack-reused 846
Receiving objects: 100% (937/937), 24.30 MiB | 2.79 MiB/s, done.
Resolving deltas: 100% (379/379), done.
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> ls
C-ESM-EP
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM> cd C-ESM-EP/
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP> ls
cesmep_simu_finder.py  Documentation          PMP_C-ESM-EP.py  set_available_period_ts_clim.py  standard_comparison
clean_out_error.sh     lite_main_C-ESM-EP.py  README.md        setenv_C-ESM-EP.sh
custom_plot_params.py  main_C-ESM-EP.py       run_C-ESM-EP.py  share
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP> cd standard_comparison/
(cesmep  DEMO_CNRM/C-ESM-EP/standard_comparison> ls
AtlasEx                 NEMO_main                         PISCES
Atmosph          EP.sh  NEMO_zonmean                      SH_Polar_Atmosphere_StdPressLev
Atmosph      -ESM-EP.sh NH_Polar_Atmosphere_StdPressLev   SH_Polar_Atmosphere_Surface
Atmosph          eries  NH_Polar_Atmosphere_Surface       TurbulentAirSeaFluxes
cesmep_atlas_style_css  Monsoons           ORCHIDEE
datasets_setup.py       NEMO_depthlevels   ParallelCoordinates_Atmosphere
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP/standard_comparison>
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP/standard_comparison> ls NEMO_main/
params_NEMO_main.py
(cesmep_env) jservon@ciclad-ng:~/DEMO_CNRM/C-ESM-EP/standard_comparison>
```

Working directory

Comparison directory:
- datasets_setup.py
- Atlas directories

Atlas directory:
- params_atlas.py
- .e & .o files