



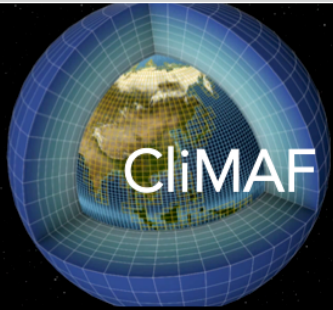
CENTRE EUROPÉEN DE RECHERCHE ET DE FORMATION AVANCÉE EN CALCUL SCIENTIFIQUE



CliMAF Earth System Model Evaluation Platform (C-ESM-EP)

***Laure Coquart, Gaëlle Rigoudy,
Jérôme Servonnat
Cluc du 08/10/2019***

La C-ESM-EP, c'est quoi ?



CliMAF Earth System Model Evaluation Platform



Comparison setup: **standard_comparison_CMIP6CNRM_triatlas_scenario_ssp370-2015**

- Main Time Series
- Atmosphere Surface
- Atmosphere Standard press. lev. - seasonal
- Atmosphere Zonal Mean
- NEMO general diagnostics
- NEMO - T & S @depth
- ENSO CLIVAR Diagnostics

Comparison directory: /data/scratch/globc/dcom/CMIP6_TOOLS/C-ESM-EP/standard_comparison_CMIP6CNRM_triatlas_scenario_ssp370-2015

- C-ESM-EP Wiki - Documentation
- CliMAF documentation

<http://cerfacs.fr/giec6>



Plan de la présentation

- ◆ Contexte
- ◆ CliMAF
- ◆ La C-ESM-EP
- ◆ Atlas atmosphériques et océaniques obtenus avec la C-ESM-EP sur les simulations TRIATLAS CNRM-ESM2-1_ssp370-r[1-5]i1p1f2 (2015-2100)
- ◆ Conclusions, perspectives

◆ La C-ESM-EP :

- Développée par Jérôme Servonnat (JS) à l'IPSL pour faire des Atlas « standards » pour évaluer un modèle de climat en cours de développement
- Ecrite en Python 2 et basée sur CliMAF (a Climate Model Assessment Framework)
- Ensemble propre d'observations et de réanalyses *ReferenceDatasets_CESMEP* (177Go)

◆ CliMAF :

- Librairie développée lors de l'ANR CONVERGENCE (2013-2017) via une collaboration entre l'IPSL/Météo-France/le Cerfacs. Elle est écrite en Python 2
- Basée sur nco, cdo, NCL pour faire les diagnostics (opérateurs surchargés) + toutes les fonctions et méthodes de la librairie
- Actuellement: Gaëlle Rigoudy (GR) a repris le développement de CliMAF à Météo-France, après le départ de Stéphane Sénési (SS) à la retraite en mai 2019.



➔ Nécessité d'Atlas standards au Cerfacs en plus de CVDP

- ESMValTool v2.0 pas disponible
- Collaboration de Marie-Pierre Moine avec JS et SS pour le développement de CliMAF dans le cadre de l'ANR CONVERGENCE (2013-2017)
- Diagnostics codés en utilisant CliMAF pour PRIMAVERA par MPM
- Début de portage de la C-ESM-EP sur nemo_lenovo par MPM mais priorite sur les simulations PRIMAVERA
- Reprise du portage de la C-ESM-EP au Cerfacs mi novembre 2018
- **Bocaux CliMAF/C-ESM-EP** avec SS puis GR au CNRM à Météo-France, en visioconférence avec JS à l'IPSL, le plus souvent possible depuis novembre 2018 le jeudi après midi, pour me former à CliMAF et surtout à la C-ESM-EP



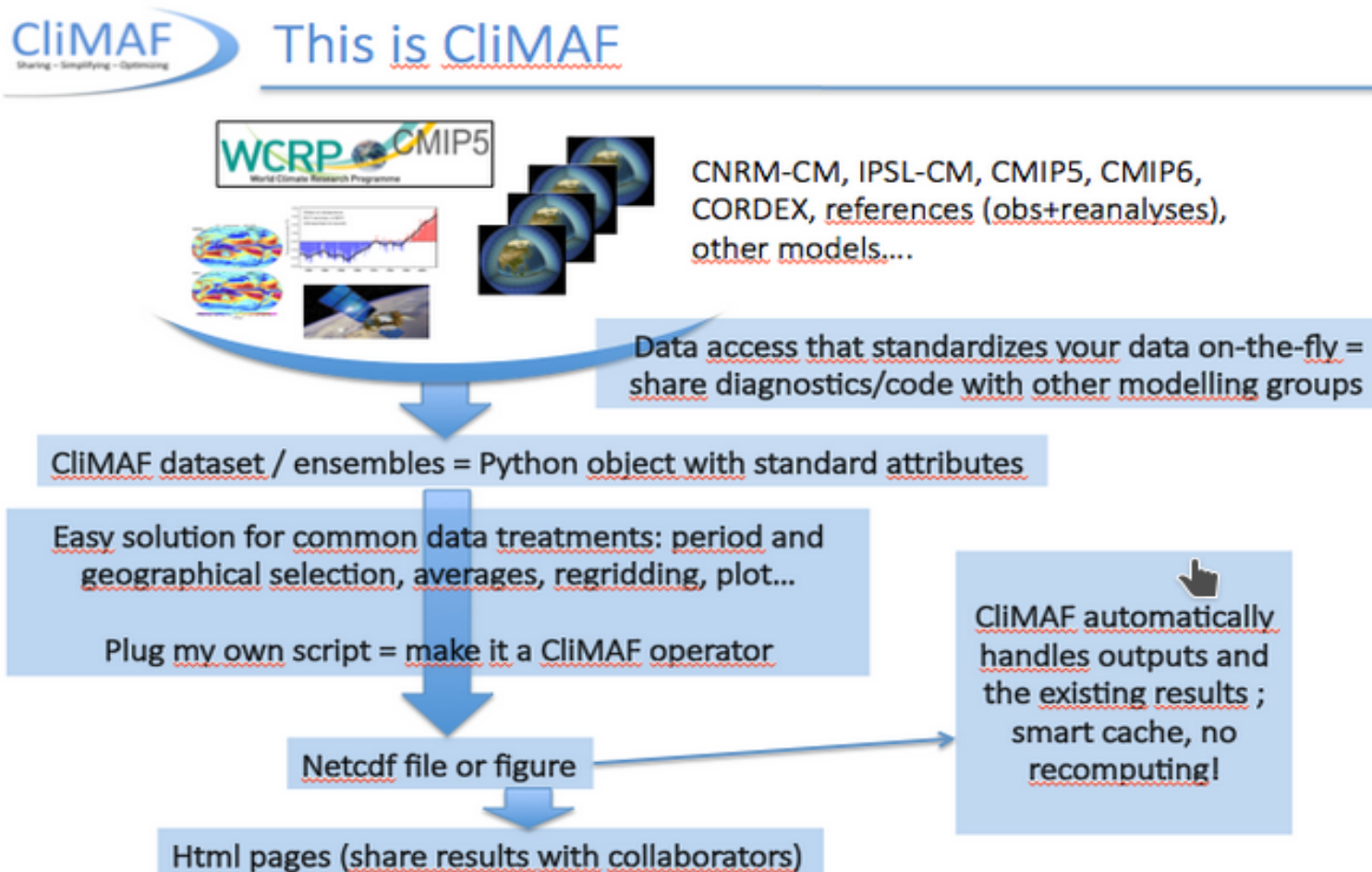
Portage de la C-ESM-EP au Cerfacs

- Gestion des données Cerfacs avec CliMAF (création d'un projet+dataset pour le spinup PRIMAVERA), calcul d'une moyenne et création d'un plot avec CliMAF, sur kraken
- Portage de (C-ESM-EP v1.0+CliMAF 1.2.11) sur kraken sur le nœud visu (02/2019)
- Après debogage, Atlas C-ESM-EP du membre r1i1p1f2 du spinup PRIMAVERA, sur kraken
- Portage de (C-ESM-EP v1.0+CliMAF 1.2.11) sur scylla où sont toutes nos données (04/2019)
- Portage de (C-ESM-EP v2.0+CliMAF 1.2.12) sur scylla (06/2019) : plus modulaire, et avec CliMAF « optimisée et nettoyée » par GR

On utilise donc maintenant (C-ESM-EP v2.0+CliMAF 1.2.12) sur scylla

La C-ESM-EP : basée sur CliMAF

<https://climaf.readthedocs.io/en/latest/index.html>



CliMAF : 'datasets' permettant de définir où sont localisées les données
+ calcul de diagnostics + création de figures + tables html

Dans les sources de la **C-ESM-EP** : `from climaf.api import *`

`python run_C-ESM-EP.py comparaison composante1,composante2,...`

Envoie en // autant
de jobs que de
composantes

comparaison contient les **composantes** qui seront tracées sur les Atlas du Cerfacs :
*MainTimeSeries ; Atmosphere_Surface ; Atmosphere_StdPressLev ;
Atmosphere_zonmean ; NEMO_main ; NEMO_depthlevels ; ENSO*

`run-C-ESM-EP.py → job-C-ESM-EP.sh`
`→ python main-C-ESM-EP.py - -comparison comparaison - -component composante[i]`

CLIMAF : 'datasets' permettant de définir où sont localisées les données + calcul de diagnostics + création de figures + tables html

Dans C-ESM-EP : main-C-ESM-EP.py pour Atmosphere_Surface

PARTIE 1 de main Récupère les instructions

- Nom du diagnostics_file (diagnostics_Atmosphere_Surface)
- Execfile (datasets_setup)

C-ESM-EP/comparaison

PARTIE 2 de main Création du tableau html avec les diags

- Execfile (param_file)
- Execfile (diagnostics_file)

C-ESM-EP/**share**/cesmep_diagnostics

C-ESM-EP/**share**/cesmep_modules

PARTIE 3 de main

- Finalise les chemins du tableau html
- Ecrit le fichier html

C-ESM-EP/comparaison/Atmosphere_Surface

La C-ESM-EP au Cerfacs

CliMAF : 'datasets' permettant de définir où sont localisées les données + calcul de diagnostics + création de figures + tables html : **ajout de 'atCerfacs' + datasets cerfacs**

➔ Dans le répertoire C-ESM-EP :

- **atCerfacs** dans run-C-ESM-EP.py pour lancer les jobs sur les queues des machines du Cerfacs
- **atCerfacs** dans setenv_C-ESM-EP.sh qui définit l'environnement (python2/venvs/cesmep1.0)
- **atCerfacs** dans locations.py qui décrit les chemins aux pages html sur le serveur web domo

Dans comparaison :

Atmosphere_Surface :

job-C-ESM-EP.sh

if atCerfacs:
Commandes SLURM pour
pour lancer
main-C-ESM-EP.py au Cerfacs

datasets_setup.py

if atCerfacs:
models = dict (dataset Cerfacs)
défini dans CliMAF

Ajout de notre
dataset Cerfacs définit
dans CliMAF

params_Atmosphere_Surface.py



Atlas C-ESM-EP de CNRM-ESM2-1_ssp370

Main Time Series: tos, tas

Atmosphere Surface (seasonal): tas, pr, hfls, hfss, psl, uas, vas, tauu, tauv, hurs, rsutcs, rsut, rlut, rlutcs
+ albt, albs, crest, crelt, crett, cress

Atmosphere Standard press. lev. – seasonal: ua, va, ta, hur, hus, zg

Atmosphere Zonal Mean (seasonal): ua, va, ta, hur, hus

NEMO General Diagnostics (seasonal): tos, wfo, sos, zos, mlotst, sic, sit, sivolu

NEMO – T & S @depth: to, so

ENSO CLIVAR Diagnostics : selon recommandations du groupe CLIVAR

http://cerfacs.fr/giec6/C-ESM-EP/index_cesmep.php

comparaison=standard_comparaison_CMIP6CNRMtriatlas_scenario_ssp370-2015



Conclusions

- ◆ Grâce aux « bocalx » du jeudi, portage de la C-ESM-EP sur scylla plus facile et plus rapide
- ◆ Intégration de nos modifications dans les versions officielles
- ◆ Logiciel modulaire, parallèle si nécessaire (mais pas encore vraiment testé sauf pour les niveaux des variables océaniques 3D)
- ◆ Support de GR et JS même en dehors des bocalx 😊
- ◆ Interpolation (bilinéaire) des simulations sur la grille des observations ou des réanalyses
- ◆ Diagnostics atmosphériques et océaniques
- ◆ Logiciel permettant de tracer tous les membres d'une simulation sur la même figure et/ou de comparer plusieurs simulations entre elles
- ◆ Pour l'instant utilisation des diagnostics tels quels (et pas tous)
- ◆ Pas d'ajout de nouvelles variables
- ◆ Version v2.0 beaucoup plus accessible qu'ESMValTool (testé sur nemo_lenovo cet été)



Perspectives

- ◆ CliMAF en Python 3, cdo ➔ mcdo, gestion des ensembles
- ◆ A long terme : Intégration des diagnostics de CVDP ? d'ESMValTool ?
- ◆ C-ESM-EP : pas de développements particuliers prévus (développement du modèle passé). Lié a CliMAF (Python 3)
- ◆ Utilisation d'ERA5 à la place d'ERA-INT pour les diagnostics atmosphériques (voir [tests](#))
- ◆ Participation aux boccux 2019-2020
- ◆ Ajout d'autres composantes, d'autres variables ?
- ◆ A plus long terme : ajout de diagnostics propres au Cerfacs ?
- ◆ En // utilisation d'ESMValTool (PRIMAVERA, APPLICATE)



Transparents en plus



Dataset dans CliMAF

```
# -- List of simulations: models = [dict(...), dict(...), ...]
# --
# -- Specify the list of 'datasets' (list of python dictionaries) you want to assess with the atlas.
# -- A CliMAF dataset is defined by:
# --     - project: a pre-defined organization of data (ex: CMIP5, IGCM_OUT, EM...)
# --     - simulation: the simulation name
# --     - frequency: monthly, seasonal, yearly...
# --     - period: the time period (if frequency = monthly or yearly)
# --       CliMAF will extract this period from the files in the directory
# --       pointed at by the request
# --     - clim_period: a character string indicating the period over which you want
# --       your climatology (pre-computed, or to be done)
# --       => pre-computed for frequency='seasonal'
# --       => to be done on demand for frequency='monthly'
# --     - ts_period: a character string indicating the period over which you want
# --       your time series
# --
```

I



Fichier datasets_setup.py

```
# -----  
from climaf.site_settings import onCiclad, atTGCC, atCNRM, atCerfacs  
from CM_atlas import *
```

I

```
if atCerfacs:  
    models = [
```

```
# Sorte de dataset mais que avec les attributs communs a toutes les variables et simus  
dict(project = 'CMIP6CERFACS', model='CNRM-ESM2-1', simulation='ssp370', institute='CNRM-CERFACS',  
    mip='ScenarioMIP', realization='r1ilp1f2', period='2015-2100', frequency='monthly',  
    customname='CNRM-ESM2-1_ssp370_r1ilp1f2'  
),  
dict(project = 'CMIP6CERFACS', model='CNRM-ESM2-1', simulation='ssp370', institute='CNRM-CERFACS',  
    mip='ScenarioMIP', realization='r2ilp1f2', period='2015-2100', frequency='monthly',  
    customname='CNRM-ESM2-1_ssp370_r2ilp1f2'  
),  
dict(project = 'CMIP6CERFACS', model='CNRM-ESM2-1', simulation='ssp370', institute='CNRM-CERFACS',  
    mip='ScenarioMIP', realization='r3ilp1f2', period='2015-2100', frequency='monthly',  
    customname='CNRM-ESM2-1_ssp370_r3ilp1f2'  
),  
dict(project = 'CMIP6CERFACS', model='CNRM-ESM2-1', simulation='ssp370', institute='CNRM-CERFACS',  
    mip='ScenarioMIP', realization='r4ilp1f2', period='2015-2100', frequency='monthly',  
    customname='CNRM-ESM2-1_ssp370_r4ilp1f2'  
),  
dict(project = 'CMIP6CERFACS', model='CNRM-ESM2-1', simulation='ssp370', institute='CNRM-CERFACS',  
    mip='ScenarioMIP', realization='r5ilp1f2', period='2015-2100', frequency='monthly',  
    customname='CNRM-ESM2-1_ssp370_r5ilp1f2'  
),
```

I



Fichier params_Atmsosphere_Surface.py

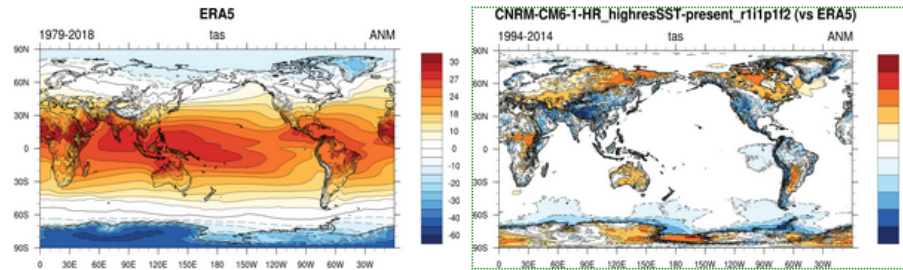
```
# ----- >
# -- Atmosphere diagnostics
# -- This section is based on the same mechanisms as Atlas Explorer; it is
# -- thus possible to use the functionalities (python dictionaries to add options
# -- with a variable)
# ----- >
my_seasons = ['ANM', 'DJF', 'JJA']
atlas_explorer_variables_list = ['tas', 'pr', 'hfls', 'hfss', 'uas', 'vas', 'tauu', 'tauv', 'psl', 'hurs', 'albt',
                                'albs', 'rsutcs', 'rsut', 'rlut', 'rlutcs', 'crest', 'crelt', 'crett', 'cress']

atlas_explorer_variables = []
for var in atlas_explorer_variables_list:
    for seas in my_seasons:
        atlas_explorer_variables.append(dict(variable=var, season=seas, table='Amon',
                                              project_specs=dict(
                                                  IGCN_OUT=dict(DIR='ATM'),
                                                  CMIP6CERFACS = dict(table = 'Amon', grid='gr'),
                                              ),
                                              ))
```

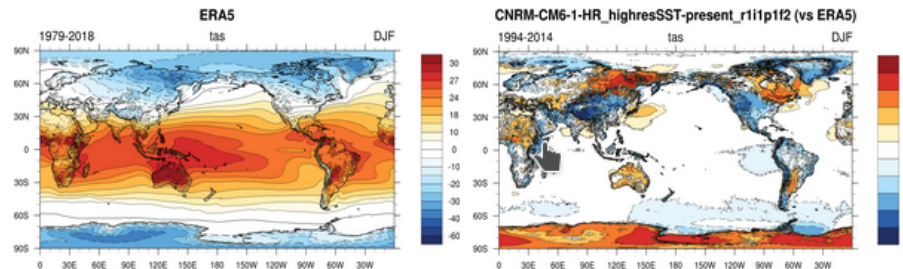
tas versus ERA5 pour highresSST-present

Atmosphere Surface

- 2M Temperature (tas) ; season = ANM ; REF = ERA5



- 2M Temperature (tas) ; season = DJF ; REF = ERA5



- 2M Temperature (tas) ; season = JJA ; REF = ERA5

