

User's and Developer's Guide

October 2018

Deutsches Zentrum für Luft- und Raumfahrt (DLR), Institut für Physik der Atmosphäre, Oberpfaffenhofen,
Germany

<http://www.esmvaltool.org/>

Contents

I	Preface	1
II	Known Issues	5
III	User's Guide	9
IV	Developer's Guide	47
V	Annex A - tables	77
VI	Annex B - environment	87
VII	Annex C - namelists	95

Part I

Preface

About this document

1.1 Foreword

This user's and developer's guide consists of parts targeting two overlapping categories of scientists working with the Earth System Model Evaluation Tool (ESMValTool):

1. Part A: User's Guide (Part [III](#)): this part gives an introduction to the ESMValTool including installation, running the ESMValTool, and available user settings of existing diagnostics and performance metrics. The target group would typically consist of scientists mostly interested in running the ESMValTool as provided either on CMIP model simulations or on simulations performed with other Earth system models, and on observations. An overview on the available diagnostics and metrics packages including a description of the user settings and example plots can be found in Annex C (part [VII](#)).
2. Part B: Developer's Guide (Part [IV](#)): this part gives additional technical details on the ESMValTool not necessarily needed to apply the ESMValTool as well as an introduction to implementing new variables and new diagnostics. This part is mostly intended for scientists interested in technical details as well as in contributing to the development of the ESMValTool by adding new nameslists and code for additional diagnostics or performance metrics.

For the developer's guide, it is assumed that the user/developer is already familiar with the ESMValTool framework introduced in the user's guide.

Please report problems and bugs to the ESMValTool Core Development Team (see [Section 11.2](#) and <http://www.esmvaltool.org/>). Thank you!

1.2 ESMValTool website

The ESMValTool website is <http://www.esmvaltool.org/>

1.3 Main reference

The main reference paper for the ESMValTool is:

Eyring, V., Righi, M., Lauer, A., Evaldsson, M., Wenzel, S., Jones, C., Anav, A., Andrews, O., Cionni, I., Davin, E. L., Deser, C., Ehbrecht, C., Friedlingstein, P., Gleckler, P., Gottschaldt, K.-D., Hagemann, S., Juckes, M., Kindermann, S., Krasting, J., Kunert, D., Levine, R., Loew, A., Mäkelä,

J., Martin, G., Mason, E., Phillips, A. S., Read, S., Rio, C., Roehrig, R., Senfleben, D., Sterl, A., van Uft, L. H., Walton, J., Wang, S., and Williams, K. D.: ESMValTool (v1.0) – a community diagnostic and performance metrics tool for routine evaluation of Earth system models in CMIP, *Geosci. Model Dev.*, 9, 1747-1802, doi: 10.5194/gmd-9-1747-2016, 2016.

Part II

Known Issues

ESMValTool known issues

NCL 6.2.1+

- The function “isfilepresent” has been updated in NCL 6.2.1 in a way that is backwards-incompatible. A new function “fileexists” has been introduced with NCL v6.2.1:

https://www.ncl.ucar.edu/prev_releases.shtml#BackwardsIncompatibleChanges6.2.1

This issue can be addressed by using the ESMVal-function “isfilepresent_esmval” instead of “isfilepresent” and “fileexists”.

NCL 6.3.0

- WAMonsoon: There is a problem with the routine mreg_part_corr (local opt should be removed and undef(“mreg_part_corr”) should be positioned just before.
- A missing “undef” before the declaration of the function “mreg_part_corr” in the NCL library “\$NCARG_ROOT/lib/ncarg/nclscripts/csm/shear_util.ncl” causes the ESMValTool to crash when this library is loaded more than once within the same execution. This unfortunately happens quite often within the tool, since libraries can be loaded multiple times within the various scripts (diag_scripts, lib/ncl, plot_scripts). *At this time, only versions of NCL older than v6.3.0 can be used with the ESMValTool.* This problem will hopefully be fixed with the next NCL release.

Namelist “SouthernHemisphere”

- Models with “dash = 0” in their style definition (diag_scripts/lib/python/style.cfg) will not be plotted in the “fraction occurrence histograms of binned cloud cover” plots created by the diagnostic script diag_scripts/SouthernHemisphere_scatter.py.

Part III

User's Guide

The Earth System Model Evaluation Tool (ESMValTool) is a community-development that aims at improving diagnosing and understanding of the causes and effects of model biases and inter-model spread. The ESMValTool is open to both users and developers encouraging open exchange of diagnostic source code and evaluation results from the Coupled Model Intercomparison Project (CMIP) ensemble. This will facilitate and improve ESM evaluation beyond the state-of-the-art and aims at supporting the activities within CMIP and at individual modelling centers. We envisage running the ESMValTool routinely on the CMIP model output utilizing observations available through the Earth System Grid Federation (ESGF) in standard formats (obs4MIPs) or made available at ESGF nodes.

The goal is to develop a benchmarking and evaluation tool that produces well-established analyses as soon as model output from CMIP simulations becomes available, e.g., at one of the central repositories of the ESGF. This is realized through standard namelists that reproduce a certain set of diagnostics and performance metrics that have demonstrated its importance in benchmarking Earth System Models (ESMs) in a paper or assessment report, such as Chapter 9 of the Intergovernmental Panel on Climate Change (IPCC) Fifth Assessment Report (AR5) (Flato et al., 2013). The expectation is that in this way a routine and systematic evaluation of model results can be made more efficient, thereby enabling scientists to focus on developing more innovative methods of analysis rather than constantly having to “reinvent the wheel”.

In parallel to standardization of model output, the ESGF also hosts observations for Model Intercomparison Projects (obs4MIPs) and reanalyses data (ana4MIPs). obs4MIPs provides open access data sets of satellite data that are comparable in terms of variables, temporal and spatial frequency, and periods to CMIP model output (Taylor et al., 2012). The ESMValTool utilizes these observations and reanalyses from ana4MIPs plus additionally available observations in order to evaluate the models performance. In many diagnostics and metrics, more than one observational data set or meteorological reanalysis is used to assess uncertainties in observations.

3.1 Objectives and approach

The main idea of the ESMValTool is to provide a broad suite of diagnostics which can be performed easily when new model simulations are run. The suite of diagnostics needs to be broad enough to reflect the diversity and complexity of Earth System Models, but must also be robust enough to be run routinely or semi-operationally. In order to address these challenging objectives the ESMValTool is conceived as a framework which allows community contributions to be bound into a coherent framework.

3.2 Architecture

Figure 1 shows a schematic of the ESMValTool architecture: the workflow manager (controlled by the Python script “main.py”) runs a set of diagnostics on data provided by, for instance, a data archive. The configuration and the settings of each diagnostic are specified in namelists read and passed to the diagnostics by the workflow manager. The results which typically comprise of netCDF files and/or plots are stored in output folders along with log-files summarizing the data used, references, and technical details to ensure traceability and reproducibility of the results.

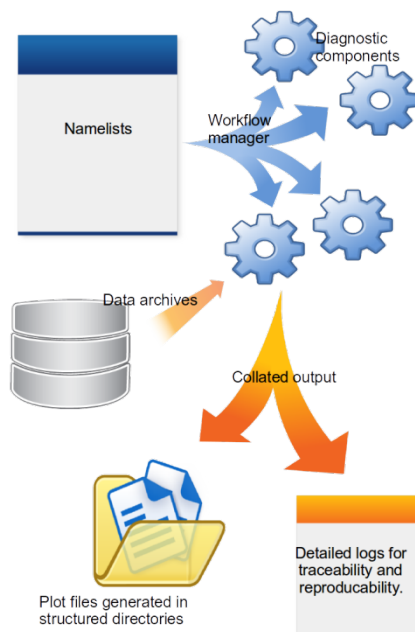


Figure 1: Schematic of the system architecture. The workflow manager (main.py) passes information to the diagnostics; results and log-files are written to dedicated folders.

4.1 Prerequisites

The ESMValTool has the following software requirements (note that specific diagnostics might require additional software packages):

- Unix(-like) operating system
- NCAR Command Language (NCL 2014, <http://www.ncl.ucar.edu/>) version 6.4 or higher (note: NCL version 6.3 is not supported, see known issues, Part II).
- Common GNU utilities such as “wc”, “date”, “basename”, and “more”, which are usually part of the standard Linux distribution.
- The statistical computing software R (<https://www.r-project.org/>) to run diagnostics written in R. A working installation of R and the executable Rscript in the default search path are required. In addition, the netCDF for R libraries (ncdf / ncdf4) are needed. Currently, only the diagnostics “Standardized Precipitation index (SPI)” and “Ozone and associated climate impacts (Eyring13, fig. 6)” (see Part VII) require R. More diagnostics written in R might be added in the future.
- The sea ice diagnostics (and derived diagnostics such as, for instance, the ESA CCI namelist - see Section 25) require the Climate Data Operators (cdo): <https://code.zmaw.de/projects/cdo>. The cdo executable has to be in the default search path (callable via the command “cdo”).
- Python version 2.7.x for running the ESMValTool workflow manager main.py; most diagnostics written in Python require installation of additional Python packages such as, for instance, Geometry Engine (GEOS), scientificpython, netCDF4, cdo, geoval, cartopy, and iris.

The required Python packages can be installed with the following commands:

1. install *anaconda* (2-5.0.1 or later):
 - download anaconda from <https://www.anaconda.com/download/#linux>
 - install anaconda, e.g.

```
chmod 755 Anaconda2-5.0.1-Linux-x86_64.sh
Anaconda2-5.0.1-Linux-x86_64.sh
```

2. install *basemap*

```
conda install basemap
```

3. install netcdf library

```
conda install netcdf4
```

4. install *geoval*

```
git clone https://github.com/pygeo/geoval.git
cd geoval
python setup.py build
python setup.py install
```

create symbolic link for geoval-lib in anaconda directory, e.g.

```
ln -s /home/username/geoval/build/lib.linux-x86_64-2.7/geoval /home/username/
↪anaconda2/lib/
```

5. install *python cdo*

```
conda config --add channels conda-forge
conda install cdo
conda install python-cdo
```

6. it might be needed to replace the cdo executable with more stable version if the version installed in step 5) crashes (e.g. Python diagnostics of namelist “ESA CCI” (Section 25)).

- go to <https://code.mpimet.mpg.de/projects/cdo/> and download executable or source code and compile your own executable
- copy the new cdo executable to your anaconda bin directory, e.g. /home/username/anaconda2/bin/

7. install *cartopy*

```
conda install cartopy
```

8. install *gdal*

```
conda install gdal
```

9. install *iris*

```
conda install iris
```

10. update all conda packages

```
conda update --all
```

Attention: It is strongly recommended to use the Python distribution Anaconda (<https://www.continuum.io/>), as it allows the user to install additional Python libraries and extensions in a simple way and without modifying the installed Python distribution (i.e., without root permissions). The installation instructions for the additional Python packages listed above are given for Anaconda.

4.2 Obtaining the source code

The ESMValTool is available on GitHub at <https://github.com/ESMValGroup/ESMValTool> (for details see Section 16). The ESMValTool is released under the Apache License, version 2.0 and citation of the ESMValTool paper (“Software Documentation Paper”) is kindly requested upon use alongside with the software doi (doi:10.17874/ac8548f0315) and version number:

- Eyring et al., ESMValTool (v1.0) – a community diagnostic and performance metrics tool for routine evaluation of Earth System Models in CMIP, *Geosci. Model Dev.*, 9, 1747-1802, 2016.*

Besides the above citation, users are kindly asked to register any journal articles (or other scientific documents) that use the software at the ESMValTool webpage (<http://www.esmvaltool.org/>). Citing the Software Documentation Paper and registering your paper(s) will serve to document the scientific impact of the Software, which is of vital importance for securing future funding. You should consider this an obligation if you have taken advantage of the ESMValTool, which represents the end product of considerable effort by the development team.

The ESMValTool is developed in a version controlled repository (see Section 16 for details). In addition to using the software, we would therefore like to encourage the community to join the Software Development Team and to contribute additional diagnostics and performance metrics or other software improvements. Contributing back the new diagnostics and performance metrics or other software improvements will help to enhance the capability of the Software, which is of vital importance for securing future funding. You should consider this an obligation if you have taken advantage of the Software, which represents a product of considerable effort by the development team.

Interested developers are welcome to contact the core development team (see Section 11.2).

4.3 Software installation

The ESMValTool can be downloaded from GitHub (for details see Section 16) to any local directory. While the ESMValTool itself does not need to be installed besides downloading/copying the ESMValTool directories to a local folder, it relies on specific software to be available on your system. Please see Section 4.1 for details.

4.4 Verification of the installation

Once you have ESMValTool installed you can verify your installation following the out-of-the-box steps listed below. These tests will let you execute a few simplified namelists that will verify that the dependencies for the general control flow of ESMValTool are in place and working properly. The tests will not verify more specific dependencies used by some Python and R diagnostics, such dependencies will have to be installed separately. Test procedure:

```
1. <INSTALL ESMValTool and dependencies>
2. <cd INTO YOUR INSTALLATION>
3. wget http://goo.gl/ciHCsO -O test-data.tar
4. tar xf test-data.tar
5. wget http://goo.gl/A7pPEz -O test-nml.tar
6. tar xf test-nml.tar
7. ./main.py test-nml/namelist_SAMonsoon-and-WAMonsoon.xml
8. ./main.py test-nml/namelist_SAMonsoon-pr-with-MPI.xml
9. ./main.py test-nml/namelist_SAMonsoon-pr-with-TRMM.xml
10. ./main.py test-nml/namelist_SAMonsoon-pr.xml
```

For each of step 7-10, manually verify that no errors were reported (standard out) and check that diagnostic output figures have been produced in the subfolders “work/plot_*”.

ESMValTool namelists

The ESMValTool namelists are the “control centers” acting as interfaces between the user and the various scripts and configuration files that make up the ESMValTool. A namelist specifies a list of diagnostics to run, global flags and a list of models and observations that are used within the diagnostics. Namelists are text files written in XML (EXtensible Markup Language) [XML]. As a simple text file, the XML-namelist can be easily modified by the user.

For any given namelist “*namelist.xml*”, the ESMValTool is invoked from the command line via (see also Section 8):

```
python main.py nml/namelist.xml
```

The Python “workflow manager” *main.py* will parse the namelist (*namelist.xml*) and call all diagnostic scripts listed in the namelist. This sequence is schematically `python main.py nml/namelist.xml` depicted in [Figure 1](#) and involves the following steps:

1. Parse the namelist
2. Identify the input files on the file system
3. Run an NCL script to check and reformat the input files
4. If needed, run a NCL script to compute derived variables such as, for instance, climate indices
5. Run the diagnostic script (NCL/Python/R/etc.)
6. Repeat previous steps until all diagnostics listed in the namelist are processed

The script *main.py* processes the information in the XML namelist to be used by each of the supported programming languages (currently NCL, Python and R) used for the diagnostic scripts. This means that different diagnostics, even if implemented in different programming languages, can be called within the same namelist. Any changes to the settings of the namelist will be passed to each diagnostic script.

Note that the coupling between the namelist and the diagnostic scripts is “loose”. The Python workflow manager *main.py* passes all information in the namelist to the target diagnostic script, e.g., via intermediate files or environment variables, but it is up to the diagnostic script to act on that information.

Basic structure of a namelist

<GLOBAL>

controls the general settings (see [Table 5.1](#)) ; see Section 5.1, “More on the <GLOBALS>-tag” below for details

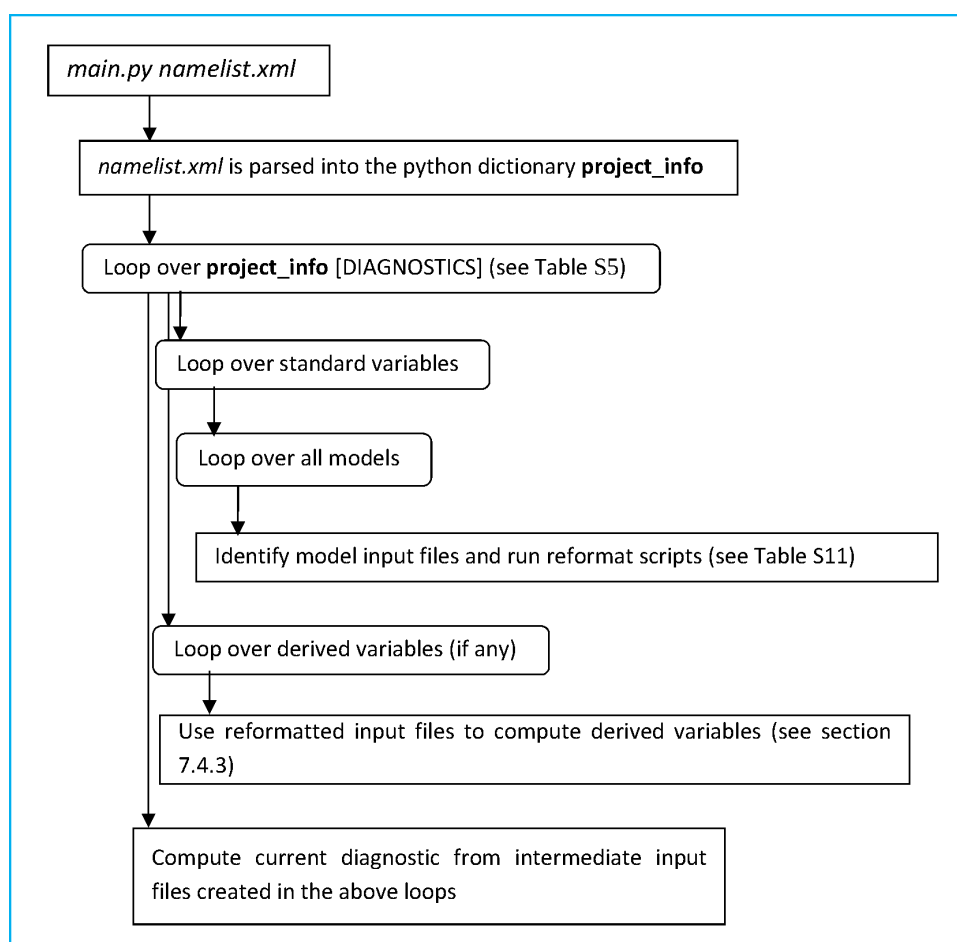


Figure 1: ESMValTool control flow.

</GLOBAL>**<MODELS>**

defines the models/observations and years to be processed and their pathnames; see Section 5.2, “More on the <MODELS>-tag” below for details

</MODELS>**<DIAGNOSTIC>**

defines which diagnostics are run (see Table 5.3); each diagnostic is enclosed in an opening <diag> and closing </diag>-tag; see Section 5.3, “More on the <DIAGNOSTICS>-tag” below for details

</DIAGNOSTIC>

Please note that the “loose coupling” described above applies particularly to the settings defined in the two elements <GLOBAL> and <DIAGNOSTIC>.

5.1 More on the <GLOBAL>-tag

Table 5.1 summarizes the tags defined in the <GLOBAL> section of the namelist. Some of these tags (e.g., `regridding_dir`) are specific to some diagnostics and not generally defined in all namelists.

Table 5.1 Tags of the <GLOBAL> section of the namelist. Note that not all tags might be used by a diagnostic.

Name	Type	Description
<code>climo_dir</code>	string	Path for intermediate files (netCDF)
<code>exit_on_warning</code>	boolean	Stop on warnings
<code>force_calc</code>	boolean	Force diagnostic specific files to be recreated
<code>force_gradecalc</code>	boolean	Force recalculation of model grading (perfmetrics)
<code>force_processing</code>	boolean	Force certain intermediate files (netCDF) to be recreated instead of using cached files
<code>force_taylorcalc</code>	boolean	Force recalculation of data for Taylor plot (perfmetrics)
<code>max_data_blocksize</code>	integer	Currently not used
<code>max_data_filesize</code>	integer	Limits internal memory handling in some core NCL scripts
<code>output_file_type</code>	string	File format of plots (ps, pdf, eps, png); not all formats supported by all diagnostic scripts
<code>plot_dir</code>	string	Output path for plots
<code>read_from_vault</code>	boolean	Retrieve computed diagnostic fields from netCDF
<code>regridding_dir</code>	string	Path for intermediate files used by NCL regridding routines
<code>show_debuginfo</code>	string	Generate a second version of each figure with explanatory text overlaid
<code>tags</code>	string	Comma separated list of tags used for reporting and visualization (see Section 10.3 for details)
<code>verbosity</code>	integer	Verbosity level (0 = minimum output, 4=maximum output)
<code>write_netcdf</code>	boolean	Write results to netCDF file
<code>write_plot_vars</code>	boolean	Currently not used
<code>write_plots</code>	boolean	Produce plots
<code>wrk_dir</code>	string	Output path for data (netCDF, acknowledgements)

5.2 More on the <MODELS>-tag

Each data set is specified by a <model> line with the first entry of each model line being the “project specifier” (see Table 5.2). The project specifier refers to a Python class that is used to parse the model line in the namelist.

For example, a model line with the “CMIP5” specifier looks like:

```
<model> CMIP5 name mip experiment ensemble start-year end-year path </model>
```

- Optionally, the element “mip” can be replaced with “MIP_VAR_DEF” if the tag “MIP” is specified in the <variable> tag (see [Table 5.2](#)), e.g.:

```
<variable MIP="cfDay"> rlut </variable>
```

```
<model> CMIP5_ETHZ MPI-ESM-LR MIP_VAR_DEF amip r1i1p1 1980 1985 @{MODEL-  
PATH}/ETHZ_CMIP5/ </model>
```

- The element “experiment” can be replaced with “EXP_VAR_DEF” if the tag “EXP” is specified in the <variable> tag (see [Table S4](#)), e.g.:

```
<variable MIP="Omon" EXP="esmHistorical"> fgco2 </variable>
```

```
<model> CMIP5_ETHZ NorESM1-ME MIP_VAR_DEF EXP_VAR_DEF r1i1p1 1960 2005  
@{MODELPATH}/ETHZ_CMIP5 </model>
```

The project specifier “CMIP5” will search for files in “path” with filenames matching the pattern

```
_mip_name_experiment_ensemble_
```

Here, the leading asterisk is a placeholder for the variable, which is defined in the <DIAGNOSTICS>-tag (see below), the trailing asterisk is a placeholder for the start/end date of the data set. This naming convention conforms to the syntax used for CMIP5 DRS filenames (as implied by the project specifier name). By implementing their own project specifier classes into the Python code (*interface_scripts/projects.py*), the user can handle data sets that follow different file naming conventions or require additional information to be passed along in addition to the filename. [Table 5.2](#) gives a summary of the available project specifiers and arguments to be used in each <model> line.

[**Note: Examples for the most commonly used project specifiers CMIP5, CMIP5_ETHZ, OBS, and obs4mips as well as downloading instructions and information on the required local directory structure for the model / observational data can be found in [Section :numref:'diag_avail'](#).**]

The <model>-tag may also take the optional attribute “id”:

Example:

```
<model id="ERAINT"> OBS ERA-Interim reanaly 1 2003 2004 @{OBSPATH}/Tier3/ERA-Interim  
</model>
```

The attribute *id* specifies a string that can be used to refer to the model in other places of the namelist. [Table 5.2](#) gives a summary of valid attributes in <model>-tags.

[Table 5.2](#) Project specifiers and corresponding arguments.

project specifier	argument 1	argument 2	argument 3	argument 4	argument 5	argument 6	argument 7	argument 8
ana4mips	Name	table	experiment	ensemble	realm	start year	end year	path
CCMVal CCMVal1 CCMVal2	name name name	case-name case-name case-name	ensemble ensemble ensemble	start year start year start year	end year end year end year	path path path		
CMIP5 CMIP5_ETHZ CMIP5_gridfile CMIP5_SMHI	name name name name	mip mip mip mip	experiment experiment experiment experiment	ensemble ensemble ensemble ensemble	start year start year start year start year	end year end year end year end year	Path path path frequency	grid- file path
ECEARTH	Name	experiment	ensemble	start year	end year	path		
EMAC	name	ensemble	start year	end year	path			
GO GO_gridfile	name name	table table	experiment experiment	ensemble ensemble	start year start year	end year end year	path path	
MiKlip MiK- lip_baseline0	name name	table table	experiment experiment	ensemble ensemble	realm realm	start year start year	end year end year	path path
OBS OBS_gridfile	name name	case-name case-name (insitu, sat, ground reanaly)	ensemble ensemble	start year start year	end year end year	Path path	grid- file	
obs4mips	Name	process level	ensemble	start year	end year	path		

Table 5.2 Optional attributes of the <model> tag.

Name	Type	Description
id	String	Define a name used to refer to the model data in other parts of the namelist

Table 5.2 Optional attributes of the <variable> tag.

Name	Type	Description
exclude	String	Model (id) to exclude from processing
EXP	String	Define a name used to the CMIP5 experiment, e.g., historical
MIP	String	Define a name used to refer to the CMIP5 data stream, e.g., “Amon”, “Omon”, “day”, “fx”; to be used in combination with “MIP_VAR_DEF” replacing the CMIP5 stream in the definition of a <model> tag.
ref_model	String	Define a reference model (model id)

5.3 More on the <DIAGNOSTICS>-tag

Each <diag> entry refers to one or several scripts in the folder *diag_scripts/* complemented by a variable name (see Table 1 for a list of variables) and the corresponding (input) field type (see Table 5.3). Optionally the <diag>-tag may contain additional <model>-tags; these data sets will be processed only by the diagnostic(s) listed in the current <diag> entry. In this way it is possible to define a set of models to be analyzed by all diagnostics in the namelist (in the <MODELS> section) and a set of models to be analyzed only by specific diagnostics (in the <diag> section). Available <diag>-tags are listed in Table 5.3, their optional attributes in Table 5.3.

Table 5.3 Tags of the <diag> section within the <DIAGNOSTICS> section of the namelist. There are no default values.

Name	Type	Description
description	string	1-line description / title of the diagnostic
variable_def_dir	string	Path for the variable-specific configuration file (usually variable_defs)
variable	string	Variable name: a script with the same name (variable_defs/<variable>.ncl) defines the variable to process see Table S8 for a list of variables) including possible pre-processing (e.g., calculating derived variables). Variable scripts should be located in the local folder variable_defs and written in NCL. Even though the variable scripts are written in NCL all meta data defined in the scripts are passed on to the target diagnostic script regardless of the used language (via variable attributes). If multiple variables need to be passed on to a diagnostic script, multiple <variable>-tags have to be defined.
field_type	string	Type of input field (see Table S7) that can be used by the diagnostic scripts. If multiple <variable>-tags are defined a single (which is then applied to all) or an equal number of <field type>-tags has to be defined.
diag_script_cfg_dir	string	Path for diagnostic script configuration file
diag_script	string	Name of diagnostic script; the script can be written in any language currently supported by ESMValTool (NCL, R and Python) and has to be located in the local folder diag_scripts. The settings defined in the diagnostic script configuration file defined by the diag_script cfg attribute is loaded at the beginning of the diagnostic script.
model (optional)	string	Additional data sets specific for this <diag>-section. Data sets defined here will be processed in addition to the ones defined in the MODELS section (see above) but will be ignored by other <diag>-sections.
tags	string	Comma separated list of tags used for reporting and visualization (see Section 10.3 for details)

Table 5.3 Optional attributes of selected tags in the <diag> section.

Name	Type	Parent tag	Description
ref_model	string	<variable>	Defines this data set as the reference data set within the diagnostic. The string ref_model refers to either the model name, as specified in Table S2, or the model attribute id as specified in Table S3. Note that because both model and observational data sets are specified via the <model>-tag any of them can be used as a reference data set.
exclude	string	<variable>	When using more than one variable corresponding to different observational data sets (e.g., precipitation and skin temperature), it is necessary to use this attribute to match which variable goes with which data set, e.g., pr with TRMM and ts with HadISST using, <variable ref_model="trmm" exclude="hadisst"> pr ... <variable ref_model="hadisst" exclude="trmm"> ts ...
cfg	string	<diag_script>	Configuration file for the diagnostic script

Table 5.3 Field types.

Name	Description
T2Ms	Monthly-mean 2d atmosphere or land surface data (longitude, latitude, time:month)
T3M	Monthly-mean 3d atmosphere data (longitude, latitude, pressure, time:month)
T2Mz	Monthly-mean zonal mean 2d atmosphere or land surface data (longitude, pressure, time:month)
T1Ms	Monthly-mean 1d atmosphere or land surface data on a certain pressure level (latitude, time:month)
T2Ds	Daily-mean 2d atmosphere data (longitude, latitude, time:day)
T3D	Daily-mean 3d atmosphere data (longitude, latitude, pressure, time:day)
T2Dz	Daily-mean zonal mean 2d atmosphere data (latitude, pressure, time:month)
T2Is	Daily instantaneous 2d atmosphere data for all years (longitude, latitude, time:day)
T3I	Daily-instantaneous 3d atmosphere data for selected years (longitude, latitude, model level, time:day)
T2Iz	Daily instantaneous zonal mean 2d atmosphere data for all years (latitude, pressure, time:day)
T1Iz	Daily instantaneous 1d field for all years (latitude-pressure, time:day)
T0I	Daily instantaneous 0d field for all years (time:day)
T0As	Annual-mean 0d atmosphere or land surface data on a certain pressure level (latitude, time:year)
F2Ms	Constant 2d land surface data (latitude, longitude)
TO2Ms	Monthly-mean 2d ocean or sea ice data (longitude, latitude, time:month)
TO3M	Monthly-mean 3d ocean or sea ice data (longitude, latitude, model level, time:month)

Table 1 Variable definition scripts.

Script name	Description
abs550aer.ncl	Absorption optical depth (550 nm)
albiscpp.ncl	ISCCP-like cloud albedo
baresoilFrac.ncl	Fraction of bare soil (land cover variable)
burntArea.ncl	Burned area (land cover variable)
chl.ncl	Chlorophyll mass concentration at the surface (ocean)
clcci.ncl	CCI Cloud Area Fraction
clhmtcci.ncl	CCI High Level Medium-Thickness Cloud Area Fraction
clhmtisccp.ncl	ISCCP high level medium-thickness cloud area fraction
clhtkcci.ncl	CCI High Level Thick Cloud Area Fraction
clhtkisccp.ncl	ISCCP High Level Thick Cloud Area Fraction
clhtncci.ncl	CCI High Level Thin Cloud Area Fraction
clhtnisccp.ncl	ISCCP High Level Thin Cloud Area Fraction
clisccp.ncl	ISCCP Cloud Area Fraction
clivi.ncl	Vertically integrated cloud ice
cllmtcci.ncl	CCI Low Level Medium-Thickness Cloud Area Fraction
cllmtisccp.ncl	ISCCP Low Level Medium-Thickness Cloud Area Fraction
clltkcci.ncl	CCI Low Level Thick Cloud Area Fraction
clltkisccp.ncl	ISCCP Low Level Thick Cloud Area Fraction
clltncci.ncl	CCI Low Level Thin Cloud Area Fraction
clltnisccp.ncl	ISCCP Low Level Thin Cloud Area Fraction
clmmtcci.ncl	CCI Middle Level Medium-Thickness Cloud Area Fraction
clmmtisccp.ncl	ISCCP Middle Level Medium-Thickness Cloud Area Fraction
clmtkcci.ncl	CCI Middle Level Thick Cloud Area Fraction
clmtkisccp.ncl	ISCCP Middle Level Thick Cloud Area Fraction
clmtncci.ncl	CCI Middle Level Thin Cloud Area Fraction
clmtnisccp.ncl	ISCCP Middle Level Thin Cloud Area Fraction
cl.ncl	Cloud area fraction (3d)
clt.ncl	Total cloud fraction
cltcci.ncl	CCI Total Cloud Fraction
cltisccp.ncl	ISCCP-like total cloud fraction
cltStderr.ncl	Standard error of total cloud fraction (observations)
clwcci.ncl	CCI Liquid Cloud Area Fraction
clwtcci.ncl	CCI Liquid Cloud Area Fraction

Continued on next page

Table 1 – continued from previous page

Script name	Description
clwvi.ncl	Vertically integrated total cloud water (ice + liquid)
co2flux.ncl	Sum of land and ocean carbon fluxes
conccnd10.ncl	EMAC aerosol variable
conccnd5.ncl	EMAC aerosol variable
conccnd14.ncl	EMAC aerosol variable
conccnmode.ncl	EMAC aerosol variable
conccnSTPd120.ncl	EMAC aerosol variable
conccnSTPd14.ncl	EMAC aerosol variable
conccnSTPd3.ncl	EMAC aerosol variable
conccnSTPd5.ncl	EMAC aerosol variable
conccnSTPmode.ncl	EMAC aerosol variable
cropFrac.ncl	Fraction of crop (land cover variable)
cSoil.ncl	Carbon mass in soil pool
cumnbp.ncl	Cumulated NBP
cVeg.ncl	Carbon mass in vegetation
diamcnmode.ncl	EMAC aerosol variable
dos.ncl	Degree of saturation
dosStderr.ncl	Degree of saturation standard error (observations)
ec532dust.ncl	EMAC aerosol variable
et.ncl	Evapotranspiration
evspsbl.ncl	Evaporation
fgco2.ncl	Surface downward CO2 flux (ocean)
gpp.ncl	Carbon mass flux out of atmosphere due to gross primary production on land
grassFrac.ncl	Fraction of grass (land cover variable)
grassNcropFrac.ncl	Fraction of grass + crop (land cover variable)
hfds.ncl	Downward heat flux at sea surface
hfls.ncl	Surface upward latent heat flux (includes both evaporation and sublimation)
hfss.ncl	Surface upward sensible heat flux
hus.ncl	Specific humidity
huss.ncl	Surface specific humidity
intpp.ncl	Carbon cycle variable
ita.ncl	Depth weighted temperature (ocean, 730 m)
iwpStderr.ncl	Ice water path standard error (observations)
lai.ncl	Leaf area index
LW_CRE.ncl	Longwave cloud radiative forcing
lwp.ncl	Vertically integrated cloud water (liquid only)
lwpStderr.ncl	Vertically integrated cloud water standard error (observations)
m1otst.ncl	Ocean mixed layer thickness
mmraer.ncl	EMAC aerosol variable
mmrbcfree.ncl	EMAC aerosol variable
mmrbc.ncl	BC mass mixing ration
mrro.ncl	Total runoff
mrso.ncl	Soil moisture content
mrsos.ncl	Surface soil moisture content
msftmyz.ncl	Ocean meridional overturning mass streamfunction
MyVar.ncl	Template
nbp.ncl	Carbon mass flux out of atmosphere due to net biospheric production on land
NET_CRE.ncl	Net cloud forcing
o2.ncl	O2 (ocean)
o2_onelev.ncl	O2 (ocean) on a single level
od550aer.ncl	Aerosol optical depth (550 nm)
od550aerStderr.ncl	Aerosol optical depth (550 nm) standard error (observations)
od550lt1aer.ncl	Fine mode aerosol optical depth (550 nm)

Continued on next page

Table 1 – continued from previous page

Script name	Description
od870aer.ncl	Aerosol optical depth (870 nm)
od870aerStderr.ncl	Aerosol optical depth (870 nm) standard error (observations)
pastureFrac.ncl	Fraction pasture (land cover variable)
pctcci.ncl	CCI Mean Cloud Top Pressure
pctisccp.ncl	ISCCP-like cloud top height
phacci.ncl	CCI Cloud Top Phase
prc-mmh.ncl	Convective precipitation in mm per hour
pr-mmday.ncl	Precipitation (total) in mm per day
pr-mmh.ncl	Precipitation (total) in mm per hour
pr.ncl	Precipitation (total)
prStderr.ncl	Precipitation (total) standard error (observations)
prw.ncl	Water vapor path
prwStderr.ncl	Water vapor path standard error (observations)
psl.ncl	Surface pressure
rldscs.ncl	Surface downwelling longwave flux (clear sky)
rlds.ncl	Surface downwelling longwave flux (all sky)
rlns.ncl	Surface Net downward Longwave Radiation
rlus.ncl	Surface upwelling longwave flux
rlutcs.ncl	TOA outgoing clear-sky longwave radiation
rlut.ncl	TOA outgoing all-sky longwave radiation
rsdscs.ncl	Surface downwelling shortwave flux (clear sky)
rsds.ncl	Surface downwelling shortwave flux (all sky)
rsdt.ncl	TOA Incoming Shortwave Radiation
rsns.ncl	Surface Net downward Shortwave Radiation
rsnt.ncl	TOA Net downward Shortwave Radiation
rsuscs.ncl	SWupSFCclr
rsus.ncl	SWupSFC
rsutcs.ncl	TOA outgoing clear-sky shortwave radiation
rsut.ncl	TOA outgoing all-sky shortwave radiation
rtns.ncl	Surface Net downward Total Radiation
rtnt.ncl	TOA Net downward Total Radiation
sconcbc.ncl	BC surface concentration
sconcl.ncl	Cl- surface concentration (aerosol)
sconcn.ncl	Na+ surface concentration (aerosol)
sconcnh4.ncl	NH4 surface concentration
sconcnno3.ncl	NO3 surface concentration
sconcoa.ncl	Organic aerosol (OA) surface concentration
sconcpm10.ncl	PM10 surface concentration
sconcpm2p5.ncl	PM2.5 surface concentration
sconcsso4.ncl	SO4 surface concentration
sfcWind.nc	Near-surface wind speed
sftlf.ncl	Land fraction
shrubFrac.ncl	Fraction shrub (land cover variable)
shrubNtreeFrac.ncl	Fraction shrub and tree (land cover variable)
sic.ncl	Sea ice area fraction
sicStderr.ncl	Sea ice area fraction standard error (observations)
sit.ncl	Sea ice thickness
sm.ncl	Volumetric moisture content of soil layer
smStderr.ncl	Volumetric moisture content of soil layer standard error (observations)
snc.ncl	Fraction of grid cell covered by snow on land
snd.ncl	Surface snow thickness
snw.ncl	Mass of snow on land
so.ncl	Sea water salinity

Continued on next page

Table 1 – continued from previous page

Script name	Description
sos.ncl	Sea surface salinity
spco2.ncl	pCO ₂ (ocean)
stratospheric_column.ncl	Stratospheric ozone column
SW_CRE.ncl	Shortwave cloud radiative forcing
talk.ncl	Total alkalinity (ocean)
ta.ncl	Air temperature
tas.ncl	Near-surface air temperature
tas-degC.ncl	Near-surface air temperature in degrees Centigrade
tauu.ncl	Surface eastward wind stress
tauv.ncl	Surface northward wind stress
tauw.ncl	Surface wind stress
theta-850.ncl	Potential temperature at 850 hPa
theta.ncl	Potential temperature
to.ncl	Sea water temperature
tos.ncl	Sea surface temperature
total_column.ncl	Total ozone column
toz.ncl	Total ozone column (alternative name)
tozStderr.ncl	Total ozone column standard error (observations)
treeFrac.ncl	Fraction tree (land cover variable)
tro3.ncl	Ozone volume mixing ratio
tro3_NHext.ncl	Ozone volume mixing ratio restricted to northern hemisphere extra tropics
tro3prof.ncl	Vertical profile of zonally averaged ozone mixing ratio
tro3_SHext.ncl	Ozone volume mixing ratio restricted to southern hemisphere extra tropics
tro3_Trop.ncl	Ozone volume mixing ratio restricted to tropics
tropospheric_column.ncl	Tropospheric ozone column
tropoz.ncl	Tropospheric ozone column (alternative name)
ts.ncl	Skin temperature
tsStderr.ncl	Skin temperature standard error (observations)
ua-1000.ncl	Wind u-component at 1000 hPa
ua-200-850.ncl	Wind u-component at 200 hPa and at 850 hPa (monsoon diagnostics)
ua-200.ncl	Wind u-component at 200 hPa
ua-700.ncl	Wind u-component at 700 hPa
ua-850.ncl	Wind u-component at 850 hPa
ua-925.ncl	Wind u-component at 925 hPa
ua.ncl	Wind u-component
uo.ncl	Sea water x velocity
va-200-850.ncl	Wind v-component at 200 hPa and at 850 hPa (monsoon diagnostics)
va-200.ncl	Wind v-component at 200 hPa
va-700.ncl	Wind v-component at 700 hPa
va-850.ncl	Wind v-component at 850 hPa
va-925.ncl	Wind v-component at 925 hPa
va.ncl	Wind v-component
vmrc2h4.ncl	EMAC chemistry variable
vmrc2h6.ncl	EMAC chemistry variable
vmrc3h6.ncl	EMAC chemistry variable
vmrc3h8.ncl	EMAC chemistry variable
vmrch3coch3.ncl	EMAC chemistry variable
vmrco_alt.ncl	EMAC chemistry variable
vmrco_azr.ncl	EMAC chemistry variable
vmrco_chr.ncl	EMAC chemistry variable
vmrco_eic.ncl	EMAC chemistry variable
vmrco_gmi.ncl	EMAC chemistry variable
vmrco_hpb.ncl	EMAC chemistry variable

Continued on next page

Table 1 – continued from previous page

Script name	Description
vmrco_lef.ncl	EMAC chemistry variable
vmrco_mlo.ncl	EMAC chemistry variable
vmrco.ncl	CO volume mixing ratio
vmrco_nwr.ncl	EMAC chemistry variable
vmrh2o.ncl	EMAC chemistry variable
vmrnox.ncl	NOx volume mixing ratio
vo.ncl	Sea water y velocity
wfpe-mmday.ncl	Water flux from precipitation and evaporation in mm day-1
wfpe.ncl	Water flux from precipitation and evaporation
xch4.ncl	Column averaged CH4 mixing ratio
xch4Stderr.ncl	Column averaged CH4 mixing ratio standard error (observations)
xco2.ncl	Column averaged CO2 mixing ratio
xco2Stderr.ncl	Column averaged CO2 mixing ratio standard error (observations)
zg.ncl	Geopotential height

Naming convention for ESMValTool namelists:

Typically, all namelists are stored in the folder *nml*, the naming convention is *namelist_XXX.xml* with “xxx” being the name of the diagnostic and/or a description of the purpose of the namelist:

1. **For papers:**

xxx = SurnameYearJournalabbreviation (e.g., stocker12jgr, stocker12sci1, stocker12sci2).

2. **For copies of reports that are not publicly available:**

xxx = OrgYearTitleabbrev (e.g., unep10water, unep11gap, roysoc09geoengineering).

3. **For grouped sets of diagnostics and performance metrics that do not follow a published paper or report:**

xxx = an intuitive name describing the scientific topic (e.g., aerosol, MyDiag, SAMonsoon, SeaIce)

5.4 Namelist configuration file

The user can define base path names in a namelist configuration file and refer to them in the actual namelist file. The configuration file such as, for instance, *config_private.xml* has the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<settings>
  <pathCollection>
    <usrpath category="userDirectory" type="output" id="WORKPATH">
      <path>./work/</path>
      <description>working directory</description>
    </usrpath>
    <usrpath category="userDirectory" type="output" id="PLOTPATH">
      <path>./work/plots/</path>
      <description>directory for output plots</description>
    </usrpath>
    <usrpath category="userDirectory" type="output" id="CLIMOPATH">
      <path>./work/climo/</path>
      <description>directory for output files</description>
    </usrpath>
    <usrpath category="simulation" type="input" id="MODELPATH">
      <path>/path/to/model/data/</path>
      <description>root directory of model data</description>
    </usrpath>
    <usrpath category="observation" type="input" id="OBSPATH">
```

(continues on next page)

(continued from previous page)

```

    <path>/path/to/data/OBS/</path>
    <description>root directory of observational data</description>
  </usrpath>
  <usrpath category="auxiliary" type="input" id="AUXPATH">
    <path>/path/to/data/AUX/</path>
    <description>root directory of auxiliary data</description>
  </usrpath>
</pathCollection>
</settings>

```

A version-controlled template exists (*config_private_template.xml*). The workflow for setting up a repository is to copy *config_private_template.xml* to *config_private.xml* and then to adapt the path names within *config_private.xml* to the user's system structure. *config_private.xml* is ignored by version control (*.gitignore*) and should be backed up elsewhere for reuse.

Inside the namelist file the configuration file can be included in the following way:

```
<include href="config_private.xml"/>
```

and referred to with the syntax:

```
@{id-of-the-usrpath}
```

Note: alternatively, explicitly defined pathnames can be used at any time.

5.5 Standard header for the namelist

For the sake of documentation, standard headers are defined and applied to all namelists and scripts in the ESMValTool. This is a template of the standard header for the main namelist. The parts in red are the ones to be modified by the author.

```

<namelist_summary>
#####
namelist_name.xml

Description
A one-sentence description of the namelist content and purpose.

Author(s)
Name Surname (Affiliation, Country - e-mail@address)

Contributor(s)
Name Surname (Affiliation, Country - e-mail@address)

Project(s)
PROJECT-NAME

Reference(s)
Reference to the paper(s) considered by this namelist (if available).
Author, N. et al., Journ. Abbrev., NN, P1-P2, doi: (YEAR)

This namelist is part of the ESMValTool.
#####
</namelist_summary>

```

5.6 Example namelist

```

<namelist>
<include href="config_private.xml"/>
<namelist_summary>
#####
# namelist_clouds.xml
#
# Description
# Diagnostics of clouds and hydrological cycle.
#
# Author(s)
# Axel Lauer (DLR, Germany - axel.lauer at dlr.de)
#
# Contributor(s)
#
# Project(s)
# EMBRACE
#
# Reference(s)
#
# This namelist is part of the ESMValTool.
#####
</namelist_summary>

<GLOBAL>
  <write_plots type="boolean">          True          </write_plots>
  <write_netcdf type="boolean">         True          </write_netcdf>
  <force_processing type="boolean">     False         </force_processing>
  <wrk_dir type="path">                 work/         </wrk_dir>
  <plot_dir type="path">               work/plots/    </plot_dir>
  <climo_dir type="path">              work/climo/    </climo_dir>
  <max_data_filesize type="integer">    100          </max_data_filesize>
  <verbosity type="integer">           1            </verbosity>
  <exit_on_warning type="boolean">     False         </exit_on_warning>
  <output_file_type>                  ps            </output_file_type>
</GLOBAL>

<MODELS>
  <model> CMIP5_ETHZ CESM1-CAM5  Amon  historical  rlilp1  2000 2004 @
  ↳{MODELPATH}/ETHZ_CMIP5/ </model>
  <model> CMIP5_ETHZ GFDL-ESM2G  Amon  historical  rlilp1  2000 2004 @
  ↳{MODELPATH}/ETHZ_CMIP5/ </model>
  <model> CMIP5_ETHZ MIROC5     Amon  historical  rlilp1  2000 2004 @
  ↳{MODELPATH}/ETHZ_CMIP5/ </model>
  <model> CMIP5_ETHZ MPI-ESM-MR  Amon  historical  rlilp1  2000 2004 @
  ↳{MODELPATH}/ETHZ_CMIP5/ </model>
  <model> CMIP5_ETHZ NorESM1-M   Amon  historical  rlilp1  2000 2004 @
  ↳{MODELPATH}/ETHZ_CMIP5/ </model>
</MODELS>

<!--
      This is an example of a comment in XML
-->

<!-- Please do not change anything below this line,
      unless you want to modify the standard diagnostic settings. -->
<DIAGNOSTICS>
  <diag>
    <description> Cloud diagnostics
    </description>

```

(continues on next page)

(continued from previous page)

```
<variable_def_dir>      ./variable_defs/      </variable_def_dir>
<variable>              lwp                   </variable>
<field_type>            T2Ms                  </field_type>
<diag_script_cfg_dir>   ./nml/cfg_clouds/      </diag_script_cfg_dir>
<model> OBS UWisc sat v2 1988 2007 @{OBSPATH}/UWisc </model>
<diag_script cfg="cfg_clouds.ncl"> clouds.ncl  </diag_script>
</diag>
</DIAGNOSTICS>

</namelist>
```

Directory structure of the ESMValTool

An overview of the directory structure used in the ESMValTool is given in [Table 14](#). This section summarizes the underlying principles of the structure.

- Common namelist settings (e.g., models, year ranges, diagnostics) are usually stored in one place.
- Less common settings may be hidden deeper in the directory structure (see also [Section 7](#)).
- Diagnostic scripts that can be used by namelist entries are also stored in one place and it generally possible to combine them in a modular way (e.g., using the output of one routine as input for another).
- Reuse of code is strongly encouraged, i.e., one place for each functionality (modularity on the technical level).
- The goal is to centralize functionality in individual functions/procedures whenever there is the possibility of reusability. New developers are encouraged to consider building on or extending existing routines before introducing new ones.
- Routines are sorted into folders according to their functionality. Whenever possible, the hierarchy level of routines is reflected by their position in the directory structure.

Configuration files

7.1 nml/cfg_diag/cfg_diag*.typ

Diagnostic-specific settings are passed via configuration files. These are collected in the `nml` directory under subdirectories named like the corresponding diagnostic (e.g., `cfg_aerosol/`, `cfg_perfmetrics/`). The suffix “`.typ`” specifies the language the routine is written in.

There might be more than one configuration script per diagnostic set. All `cfg_*` files for a diagnostic set need to be in the same folder specified by the `<diag_script_cfg_dir>` entry of the namelist (`nml/namelist_*.xml`) (see Section 5.3 and Table 5.3).

The configuration settings are specified as attributes of the variable “`diag_script_info`” (in NCL via “`diag_script_info@attribute = ...`”, see example below). In order to activate these attributes, “`diag_script_info`” must be set to “`True`”.

Example (NCL):

```
diag_script_info = True

diag_script_info@projection = "CylindricalEquidistant" ; map projection,
                                     ; e.g. Mollweide, Mercator

diag_script_info@styleset = "CMIP5" ; "CMIP5", "DEFAULT"
diag_script_info@colormap = "WhiteBlueGreenYellowRed" ; e.g., ↪
↪WhiteBlueGreenYellowRed,
                                     ; rainbow
diag_script_info@ncdf = "default" ; enable to output to netCDF; either use "default
↪"
                                     ; or give a full file name
```

Example (Python):

```
class diag_script:
    def __init__(self):
        self.info = True
        self.color = 1
        self.factor = 2.0e-3
        self.name = "example"

diag_script_info = diag_script()
```

Example (R):

```
diag_script_info<-new()
diag_script_info[["begin_ref_year"]]<-1970
diag_script_info[["end_ref_year"]]<-2000
diag_script_info[["timescale"]]<-3
diag_script_info[["seasons"]]<-c("ann", "djf", "mam", "jja", "son")
```

Running the ESMValTool

The following section gives a brief description of the steps required by a user to run an existing diagnostic. As an example, the toy diagnostic *MyDiag* is chosen to illustrate the basic steps:

1. Copy the template for the “namelist configuration file” (*config_private_template.xml*) to *config_private.xml* and adapt the path names within *config_private.xml* to your system structure (see Section 5.4). If needed, set/change the base path names for the input data (model and observations) and the output data. *config_private.xml* is ignored by version control (*.gitignore*) and should be backed up elsewhere for reuse. Note: the use of a “namelist configuration file” is optional in order to allow for machine specific standard search paths for input (and output) data without having to change the actual namelists. Alternatively, explicit path names can be used in the namelists.
2. Check/edit the main namelist *nml/namelist_MyDiag.xml*:
 - (a) Set/check the file name of the “namelist configuration file” (defining the base path names) to be used by the ESMValTool (typically, this is the second line in the namelist, e.g., `<include href=“./config_private.xml”/>`).
 - (b) If needed, set the pathnames in the <GLOBAL> section for the “work” directory (*wrk_dir*), the directory for the plots (*plot_dir*) and the directory for reformatted files (*climo_dir*). See Section 5.1 for details and Table 5.1 for a complete list of variables in the <GLOBAL> section.
 - (c) In the <MODELS> section, define the model(s) to be used, including the root path for the actual model data, e.g.,


```
CMIP5_ETHZ MPI-ESM-LR Amon historical r1i1p1 2000 2004 @{MODELPATH}/ETHZ_CMIP5/
```

 (see step 1 and Section 5.4 for details on how to set base path names such as `@{MODELPATH}`; alternatively, explicit path names can be used). See Section 5.2, Table 5.2 and Table 5.2 for details. The first year (here: 2000) and last year (here: 2004) of the model data processed for each model is specified in this section.
 - (d) Optionally, change variable and the field type in the <DIAGNOSTICS> section. See Section 5.3, Table 5.3 and Table 5.3 for details. An overview of the available “field types” is given in Table 5.3, Table 1 lists the available variables. Please note that the diagnostic section may include additional models and/or observational data.
3. Check/edit the configuration file *nml/cfg_MyDiag/cfg_MyDiag.ncl*. In case of the toy diagnostics *MyDiag*, you can for example change the map projection for the contour plot by changing the value of the attribute `diag_script_info@projection`.
4. Run the ESMValTool (in the ESMValTool root directory): `python main.py nml/namelist_MyDiag.nml`

5. The output will be written to a subdirectory named like the diagnostics package (e.g., *MyDiag*) in the directories specified in the <GLOBAL> section of the namelist (see step 1 and also Section 5.4). The default directories are: *work/MyDiag* for the NetCDF output and *work/plots/MyDiag* for the plot(s) (see also Figure 1). Acknowledgements and references are written to the file *work/refs-acknows_MyDiag.txt*.

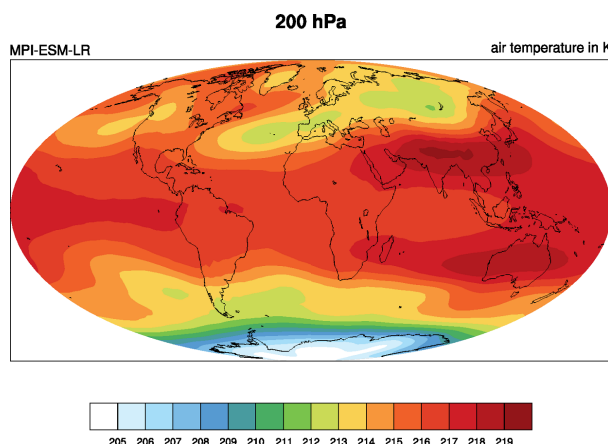


Figure 1: Example plot created by the toy diagnostic MyDiag showing the 5-year annual mean temperature at 200 hPa from the CMIP5 historical run (r1i1p1) with the MPI-ESM-LR model.

8.1 Available diagnostics and metrics

An introduction to the available diagnostics and metrics packages implemented into the ESMValTool v1.1 including a description of the user settings, observational data used, references, and example plots is given in Part VII.

8.2 Model and observational data

8.2.1 Model data

The project specifier (see Table 5.2) used in the <MODELS> section of the namelist (see Section 5.2 for details) determines the directory structure and file naming convention expected by the ESMValTool. The two most commonly used project specifiers are *CMIP5* and *CMIP5_ETHZ*. Both are used to process CMIP5 data available from the Earth System Federation Grid (e.g., <http://esgf.llnl.gov/>). In order to download CMIP5 data, registration and creation of an “openID” is required. Instructions can be found here: http://cmip-pcmdi.llnl.gov/cmip5/data_getting_started.html.

Besides downloading files individually, the CMIP5 data portal is capable of generation a script for automated download of model data using GNU *wget* (<https://www.gnu.org/software/wget/>). The *wget* script generation is recommended for downloading a large number files and/or large data volume.

Any CMIP5 files downloaded to a single directory can be moved to a CMIP5 like directory structure using the NCL script *util/CMIP5_sort/CMIP5_sort.ncl*. The script *CMIP5_sort.ncl* expects all files to be moved to the CMIP5 like directory structure in the current directory (“.”). The root directory for creating the CMIP5 like directory structure is specified in the script *CMIP5_sort.ncl* via the variable “*outpath*”. The script is run via:

```
ncl <path of the ESMValTool>/util/CMIP5_sort/CMIP5_sort.ncl
```

The CMIP5 files will be moved into the directories *outpath/experiment/mip/variable/name/ensemble/* for direct usage with the project specifier *CMIP5_ETHZ* (see below). “*experiment*”, “*mip*”, “*variable*” and “*name*” (= model name) are automatically extracted from the filename.

The project specifier CMIP5

Syntax of the *CMIP5* specifier in the <model>-tag (see Section 5.2 and Table 5.2 for details):

<model> CMIP5 name mip experiment ensemble start_year end_year path </model>

The project specifier CMIP5 will search for files in “**path**” with filenames matching the pattern

*variable_mip_name_experiment_ensemble_.nc**

Note: “variable” is specified in the <diag>-section (see Section 5.3 for details). If “variable” is a derived quantity, all variables needed to calculate the derived quantity are processed automatically.

The project specifier CMIP5_ETHZ

Syntax of the *CMIP5_ETHZ* specifier in the <model>-tag (see Section 5.2 and Table 5.2 for details):

<model> CMIP5_ETHZ name mip experiment ensemble start_year end_year path </model>

The project specifier *CMIP5_ETHZ* will search for files in “**path/experiment/mip/variable/name/ensemble**” with filenames matching the pattern

variable_mip_name_experiment_ensemble_.nc*

This directory structure resembles the ESGF CMIP5 directory structure.

Note: “variable” is specified in the <diag>-section (see Section 5.3 for details). If “variable” is a derived quantity, all variables needed to calculate the derived quantity are processed automatically.

8.2.2 Observational / reanalysis data

When possible, observations from the obs4MIPs/ana4MIPs archives are used in the model evaluation. These data are freely available from the ESGF in the same format as the CMIP simulations and can be directly used in the ESMValTool using the **obs4mips** or **ana4mips** project specifiers (see Table 5.2) in the namelist.

A collection of all observational data used by the diagnostics of the ESMValTool (*MASTER BRANCH*) is hosted at DLR and can be made available (restrictions by the data owner permitting) on request (see Table 1). The reformatted observational data can be read using e.g., the **OBS** class in the namelist (see below).

All observations are tiered as follows:

- Tier 1: data sets from the obs4MIPs and ana4MIPs archives
- Tier 2: other freely available data sets
- Tier 3: restricted data sets (e.g., license agreement required)

Observational data sets not available in the obs4MIPs/ana4MIPs archives need to be reformatted according to the CF/CMOR standard before they can be used (see Section 8.2.3 for more details).

The project specifier OBS

Syntax of the *OBS* specifier in the <model>-tag (see Section 5.2 and Table 5.2 for details):

<model> OBS name case_name ensemble start_year end_year path </model>

The project specifier OBS will search for files in “**path**” with filenames matching the pattern

OBS_name_casename_ensemble_fieldtype_variable.nc*

Note: “variable” and “fieldtype” are specified in the <diag>-section (see Section 5.3 for details). If “variable” is a derived quantity, all variables needed to calculate the derived quantity are processed automatically.

The project specifier obs4mips

Syntax of the *obs4mips* specifier in the <model>-tag (see Section 5.2 and Table 5.2 for details):

<model> obs4mips name process_level ensemble start_year end_year path </model>

The project specifier obs4mips will search for files in “**path/name/**” with filenames matching the pattern

variable_name_processlevel_ensemble_.nc*

Note: “variable” is specified in the <diag>-section (see Section 5.3 for details). If “variable” is a derived quantity, all variables needed to calculate the derived quantity are processed automatically.

8.2.3 Downloading and creating observational data sets

obs4MIPS and ana4MIPs data sets ("tier 1", see above) are freely available from the ESGF. These data sets can be used directly with the ESMValTool without the need for reformatting. Examples of such data sets include:

- AIRS
- CERES-EBAF
- CFSR
- CloudSat-L3
- GPCP-1DD, GPCP-SG
- IFS-Cy31r2
- ISCCP
- MERRA
- MISR
- MODIS
- TES
- TRMM, TRMM-L3

For the required file naming conventions and the expected directory structure see Section :num-ref:'obs_data'

For all other (non-obs4MIPs and non-ana4MIPs) data sets, reformatting routines are provided with downloading and processing instructions in the header of the scripts. These reformatting scripts can be found in:

reformat_scripts/obs/

These reformat scripts can be specified in a namelist-file (e.g. *namelist_reformat_obs.xml*) and executed by calling the main.py script with the option "-r":

```
python main.py -r namelist_reformat_obs.xml
```

This reformat namelist file contains the tag <REFORMAT> that can hold multiple <reformat_script>-tags specifying the reformat scripts to be called:

```
<REFORMAT>
<reformat_script> /PATH/TO/REFORMATSCRIPT</reformat_script>
</REFORMAT>
```

An example reformat namelist file is listed below:

```
<namelist>
<include href="./config_private.xml"/>
<namelist_summary>
#####
namelist_reformat_obs.xml

Description
Special namelist for reformatting observational data.
The currently available reformat scripts are stored in reformat_scripts/obs/
To run this namelist the -r option must be given:
    python main.py -r nml/namelist_reformat_obs.xml

This namelist is part of the ESMValTool
#####
</namelist_summary>

<REFORMAT>
```

(continues on next page)

(continued from previous page)

```

<reformat_script id=obs_1>./reformat_scripts/obs/reformat_obs_1.ncl </reformat_
↪script>
<reformat_script id=obs_2>./reformat_scripts/obs/reformat_obs_2.ncl </reformat_
↪script>
<reformat_script id=obs_3>./reformat_scripts/obs/reformat_obs_3.ncl </reformat_
↪script>

<reformat_script id=obs_N>./reformat_scripts/obs/reformat_obs_N.ncl </reformat_
↪script>
</REFORMAT>

</namelist>

```

A list of available data sets and their corresponding reformatting routines are given in [Table 1](#).

Table 1 Observational data for use with the ESMValTool. See headers of the reformatting routines for downloading and processing instructions.

Name	Tier	Description	Variables	Type	Time range	Script name
ACCESS	3	Aerosol vertical profiles	mmrbc	Campaign	–	reformat_obs_ACCESS.ncl
ACCESS-2	3	Aerosol vertical profiles	conccnd5, conccnd10	Campaign	2014-2014	reformat_obs_ACCESS-2.ncl
AERONET	2	Aerosol optical depth at 550nm	od550aer	Ground	1992-2012	reformat_obs_AERONET.ncl
AIRS	1	relative humidity, temperature	hur, hus, ta	Satellite	2003-2010	none (obs4MIPS)
Asmi11	2	Aerosol size distributions	sizecnSTP	Ground	2009-2010	reformat_obs_Asmi11.ncl
AURA-MLS-OMI	2	Tropospheric column ozone	tropoz	Satellite	2005-2013	reformat_obs_AURA-MLS-OMI.ncl
AURA-TES	2	Ozone mixing ration	vmro3	Satellite	2005-2009	reformat_obs_AURA-TES.ncl
BDBP	3	zonally averaged ozone profiles	tro3prof	Ozone sondes	1979-2007	reformat_obs_BDBP.ncl
CARSNET	2	Aerosol optical depth at 550 nm	od550aer	Ground	2002-2013	reformat_obs_CARSNET.ncl
CASTNET	2	Aerosol surface level concentration	concso4, concso3, concnh4	Ground	S1987-2012	reformat_obs_CASTNET.ncl
CERES	3	CERES syn-optic data (radiative fluxes at surface, toa)	rsuscs, rsus, rsdscs, rsds, rluscs, rlus, rldscs, rlds, rsutcs, rsut, rlutcs, rlut	Satellite	2004	reformat_obs_CERES-SYN1deg-SFC.bash, reformat_obs_CERES-SYN1deg-TOA.bash
CFSR	1	Surface pressure	psl	Reanalysis	2013	none (obs4MIPs)
CIRRUS	3	Aerosol vertical profiles	mmrbc, mmrbcfree	Campaign	late Nov. 2006	reformat_obs_CIRRUS.ncl
CLARA-A2	2	Cloud cover	clt	Satellite	1982-2015	contact ESMValtool development team
CloudSat	1	Cloud cover	clt	Satellite	2006-2010	reformat_obs_cloudsat.bash

Continued on next page

Table 1 – continued from previous page

Name	Tier	Description	Variables	Type	Time range	Script name
CMAP	2	Precipitation	pr	merged analysis	1980-2013	reformat_obs_CMAP.ncl
Concert	3	Aerosol vertical profiles	mmrbc, concn-STP14	Campaign	–	reformat_obs_CONCERT.ncl
CR-AVE	3	Aerosol vertical profiles	mmrbc	Campaign	–	reformat_obs_CR-AVE.ncl
CRU	3	Surface temperature, precipitation	tas, pr	Reanalysis	1901-2006	reformat_obs_CRU.ncl
DC3	3	Aerosol vertical profiles	mmrbc	Campaign	–	reformat_obs_DC3.ncl
Dong08-ARGO	2	Derived ocean mixed layer depth	mlost	Campaign	2001-2006	reformat_obs_Dong08-ARGO-monthly.ncl
EANET	3	Aerosol surface level concentrations	concso4, consco3, concnh4	Ground	2001-2005	reformat_obs_EANET.ncl
EMEP	2	Aerosol surface level concentration	concso4, concno3, concnh4, concnh4, concpm2p5, concpm10	Ground	1970-2012	reformat_obs_EMEP.csh
Emmons	2	Vertical profiles of gases	various	Campaign	variable	reformat_obs_Emons.csh
ERA-40	3	essential climate variables	ta, ua	Reanalysis	1960-2001	
ERA-Interim	3	Basic climate parameters	ta, ua, va, zg, hus, tas, tos, ps, psl, tauu, tauv, clwvi, clivi, sftif	Reanalysis	1979-2012	reformat_obs_ERA-Interim.ncl, reformat_obs_ERA-Interim-surffluxes.ncl
ERA-Interim fluxes	3	Basic climate parameters, surface fluxes	pr, evspsbl, hfls, hfss, rsns, rlms	Forecast	2000-2005	reformat_obs_ERA-Interim-surffluxes.ncl
ESACCI-AEROSOL	2	Aerosol optical depth at 550 nm	od550aer, od870aer, od550lt1aer, abs550aer, od550aer-Stderr, od870aer-Stderr	Satellite	1997-2011	reformat_obs_ESACCI-AEROSOL.ncl
ESACCI-CLOUD	2	Total cloud fraction, Liquid water path, Ice water path	clt, clwvi, clivi, lwpStderr, iwpStderr, cltStderr	Satellite	2007-2009	reformat_obs_ESACCI-CLOUD.ncl

Continued on next page

Table 1 – continued from previous page

Name	Tier	Description	Variables	Type	Time range	Script name
ESACCI-GHG	2	column averaged CO ₂ and CH ₄	xco2, xco2Stderr, xch4, xch4Stderr	Satellite	2003-2014	reformat_obs_ESACCI-GHG.ncl
ESACCI-OZONE	2	Total ozone column, Tropospheric column ozone, Ozone mixing ratio	toz, tro3prof, tozStderr, tro3Stderr	Satellite	2007-2008	reformat_obs_ESACCI-OZONE.ncl, reformat_obs_ESACCI-OZONE-PL.ncl
ESACCI-SIC	2	Sea ice concentration	sic, sicStderr	Satellite	2003-2010	reformat_obs_ESACCI-SIC.ncl
ESACCI-SOIL-MOISTURE	2	Degree of saturation	dos, dosStderr, sm, smStderr	Satellite	1988-2008	reformat_obs_ESACCI-SOILMOISTURE.ncl
ESACCI-SST	2	Sea surface temperature (saved as skin temperature)	ts, tsStderr	Satellite/Analysis	1992-2010	reformat_obs_ESACCI-SST.ncl
ESRL	2	CO ₂ surface level concentrations	co2	Ground	1973-2012	reformat_obs_ESRL.ncl
ETH-SOM-FFN	2	pCO ₂ ocean surface	spco2	–	1998-2011	reformat_obs_ETH-SOM-FFN.ncl
GCP	2	CO ₂ exchange	co2flux, fgco2, nbp	Reanalysis	1959-2011	reformat_obs_GCP.ncl
GLOBAL-VIEW	2	CO surface level concentrations	vmrco	Ground	1991-2008	reformat_obs_GLOBAL-VIEW.ncl
GPCC	2	Precipitation	pr	Reanalysis	1901-2010	reformat_obs_GPCC.ncl
GPCP	1	Precipitation	pr, prStderr	–	1979-2013	none (obs4MIPs)
GTO-ECV	3	Total column ozone	toz	Satellite	1996-2010	reformat_obs_GTO-ECV.ncl
HadCRUT	2	Near-surface air temperature	tas	Ground	1850-2013	reformat_obs_HadCRUT.ncl reformat_obs_HadCRUT4.ncl
HadISST	2	Sea ice concentrations and sea surface temperatures	sic, ts	Reanalysis	1870-2014	reformat_obs_HadISST.ncl
HALOE	2	Water vapour mixing ratio	vmrh2o	Satellite	1991-2002	reformat_obs_HALOE.ncl
HIPPO	3	Aerosol vertical profiles	mmrbc	Campaign	–	reformat_obs_HIPPO.ncl
HWSD	2	Soil carbon content	cSoil	Ground	2000	
IFS-Cy31r2	1	Surface pressure	psl	Reanalysis	1979-2013	none (obs4MIPs)

Continued on next page

Table 1 – continued from previous page

Name	Tier	Description	Variables	Type	Time range	Script name
IMPROVE	2	Aerosol surface level concentrations	concs04, concno3, concnh4, concbc, concoa, concpm2p5, concpm10	Ground	1988-2011	reformat_obs_IMPROVE.ncl
INCA	3	Aerosol vertical profiles	conccnSTP5, conccnSTP14, conccnSTP120	Campaign	–	reformat_obs_INCA.ncl
ISCCP	1	Cloud properties	albiscpp, cliscpp, cltiscpp, ctiscpp	Satellite	1984-2007	none (obs4MPIs)
ISCCP-FD-SRF	2	Clear-sky radiative fluxes	rsdscs, rsuscs	Satellite	1984-2009	
JMA-TRANSCOM	3	CO ₂ exchange	nbp, fgco2	Reanalysis	1985-2008	
LACE	2	Aerosol size distributions	sizecn	Campaign	–	reformat_obs_LACE.ncl
LAI3g	3	Leaf area index	LAI	Reanalysis	1982-2010	
LandFlux-EVAL	3	Evapotranspiration	et, et-sd	Synthesis product (model + observations)	1989-2005	reformat_obs_landflux-eval.ncl
MERRA	1	Precipitation	pr	Reanalysis	1979-2011	none (obs4MPIs)
MISR	1	Aerosol optical depth	od550aer	Satellite	2001-2012	none (obs4MPIs)
MLS	1	humidity, temperature	hus, husStderr, ta, taStderr	Satellite	2005-2010	none (obs4MPIs)
MODIS-CFMIP	2	Ice water path	clivi	Satellite	2003-2014	reformat_obs_MODIS-CFMIP.ncl
MODIS_L3_C6	2	Ice water path, liquid water path, total cloud cover, aerosol optical depth	clivi, clwvi, clt, od550aer	Satellite	2003-2014	reformat_obs_MODIS-L3-C6.ncl
MTE	2	Gross primary productivity of carbon	gpp	Reanalysis	1982-2008	
NCEP	2	Essential climate variables	ta, ua, va, zg, hus, tas	Reanalysis	1948-2012	reformat_obs_NCEP.ncl, reformat_obs_NCEP-daily.ncl
NDP	2	Vegetation carbon content	cVeg	Ground	2000	
NIWA	3	Total column ozone	toz	Reanalysis	1980-2010	reformat_obs_NIWA.ncl

Continued on next page

Table 1 – continued from previous page

Name	Tier	Description	Variables	Type	Time range	Script name
NOAA interpolated OLD	2	Interpolated outgoing long-wave radiation	rlut	Satellite	1975-2013	reformat_obs_NOAA-PSD-Interp.ncl
NSIDC	2	Sea ice concentrations	sic	Satellite	1978-2010	reformat_obs_NSIDC.ncl
PATMOS	2	Cloud cover	clt	Satellite	1982-2014	<i>contact ESMValtool development team</i>
Putaud	2	Aerosol size distributions	sizecn	Campaign	–	reformat_obs_Putaud.ncl
SALTRACE	3	Aerosol vertical profiles	mmrbc	Campaign	–	reformat_obs_SALT-RACE.ncl
SeaWIFS	2	Ocean biochemistry	chl		1997-2010	reformat_obs_SeaWIFS.csh
SOCAT	2	Ocean surface CO ₂	spco2		1970-2011	reformat_obs_SOCAT.csh
SRB	2	Radiative fluxes	rsut, rlut, rlutcs	Satellite	1983-2007	reformat_obs_SRB.ncl
SSMIMERIS	1	Water vapour path	prw, prw-Stderr	Satellite	2003-2008	none (obs4MIPs)
takahashi14	2	Ocean biogeochemistry	talk		2005	reformat_obs_takahashi14.csh
TC4	3	Aerosol vertical profiles	mmrbc	Campaign	–	reformat_obs_TC4.ncl
TES	1	Ozone	tro3		2006-2009	reformat_obs_TES.ncl
Texas	3	Aerosol vertical profiles	mmrbc, mmraer	Campaign	–	reformat_obs_Texas.ncl
Tilmes	2	Ozone mixing ratios	vmro3	in-situ	1995-2009	reformat_obs_Tilmes.ncl
TOMS	2	Total ozone column	toz	Satellite	1990	
TRMM-3B42	2	Precipitation	pr	Satellite	1998-2014	reformat_obs_TRMM-3B42-daily.ncl, reformat_obs_TRMM-3B42-3hourly.ncl
UCN-Pacific	3	Aerosol vertical profiles	conccnSTP3	Campaign	–	reformat_obs_UCN-Pacific.ncl
UWisc	3	Liquid water path	clwvi, lwp-Stderr	Satellite	1988-2007	reformat_obs_UWisc.ncl
WHOI-OAFlux	2	Global ocean heat flux and evaporation	hfss, hfss	Analysis	1958-2013	reformat_obs_WHOI-OAFlux.ncl
WOA09	2	Climatological ocean fields	so, sos, to, tos	Analyzed climatology	–	reformat_obs_WOA09.ncl
woa2005	2	Ocean biogeochemistry	o2		2005	reformat_obs_woa2005.csh

8.3 The acknowledgements log file

Each diagnostics in the tool automatically generates a log file containing a list of authors/contributors, details on the projects to be acknowledged and the reference papers to be cited. It also provides a list of the used model and observational data with the corresponding references.

The log is created automatically when running the ESMValTool. The log file is named *refs-*

acknow_<diagnostics>.txt and written to the directory defined in the <GLOBAL> section of the namelist (variable *wrk_dir*, see Section 5.1), e.g., *work/refs-acknows_MyDiag.txt* (see also Section 8, step 4).

An example excerpt of an acknowledgements log file is provided below.

Example

```
-----
+++++++ ESMValTool REFERENCES and ACKNOWLEDGEMENTS LOG ++++++
-----

Namelist file: namelist_perfmetrics_CMIP5.xml
Creation date: Wed Dec 16 22:58:29 CET 2016
ESMValTool version: 1.1.0
Host name: ###
User name: ###

Licensed under the Apache License, Version 2.0 (the "License"); you may
not use this file except in compliance with the License. You may obtain
a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Please acknowledge the use of the ESMValTool.
Please cite Eyring et al., ESMValTool (v1.0) -- a community diagnostic and
performance metrics tool for routine evaluation of Earth System Models in
CMIP, Geosci. Model Dev., 2016.
For the specific diagnostics, see below.

=====
=== perfmetrics_main.ncl ===

AUTHOR(S):
-A- ###

CONTRIBUTOR(S):
-C- ###
-C- ###
-C- ###

REFERENCE(S) FOR THIS DIAGNOSTIC:
-R- Please cite Righi et al., Geosci. Model Dev., 8, 733-768
doi:10.5194/gmd-8-733-2015, 2015.
-R- Please cite Gleckler et al., J. Geophys. Res., 113, D06104,
doi:10.1029/2007JD008972, 2008.

REFERENCE(S) FOR THE OBSERVATIONS:
-R- NCEP - Kalnay et al., Bull. Amer. Meteor. Soc., 77, 437-470, 1996.
-R- ERA-Interim
-R- AIRS
-R- CERES-EBAF
-R- SRB

ACKNOWLEDGEMENTS FOR THE PROJECTS:
-P- EU FP7 project EMBRACE
-P- DLR project ESMVal
```

(continues on next page)

(continued from previous page)

PREPROCESSING/REFORMATTING (ESMValTool v1.1.0):

Variable: ta

Model: ERA-Interim

Input file(s):

(1) OBS_reanaly_ERA-Interim_1_T3M_ta_2000-2001.nc

Original source file(s) of all input file(s):

-S- (1)

\@{\OBSPATH}/Tier3/ERA-Interim/OBS_ERA-Interim_reanaly_1_T3M_ta.nc

Fixes applied to original source file(s): none

Reference(s) of original source file(s):

(1) Dee, D. P. et al., Q. J. Roy. Meteor. Soc., 137, 553-597,

doi:10.1002/qj.828, 2011.

Model: MPI-ESM-LR

Input file(s):

(1) CMIP5_ETHZ_Amon_historical_MPI-ESM-LR_r11p1_T3M_ta_1998-2002.nc

Original source file(s) of all input file(s):

-S- (1) \@{\MODELPATH}/ETHZ_CMIP5/historical/Amon/ta/MPI-ESM-LR/r11p1/

ta_Amon_MPI-ESM-LR_historical_r11p1_199001-199912.nc

(tracking_id: ea695cd3-6234-4ddf-a68e-b4d82a2e7305)

-S- (2) \@{\MODELPATH}/ETHZ_CMIP5/historical/Amon/ta/MPI-ESM-LR/r11p1/

ta_Amon_MPI-ESM-LR_historical_r11p1_200001-200512.nc

(tracking_id: f9134520-0445-4461-9a48-14d8663dab74)

Fixes applied to original source file(s):

./reformat_scripts/fixes/CMIP5_MPI-ESM-LR_fix.ncl

[...]

Part IV

Developer's Guide

Writing a diagnostic script or a metrics set

9.1 Basics

The development of a new diagnostic (or set of diagnostics) requires the following steps before getting started:

- Creating a *FEATURE BRANCH* (see Part VI) in the applicable project subdirectory of the Git repository (via Git, see Section 16). Developers are encouraged to work actively through the Git repository. Regular “commits” to the repository help to document changes introduced to the ESMValTool and allow for efficient sharing of code with other developers.
- Creating a standard namelist following the template described in Section 5.5.

General coding rules and conventions

- Regular updates of the *FEATURE BRANCH* (see Part VI) are strongly recommended in order to keep it synchronized with the *DEVELOPMENT BRANCH* (see Part VI).
- Modularizing all diagnostic scripts as much as possible, using the general-purpose code in *lib/* and separating the diagnostic calculations from the plotting routines.
- Before creating new functions or procedures, it should be considered to use or extend the existing routines within *lib/*. Each header (see Section 9.2) provides an overview of the already implemented functions and procedures.
- Functions and procedures specific to a given diagnostic shall go in the subdirectory *diag_scripts/aux/<diagnostic>* (see Section Table 14).
- Main namelist, *diag_scripts*, functions and procedures shall be documented within the respective file using the templates (see Section 5.5 and Section 9.2).
- Each *diag_script* shall contain a call to the function *write_reference* (see also Section 9.8) in order to generate a respective acknowledgements log file (Section 9.8).

The reintegration of the feature branch into the *DEVELOPMENT BRANCH* (see Part VI) can only be done by the core development team (see Section 11.2) who shall be contacted as soon as the *FEATURE BRANCH* is ready for integration into the *DEVELOPMENT BRANCH*. Before contacting the core development team the following items should be checked:

- The new *FEATURE BRANCH* runs with different configuration options.
- If the *lib/* routines have been modified, all the diagnostics using these routines have to be tested (see automated testing, Section 9.10).

- The new code complies with the coding rules and standards (see Section 9.7) and follows the ESMValTool directory structure (see Table 14).
- All authors, contributors and data are properly acknowledged and referenced in the acknowledgements log file (see Section 9.8).
- If the new observational data are used, the scripts to “cmorize” these data shall also be made available and placed as *reformat_obs_<name>* into the folder *reformat_scripts/obs/*. Once the *FEATURE BRANCH* has been integrated into the *DEVELOPMENT BRANCH* (see Part VI), it shall be deleted from the repository.

9.2 Standard template

All (diagnostic) scripts and namelists in the ESMValTool are documented following the standards defined by templates (see Section 5.5 for the namelist template). The following describes the standard header for diagnostics scripts. The parts marked as *[text]* are the ones to be modified by the author.

- The modification history is in reverse chronological order (i.e., most recent on top) and the last entry always contains the “written” statement (optionally with a statement such as “based on” if derived from existing code).
- The author of each entry in the modification history is indicated with the author id as given in the author list in the master reference file (*doc/MASTER_authors-refs-acknow.txt*, e.g., A_surn_na = surname, name).
- All lines should be limited to a maximum of 79 characters (see Section 9.6). Exceptions can be made to improve the readability of the code.

```
;;#####
;; TITLE OF THE DIAGNOSTIC
;; Author: [Name Surname (Affiliation, Country)]
;; [PROJECT-NAME]
;;#####
;; Description
;;     [A short description of the diagnostic]
;;     [Additional description of the diagnostic]
;;     [Add more bullets if required]
;;
;; Required diag_script_info attributes (diagnostics specific)
;;     [att1]: [short description]
;;         [keep the indentation if more lines are needed]
;;     [att2]: [short description]
;;
;; Optional diag_script_info attributes (diagnostic specific)
;;     [att1]: [short description]
;;     [att2]: [short description]
;;
;; Required variable_info attributes (variable specific)
;;     [att1]: [short description]
;;     [att2]: [short description]
;;
;; Optional variable_info attributes (variable specific)
;;     [att1]: [short description]
;;     [att2]: [short description]
;;
;; Caveats
;;     [List possible caveats or limitations of this diagnostic]
;;     [Features to-be-implemented shall also be mentioned here]
;;
;; Modification history
;;     [YYYYMMDD-A_X4Y4: extended...]
;;     [YYYYMMDD-A_X3Y3: bug-fixed...]
;;     [YYYYMMDD-A_X2Y2: adapted to...]
```

(continues on next page)

(continued from previous page)

```

;;      [YYYYMMDD-A-X1Y1: written.]
;;
;; #####

load ...
load ...

begin
    ...
    ...
end

```

9.3 Library functions

The folder *diag_scripts/lib/* contains general purpose routines used by several diagnostic scripts, these library routines are grouped in subfolders by language, i.e.,

diag_scripts/lib/ncl

diag_scripts/lib/python

diag_scripts/lib/R

Library routines are grouped into individual files by topic, some examples for the NCL library routines are:

- *diag_scripts/lib/ncl/latlon.ncl*: routines to compute grid cell areas, weighted area averages, etc...
- *diag_scripts/lib/ncl/regridding.ncl*: routines interfacing the ESMF regridding functions in NCL
- *diag_scripts/lib/ncl/statistics.ncl*: statistical routines not (yet) implemented in the standard distribution of NCL
- *diag_scripts/lib/ncl/style.ncl*: centralized control of NCL plot styles, e.g., defines line colors/dashes/thickness for each model name in CMIP5, based on the style files in *diag_scripts/lib/ncl/styles/*.

For further details on the library functions, see the documentation given in the header of the functions themselves (see Section 9.2 for a template).

9.4 Plotting functions

The folder *plot_scripts/* contains general purpose routines used for plotting by the diagnostic scripts. The plotting functions should facilitate the separation of computing the diagnostic and displaying the result. To this end they should handle both the case when called directly from the diagnostic script (with data to visualize as an argument), and the case when the computed diagnostic is passed along as a netCDF file. These plotting routines are grouped in subfolders by language,

- *plot_scripts/ncl*
- *plot_scripts/python*
- *plot_scripts/R*

Each subfolder further groups the plotting routines into files by topic, e.g., for the NCL library routines:

- *plot_scripts/ncl/contour_maps.ncl*: interfaces NCL plotting routines for contour map plots, contour polar maps and adding markers to contour maps
- *plot_scripts/ncl/scatterplot.ncl*: interfaces NCL plotting routines for scatter plots

For further details on the plotting functions, see the inline documentation in the functions themselves.

9.5 Adding new variables

Adding new variables requires changes to *reformat_scripts/recognized_vars.dat* (Section 9.5.1) and possibly also to *reformat_scripts/recognized_units.dat* (Section 9.5.2). In addition, a new definition file *variable_defs/<varname>.ncl* is needed (Section 9.5.3; see Table 1 for a list of currently available variable definition scripts). If the variable is a **non-derived** variable (explained in Section 9.5.3) it also needs to be defined in a file named *reformat_scripts/cmor/CMOR_<variable>.dat* (see Section 9.5.4).

9.5.1 reformat_scripts/recognized_vars.dat

New variables have to be added to *reformat_scripts/recognized_vars.dat*. Two lines are added per variable:

- `std_name = varname`
standard CMOR variable name
- `alt_name = alternative name 1, alternative name 2, ...`
comma separated list of alternative variable names

Example (surface pressure)

- `std_name = ps`
- `alt_name = aps,PS,psurf`

The ESMValTool reformat scripts will look for variable “varname” in the input files. If not found, the alternative variable names “alternative name 1”, “alternative name 2”, etc. are tried before an error message is issued that the variable could not be found.

9.5.2 reformat_scripts/recognized_units.dat

The file *reformat_scripts/recognized_units.dat* contains a list of known units. If needed, the unit of the newly added variable can be added. There are two lines per unit:

- `std_name = unit`
standard CMOR unit
- `alt_name = alternative unit`
comma separated list of possible alternative units and corresponding conversion factor, defined as
`units[cmor] = units[alternative] * factor`

Example (dobson units)

- `std_unit = DU`
- `alt_unit = g m-2, 4.6707e-5, kg m-2, mol m-2, 2.2414e-3`

9.5.3 variable_defs/varname.ncl

The file *variable_defs/<varname>.ncl* is a NCL script containing the declaration of the variable “varname” including its specific attributes. In case of derived variables, a function “calculate” calculating the derived variable must be defined in the script *<varname>.ncl* (see Table 1 for a list of currently available variable definition scripts).

Remarks

1. For derived variables, a statement specifying the (standard, non-derived) variables required to calculate the derived variable is needed. In the example given below, this statement in the beginning of the NCL script looks like

*; Requires: rsut:T2*s,rsutcs:T2*s*

In this example, the two standard variables “rsut” and “rsutcs” are needed to calculate the shortwave cloud forcing.

2. Variable attributes are specified as attributes of the variable “variable_info” (see examples below). In order to activate the variable attributes, “variable_info” must be set to “True”. Some examples for frequently used attributes are:

- variable_info@derived = False (True)
- variable_info@long_name = “...”
- variable_info@units = “...”
- variable_info@standard_name = “...”
- variable_info@short_name = “...”

Example (precipitation, standard variable)

```
; Requires: none
variable_info = True
variable_info@derived = False
```

Example (shortwave cloud forcing, derived variable)

```
; Requires: rsut:T2*s,rsutcs:T2*s

[...]

variable_info = True
variable_info@derived = True
variable_info@long_name = "CS Shortwave cloud radiation effect"
variable_info@units = "W m-2"

undef("calculate")
function calculate( index [1] : integer, \
                  variable [1] : string, \
                  field_type [1] : string )
;;
;; Arguments:
;;   index - index to current infile defined in the
;;           'interface_data/ncl.interface'-file
;;   variable - Current variable as string
;;   field_type - string with field type classification
;; Return value:
;;   data_new logical

local tmp, tmp1, tmp2, dum1, dum2, dum, i, verbosity
begin
    data_new = True
    tmp1 = read_data(index, "rsut", "T2Ms")
    tmp2 = read_data(index, "rsutcs", "T2Ms")
    dum1 = extract_data(index, tmp1, -1, 0, 0)
    dum2 = extract_data(index, tmp2, -1, 0, 0)

    dum = dum1
    dum = dum2 - dum1
    dum@long_name = variable_info@long_name
    dum@units = variable_info@units
```

(continues on next page)

(continued from previous page)

```

add_data_var(index, data_new, dum, variable)

return(data_new)
end

```

9.5.4 reformat_scripts/cmor/CMOR_variable.dat

Each standard variable (non-derived) also needs a configuration file indicating the expected units of the variable. The expected units are read from the file *reformat_scripts/cmor/CMOR_variable.dat* which follows the definitions in the official CMOR tables for CMIP5. If this file is missing for a specific variable, it can be downloaded from <http://pcmdi.github.io/cmor-site/tables.html>. If a CMOR table for the new variable is not available, the user can create a new one based on the existing tables (e.g., following the example in *reformat_scripts/cmor/CMOR_mmrbcfree.dat* based on *reformat_scripts/cmor/CMOR_mmrbc.dat*).

Example, reformat_scripts/cmor/CMOR_pr.dat

```

SOURCE: CMIP5
!=====
variable_entry:    pr
!=====
modeling_realm:    atmos
!-----
! Variable attributes:
!-----
standard_name:     precipitation_flux
units:             kg m-2 s-1
cell_methods:      time: mean
cell_measures:     area: areacella
long_name:         Precipitation
comment:           at surface; includes both liquid and solid phases from all_
↳types
                   of clouds (both large-scale and convective)
!-----
! Additional variable information:
!-----
dimensions:        longitude latitude time
out_name:          pr
type:              real
valid_min:         0
valid_max:         0.001254
ok_min_mean_abs:   2.156e-05
ok_max_mean_abs:   3.215e-05
!-----

```

9.6 Coding rules and standards

The purpose of the code conventions used in ESMValTool is to ensure a high degree of consistency in the code layout. Consistently structured code increases readability and understanding of the code making it easier for developers and users work with a given piece of the code base. It is important to emphasize two points:

- Checking the code consistency should be done by software as this allows the check to be done automatically.
- Code checkers are available at *util/ncl-checker/pep8.py* (NCL) and *util/pep8-checker/pep8.py* (Python).

The code conventions are guidelines and should be treated as such. There are circumstances when it is advisable, for various reasons such as improved readability, to ignore some of the guidelines.

Code conventions used for Python

Python code should conform to the PEP-8 style guide [PEP8 2001]. Recommended tools to check Python code is the official PEP8-checker that is provided with the ESMValTool distribution (*util/pep8-checker/pep8.py*) and PyFlakes.

To use it on a python file, cd into *util/pep8-checker/*, and run,

```
cd util/pep8-checker
python pep8.py <path-to-python-file>
```

Python: Pyflakes

Besides the PEP8-checker also the use of the ‘pyflake’-tool is recommended (see the pyflakes homepage <https://pypi.python.org/pypi/pyflakes> for details). For a local install of pyflakes, try virtualenv, e.g., if the virtualenv already is installed, run

```
source sandbox-pybot/bin/activate
pip install --upgrade pyflakes
pyflakes <python-file>
```

Code conventions for NCL

NCL code in ESMValTool should follow the PEP-8 style guides. An NCL adapted version of the Python PEP-8 checker is available in the ESMValTool repository (*util/ncl-checker/pep8.py*). Please note that the NCL checker may report some false-positive (e.g., the reading symbol *->* is not recognized as such).

To use the NCL version of the PEP8-checker provided with the ESMValTool distribution, run

```
cd util/ncl-checker
python pep8.py <path-to-NCL-file>
```

The NCL-version is adaption of the Python checker and works satisfactorily as long as one keeps in mind the false positives it finds due to language differences between Python and NCL. These false positives may be addressed in the future depending on priorities.

Code conventions for R

The code conventions for R should conform to the formatting produced by the R parser tree. This method is further described at “Tidying-R-code” (<https://cran.r-project.org/doc/manuals/R-exts.html#Tidying-R-code>). Note that this method can only be considered semi-automatic since it does preserve comments (they need to be repatched) and does not produce very nice line breaks.

9.7 Documentation of software

In order to ensure that all code can be maintained, all diagnostic packages must be well documented. It is the responsibility of the software developers to embed their documentation into the namelist and source code. For details see Sections 9.9 and 10. Documentation systems exist to organize embedded documentation into well structured, linked documents.

- **R:** documentation should follow CRAN guidance.
- **Python:** the Sphinx package allows embedded documentation to be assembled into indexed web pages (see Section 9.9)
- **NCL and namelists:** a Sphinx extension has been developed to extract code documentation for NCL and namelists (see Section 9.9)

9.8 The acknowledgements log file

The acknowledgements log file automatically created by each diagnostic (see also Section 8.1) is written by the function *write_references* (*interface_scripts/messaging.ncl*, see below), which uses the tags defined in the master reference/acknowledgements file (*doc/MASTER_authors-refs-acknow.txt*) as input. This master file lists all

authors and contributors (tags starting with A_), the diagnostic references (tags with D_), references for observational data (tags E_) and projects (tags P_).

The function `write_references`

The function `write_references` (defined in *interface_scripts/messaging.ncl*) should be called at the end of each diagnostic script in order to write the acknowledgements log file. The function has the arguments “author(s)”, “contributors”, “diagnostics”, “observations”, “projects” which are arrays of strings. All strings (“tags”) used must be defined in the master reference file *doc/MASTER_authors-refs-acknow.txt*. The tags are then replaced by the function `write_references` with their definition when writing the acknowledgements log file. All tags in the master reference file are sorted by category of which there are four in total:

```
A_XXX = authors, contributors (xxx = author name)
e.g., A_###

D_XXX = diagnostics
e.g., D_righi15gmd = Righi et al., Geosci. Model Dev., 8, 733–768 doi:10.5194/gmd-
→8-
    733–2015, 2015.

E_XXX = observational data
e.g., E_era40 = ERA40

P_XXX = project
e.g., P_embrace = EU FP7 project EMBRACE

write_references(diag_script, \
    "A_###", \
    ("/D_righi15gmd", "D_gleckler08jgr"/), \
    ("/E_kalnay96bams", "E_erainterm", "E_airs", "E_ceresebaf", "E_srb"/), \
    ("/P_embrace", "P_esmval/"))
```

9.9 Documentation of source code

The Sphinx documentation generator (<http://sphinx-doc.org>) is used to organize and format ESMValTool documentation, including text which has been extracted from source code. Sphinx can help to create documentation in a variety of formats, including HTML, LaTeX (and hence printable PDF), manual pages and plain text.

Sphinx may be obtained from <http://sphinx-doc.org/install.html>; an overview of its workings is available at <http://sphinx-doc.org/tutorial.html>. In ESMValTool, Sphinx has been used to set up the files in *doc/sphinx*. Running `make <target>` in that directory will cause the documentation to be built, and its output placed in the *build/<target>* subdirectory. Here, *<target>* is the format required for example, *html*, *latexpdf*, *man* or *text* for the four example formats mentioned above. Running `make` by itself will generate a complete list of output formats.

Sphinx was originally developed for documenting Python code, and one of its features is that it is able using the so-called autodoc extension to extract documentation strings from Python source files and use them in the documentation it generates. This feature apparently does not exist for NCL source files (such as those which are used in ESMValTool), but it has been mimicked (or more-or-less reverse-engineered) here via the Python script *doc/sphinx/scripts/process_ncl_docs.py*, which walks through a subset of the ESMValTool NCL scripts, extracts function names, argument lists and descriptions (from the comments immediately following the function definition), and assembles them in a subdirectory of *doc/sphinx/source*. These output files are in the so-called reStructuredText format (see, e.g., <http://docutils.sourceforge.net/rst.html>), which is the markup language used by Sphinx; running `make` in *doc/sphinx* builds the ESMValTool documentation from them, as noted above.

Note: See Section 10.2 for more details on how to document a new diagnostic.

9.10 Automated testing

Any changes to a programming code have the risk of introducing unwanted side effects on some other parts of a code and introduce bugs. Routine and automated testing is therefore essential to maximize the code quality and ensure integrity of all diagnostics implemented within ESMValTool.

9.10.1 Setup and general workflow

Automated testing within the ESMValTool is implemented on two complementary levels:

- **unittests** are used to verify that small code units (e.g. functions/subroutines) provide the expected results
- **integration** testing is used to verify that a diagnostic integrates well into the ESMValTool framework and that a diagnostic provides expected results. This is verified by comparison of the results against a set of reference data generated during the implementation of the diagnostic.

Installation of the test environment

All scripts required to run the test environment are provided together with the ESMValTool code. Two external python packages are required which can be installed using the python package manager (pip; <https://pypi.python.org/pypi/pip>) as follows in a linux environment:

```
# install nosetests (https://nose.readthedocs.org/en/latest/)``
pip install nose
# install easytest
pip install easytest
```

General functionality of testing framework

Each diagnostic is expected to produce a set of well-defined results. These are files in a variety of formats and types (e.g. graphics, data files, ASCII files ...). While testing results of a diagnostic, a special namelist file is executed by ESMValTool which runs a diagnostic on a limited set of test data only. A small test data set is chosen to minimize executing time for testing while ensuring on the other hand that the diagnostic produces the correct results. The following general tests are implemented at the moment for diagnostics with available test data:

- **Check for file availability:** a check is performed that all required output data have been successfully generated by the diagnostic. A missing file is always an indicator for a failure of the program.
- **File checksum:** While the previous test only checks if a file is available, the checksum verifies if the content of a file is similar. Currently the MD5 checksum is used to verify that contents of a file are the same. The MD5 checksum is a good proxy for the similarity of two files and is used regularly to ensure integrity between files when transferring files between different computers.
- **Graphics check:** For graphic files an additional test is therefore implemented which verifies that two graphical outputs are identical. This is in particular useful to verify that outputs of a diagnostic remain the same after code changes.

Testing the ESMValTool diagnostics

Unit tests are implemented for each diagnostic independently. Details on running unit tests using **nose** is as simple as going to the ESMValTool root directory and then execute the following shell command:

```
# run nosetests
nosetests
```

This will search recursively for test files and execute these tests. A statistic on success and failures is provided at the end of execution. More details on using nose can be found in the package's documentation (<https://nose.readthedocs.org/en/latest/>).

To run integration tests for each diagnostic, a small script needs to be written once. An example for a file named `esmvaltooltest.py` is provided in Section 9.10.2. To run all tests for diagnostics implemented in this file the following command needs to be executed:

```
# run integration tests
python esmvaltooltest.py
```

A summary of success and failures is provided as output.

9.10.2 Example test implementation for a diagnostic

In the following an example is given how to implement a test environment for a new diagnostic with just a few lines of code. File: esmvaltooltest.py

```
"""
sample script for ESMValTool testing
"""

from esmvaltool import ESMValToolTest

"""
Define a new class for testing a particular diagnostic
"""

class PerfMetricCMIP5Test(ESMValToolTest):
    def __init__(self):
        # 1) define here the name of the test namelist
        nml_name = 'namelist_perfmetrics_CMIP5_test.xml'

        # 2) specify here the full path of the namelist
        # (relative to ESMValTool root)
        nml = 'nml/test_suites/dlr/' + nml_name

        # 3) define here the location of the reference data directory
        # note that it is expected that the directory has the same
        # name as the namelist
        refdir = esmval_dir + os.sep + os.path.splitext(nml_name)[0] + \
            '/output/plots/'

        # initialize the parent class
        super(PerfMetricCMIP5Test, self).__init__(nml=nml,
            refdirectory=refdir, esmval_dir=esmval_dir)

# -----

# This is how you run a test
PT = PerfMetricCMIP5Test() # create instance of test class
PT.run_nml() # run the testing namelist
PT.run_tests(execute=False, graphics=None,
    checksum_files='all', files='all') # perform tests
```

9.11 Checklist

The following table can be used as a list of items to be done/checked when writing a new diagnostic.

Table 9.11 Example checklist for implementing new diagnostics and new observational datasets.

diagnostic / namelist	model data	observational data
header in diagnostic code	preprocessing (reformatting routines - if applicable)	preprocessing (reformatting script)
header in namelist	list of tools (if applicable)	list of tools (if applicable)
documentation of diagnostic (.rst file + example images, see Section 10)	references	header in reformatting script (if applicable)
provenance (tagging) implemented in diagnostic code (see Section 10.3)	test data	references
testing namelist (runs diagnostic with reduced input data)		test data
coding rules and standards (see Section 9.6) have been followed		
<i>FEATURE BRANCH</i> updated with latest <i>DEVELOPMENT BRANCH</i>		
list of tools, libraries, etc.		
references		
contact person for scientific support		

Scientific documentation of a diagnostic script or metrics set

10.1 Basics

An important part of the implementation of a new diagnostic script is the documentation of the diagnostic itself as well as the documentation of the observational data sets (see also Section 9). The former should comply with the standard template for new diagnostics (see Section 9.2 below) and the latter should include instructions how to download the observational data and, if necessary, scripts to convert it to the format required in ESMValTool (see Section 10.4 below).

The documentation of a diagnostic requires the following steps:

- creating a documentation in reStructuredText format (see, e.g., <http://docutils.sourceforge.net/rst.html>) following the structure defined in the template *doc/sphinx/source/namelist/namelist_template.rst* (see also Section 9.2)
- the new documentation file is saved as *doc/sphinx/source/namelist/namelist_<yourdiag>.rst*
- all image files are stored in the directory (to be created) *doc/sphinx/source/namelist/figures/<yourdiag>*
- the new documentation file is added to the alphabetically sorted list of all namelists in the main document *doc/sphinx/source/index.rst* (under “Annex C - namelists”)

Note: Example image files are usually in the format .png and should have a maximum file size of a few hundred kb or less.

10.2 Standard template

When implementing a new diagnostic script or metrics set, it should be documented starting from the standard template given below. Examples can be found in Part VII (Annex C - namelists):

```
Title of diagnostic/performance metric
=====

Overview
-----
```

(continues on next page)

(continued from previous page)

Provide a brief scientific description of the diagnostic including references.

Available namelists and diagnostics

Namelists are stored in nml/

* namelist_mydiag_1.xml

* namelist_mydiag_2.xml

Diagnostics are stored in diag_scripts/

* mydiag_1.py

* mydiag_2.ncl

User settings

List and describe all settings and input parameters used by your diagnostic.

Variables

* variable 1 (...)

* variable 2 (...)

Observations and reformat scripts

* observational dataset 1, reformat script: reformat_scripts/obs/reformat_obs_mydata1.py

References

* reference 1

* reference 2

Example plots

.. figure:: /namelists/figures/mydiag/myexample1.png

:scale: 50 %

:alt: xxxx

Caption can go here.

10.3 Provenance (tagging)

In order to ensure provenance of the ESMValTool results a set of tags is added to all namelists and diagnostics. These tags are written as meta data to all figures produced by the ESMValTool and can be used for reporting and visualization. There are two kinds of tags: tags included in the namelist (“namelist tags”) and tags in the individual diagnostics (“diagnostic tags”). Namelist tags include *global* tags such as “main reference” and “project” as well as *diagnostic specific* tags for each individual diagnostic block such as “theme” and “realm”. Diagnostic tags include “domain”, “plot type” and “statistics”.

Note: All available standard tags are defined in *doc/MASTER_authors-refs-acknow.txt*.

- D_xxxx (reference),
- P_xxxx (project),
- R_xxxx (CMIP6 realm),
- T_xxxx (theme),
- DM_xxxx (domain),
- PT_xxxx (plot type),
- ST_xxxx (statistics).

In addition to the namelist and diagnostic tags, also the name of the actual namelist, variable name(s), model name(s), contributing author(s) and tracking IDs of the input files are collected as meta-data to be written into the EXIF header of all figure files (.png).

Note: For tagging to work properly, all plots must be written to individual output files, i.e. no multi-page (.ps) files are supported.

10.3.1 Example namelist tags in <GLOBAL>

All available tags are defined in *doc/MASTER_authors-refs-acknow.txt*.

```
<GLOBAL>
.
.
<tags> D_righi15gmd, P_cmug </tags>
</GLOBAL>
```

10.3.2 Example namelist tags in <diag>

All available tags are defined in *doc/MASTER_authors-refs-acknow.txt*.

```
<diag>
.
.
<tags> R_ocean, R_land, T_phys </tags>
</diag>
```

10.3.3 Example diagnostic tags (NCL)

Header section

```
; this include defines the procedure ESMValMD used to write the
; meta-data to the image file(s)

load "./diag_scripts/lib/ncl/meta_data.ncl"
```

Example code after writing figure

```
; the field 'tags' is generated by the ESMValTool framework and contains
; all tags defined in the namelist;

; add diagnostic specific tags e.g., domain (DM_xxxx), plot type (PT_xxxx),
; statistics (ST_xxxx)
; (all available tags are defined in doc/MASTER_authors-refs-acknow.txt)
```

(continues on next page)

(continued from previous page)

```

alltags = array_append_record(tags, ("/DM_global", "PT_geo", "ST_mean"/), 0)

; define caption for the plot produced by the ESMValTool

caption = "This is an example caption. Model: " + models@name(0) + ", \
        variable: " + variables(0)

; define an arbitrary but unique id for the image file

id = diag_script + "_example_id"

; generate list of all input files processed for this particular figure

climofiles = interface_get_inpaths(imod) + "/" \
            + interface_get_infile(variables(0), field_types(0), imod)

; define list of contributing authors for this diagnostic script

contrib_authors = ("/A_gott_kl", "A_eval_ma"/)

; =====
; write meta-data to image file
; =====

; parameters for procedure ESMValMD
; -----
; outfile: filename of image file created
; alltags: concatenated list of all tags (from the namelist + diagnostic specific)
; variables: list of all variables pocessed
; models@name: list of all model/dataset names processed
; climofiles: list of all input files processed
; diag_script: name of diagnostic script (generated by the ESMValTool framework)
; contrib_authors: list of contributing authors for this diagnostic script

ESMValMD(outfile, alltags, caption, id, variables(0), models@name(0), \
        climofiles, diag_script, contrib_authors)

delete([/alltags, caption, id, climofiles/])

```

10.3.4 Example diagnostic tags (Python)

Header section

```

# this include defines the procedure ESMValMD used to write the
# meta-data to the image file(s)

from ESMValMD import ESMValMD

```

Example code after writing figure

```

E = ESMValProject(project_info)
diag_script = E.get_diag_script_name()

# filename of plot
plot_type = project_info['GLOBAL']['output_file_type']
oname = plot_dir + 'ww09_metric_multimodel.' + plot_type

# tags from the ESMValTool framework (namelist)

```

(continues on next page)

(continued from previous page)

```

basetags = [x.strip() for x in project_info.get('GLOBAL')['tags']]

# create list of model names
models = []
for model in E.project_info['MODELS']:
    models.append(model.split_entries()[1])

# variable name(s)
variables = project_info['RUNTIME']['currDiag'].get_variables()

# all variable tags must be preceeded by "V_"
vartags = ['V_' + item for item in variables]

# all model tags must be preceeded by "M_"
modeltags = ['M_' + item for item in models]

# define caption for the plot produced by the ESMValTool
caption = 'Cloud Regime Error Metric (CREM) following Williams and Webb (2009, ↵
↵Clim. Dyn.).'

# list of all input files processed
tmp = []
for variable in variables:
    tmp.append(E.get_clim_model_filenames(variable=variable))
model_filelist = [item for sublist in tmp for item in sublist]

# list of contributing authors
authors = 'A_will_ke'

# define an arbitrary but unique id for the image file
plot_id = '#ID_ww09_crem'

# =====
# write meta-data to image file
# =====

# parameters for procedure ESMValMD
# -----
# oname: filename of image file created
# tags: concatenated list of all tags (from the namelist + diagnostic specific
#       + model tags + variable tags)
# model_filelist: list of all input files processed
# diag_script: name of diagnostic script (generated by the ESMValTool framework)
# authors: list of contributing authors for this diagnostic script

ESMValMD("both",
         oname,
         basetags + ['DM_global', 'PT_bar'] + modeltags + vartags,
         caption,
         plot_id,
         model_filelist, diag_script, authors)

```

10.4 Model and observational data

10.4.1 Overview

When possible, observations from the obs4MIPs/ana4MIPs archives are used in the model evaluation (see Section 8.1). These data are freely available from the ESGF in the same format as the CMIP simulations and can be directly used in the ESMValTool using the obs4mips or ana4mips class in the namelist (see also Section 8.2).

Important links

https://www.earthsystemcog.org/projects/obs4mips/satellite_data_products

Nightly scan across nodes

<https://www.earthsystemcog.org/search/obs4mips/?template=obs4mips&limit=200>

Observational data sets not available in these archives need to be reformatted according to the CF/CMOR standard before they can be used (see guidelines in Section 12). In this case a reference to the official URL is provided such that a user can get the latest version of the data set as well as a description and a script how to convert the data set to the format required by the ESMValTool. These conversion scripts are collected in *reformat_scripts/obs/reformat_obs_<NAME>.ncl*. The reformatting routines must be documented with a standard header providing all information required to retrieve and process the data, as well as their availability (Tier 1, Tier 2, or Tier 3).

All observations are tiered as follows:

- Tier 1: data sets from the obs4MIPs and ana4MIPs archives
- Tier 2: other freely available data sets
- Tier 3: restricted data sets (e.g., license agreement required)

For Tier 2 and 3 data, the developer shall also provide links and helper scripts through the reformatting routines, following the template for the standard header described in section for the reformatting routines. An example can be found here:

reformat_scripts/obs/reformat_obs_AURA-MLS-OMI.ncl.

An overview on the available reformatting scripts for Tier 2 and 3 data is given in Table 1. The reformatted observational data (Tier 2 and Tier 3) must be named according to the OBS class definition, which considers the following naming convention:

OBS_<name>_<case>_<ensemble>_<field>_<variable>_<YYY1M1>-<YYY2M2>.nc

where:

<name> is the name of the satellite, instrument, campaign, network, model, etc. (e.g., ERA-Interim, AERONET, AURA-MLS-OMI, etc.)

<case> is the observation type (insitu, ground, sat, reanaly, campaign, etc.)

<ensemble> is the version number, processing level or station code (for ground-based networks), use 1 if not available.

It is also possible to split the output in multiple files, like in the CMIP5 class, e.g. _200101-200512.nc, 200601_201012.nc, 201101-201512.nc, etc. This is particularly useful for daily data, which are usually too large to be collected in a single file covering the whole time period.

Note: When adding new observational datasets, these datasets have to be added to Table 1 (in doc/sphinx/source/running.rst).

10.4.2 Standard header for the reformatting routines for observational data

This is a template of the standard header for the reformat_obs routines. The parts in red are the ones to be modified by the author. The modification history is given in reverse chronological order (i.e., most recent on top) and the last entry always contains the written statement. The author of each entry in the modification history shall be indicated with the author tag, as given in the master reference file (*doc/MASTER_authors-refs-acknow.txt*), e.g., A_surn_na = surname, name. All lines should be limited to a maximum of 79 characters.

```
;; #####
;; REFORMAT SCRIPT FOR THE [OBSERVATION NAME] OBSERVATIONAL DATA
;; #####
```

(continues on next page)

(continued from previous page)

```

;;
;; Tier
;;   [Information on data availability, possible options are:]
;;   Tier 1: obs4MIPs or ana4MIPs
;;   Tier 2: other freely-available data set
;;   Tier 3: restricted data set
;;
;; Source
;;   [URL to the data source or the reference]
;;
;; Last access
;;   [YYYYMMDD]
;;
;; Download and processing instructions
;;   [Short explanation on how to download and process the data]
;;
;; Caveats
;;   [List possible caveats or limitations of this script]
;;   [Features to-be-implemented shall also be mentioned here]
;;
;; Modification history
;;   [YYYYMMDD-A_xxxx_yy: extended...]
;;   [YYYYMMDD-A_xxxx_yy: written.]
;;
;; #####

load ...
load ...

```

Tasks and responsibilities

11.1 Mailing list

A mailing list has been set up for all general and technical questions on the ESMValTool such as, for instance, questions on installation, application or development. You are encouraged to subscribe to the ESMValTool user mailing list by sending an email to Listserv@dlr.de with the following text:

subscribe ESMValTool-Usr

11.2 The ESMValTool core development team

- Deutsches Zentrum für Luft- und Raumfahrt (DLR), Institut für Physik der Atmosphäre, Germany (PI)
ESMValTool Core PI and Developer: contact for requests to use the ESMValTool and for collaboration with the development team, access to the PRIVATE GitHub repository (see Part VI)
- Alfred-Wegener-Institute Bremerhaven (AWI), Germany (overseeing EU Horizon 2020 APPLICATE and TRR181 ESMValTool work)
- Barcelona Computing Center (BSC), Spain (overseeing EU Horizon 2020 PRIMAVERA ESMValTool work)
- Ludwig Maximilian University of Munich, Germany (overseeing EU Horizon 2020 CRESCENDO ESMValTool work)
- Netherlands e-Science Center (NLeSC), The Netherlands
- University of Reading, United Kingdom

Contacts for specific diagnostic sets are the respective authors, as listed in the corresponding diagnostic documentation and in the source code.

11.3 Pull requests

The development of the ESMValTool is a community effort and involves a number of tasks to allow for a smooth integration of new code into the ESMValTool (*for technical details on how to work with the version controlled repository for the ESMValTool source code see Part VI*). These tasks are divided into:

1. responsibilities for the ESMValTool Core Development Team that oversees the development and prepares releases of the ESMValTool and
2. responsibilities for the individual developers contributing code for new diagnostics or metrics.

11.3.1 ESMValTool Core Development Team

1. Responsible for the *MASTER BRANCH* and *DEVELOPMENT BRANCH* including pull requests by developers (see below)
2. Oversee collection of observations that are required to actually run all diagnostics that are in the *DEVELOPMENT BRANCH*
3. Responsible for general quality control, maintenance, and documentation (including compliance with coding rules)
4. Oversee collection of testing data (observations/reference model/idealized data) that are required for automated testing (host a local copy)
5. Responsible for new releases of the ESMValTool

11.3.2 ESMValTool developers

1. Accept the **ESMValTool license and ESMValTool Development terms of use** (see <http://www.esmvaltool.org/license.html>).
2. Provide documentation that is compliant with **documentation templates for diagnostics and metrics sets** (see Sections 5.5 and 9.2).
3. Provide well documented code that follows the coding rules and standards. It is recommended to use an existing diagnostic in the target language as a template for the development. Check with the Core Development Team if unsure which diagnostic to use as a template.
4. For each pull request to implement a diagnostic set into the *DEVELOPMENT BRANCH*:

Scientific analysis

- Provide the code for **all diagnostics and metrics** that are called; all source code files should include a standard header as defined in Section 9.2.
- Standard namelist (with settings so that it runs on, if possible, all CMIP5 models) and corresponding plots (for documentation), the namelist should include a standard header as defined in Section 5.5.
- Provide documentation of the diagnostic following the structure defined in the template `doc/sphinx/source/namelists/namelist_template.rst` (reStructuredText (RST) syntax) including example images (copy to `doc/sphinx/source/namelists/figures/<namelist>/`).
- Provide the **full set of observations** that allows a scientific application of the full standard namelist list (indicate source and if applicable license issues).
- Provide that **observations are documented**. If new observations are introduced, add an entry to the table listing the available observations (Table 1) in `doc/sphinx/source/running.rst` and that a reformat routine is available if the original source does not follow the CMOR standard.

Automated testing (see Section :numref:'auto_test')

- Provide the code for automated testing for the diagnostic set that should be integrated into the *DEVELOPMENT BRANCH* (see Part VI).
- Provide a **namelist for automated testing**.
- Provide a **reduced and small set of observations/reference model/idealized data** for each diagnostic that is called by the testing namelist.
- Provide **netCDF output + example plots for automated testing** based on the reduced dataset and the standard namelist as a reference.

5. **Name a contact person** providing (scientific) support for your diagnostics.

Guidelines for data processing

12.1 Observations

Basic steps

1. Start from the original, if possible referenced data product (avoid using unpublished data pre-processed by a third party).
2. Document how to obtain the data (e.g., link to the data source, download instructions, info on contact persons, references, etc.) in as much detail as necessary for another person to successfully obtain the data.
3. Document any pre-processing steps applied to the original data (e.g., conversion of units, interpolation to pressure levels, arithmetic calculations, conversion of file format, etc.). If possible, use only commonly available tools such as, for instance, NCAR Command Language (NCL), Climate Data Operators (cdo), netCDF Operators (NCO), Python, etc.
4. Save the data in netCDF format following the CMOR/CMIP5 or CMOR/CMIP6 conventions (see links below for variable names, units, grid definition, etc.). N.B.: The processed observational data set (step 3) does not follow the CMIP5-DRS but uses a different DRS. When naming the output observational data set file in step 3 above, follow the naming conventions in any of the existing reformat scripts in the folder “reformat_scripts/obs/”.
5. Write/provide a script (e.g., shell script, NCL program, Python code, etc.) handling all pre-processing and conversion steps, and including all the necessary information to retrieve the data from the original source. Many examples of such scripts are available as part of the ESMValTool distribution and can be found in the ESMValTool directory reformat_scripts/obs/.

12.2 Model data

- If possible, provide model data in netCDF format following the CMOR conventions (see links below, Section 12.3).
- In some cases, it might be worth creating reformat routines that read the native model output and convert the data on the fly within the ESMValTool to CMOR compliant files. Examples for such reformatting routines for the models EC-Earth, EMAC, and GFDL are available as part of the ESMValTool distribution (ESMValTool directory reformat_scripts/<model>/).

12.3 Links

- CMOR user's guide: http://www-pcmdi.llnl.gov/software/cmor/cmor_users_guide.pdf
- ESGF CMOR Data Converter: <https://cds.nccs.nasa.gov/tools-services/esgf/obs4mips/>
- CMIP5 output data requirement (variables and units): http://cmip-pcmdi.llnl.gov/cmip5/docs/standard_output.pdf
- CMIP6 preliminary output data requirement (variables and units): <http://clipc-services.ceda.ac.uk/dreq/mipVars.html>

12.4 Help

Support from the ESMValTool Core Development Team (Section 11.2) for observational data is available provided that you followed the basic steps outlined above. For model data, help from the core development team is available if you can provide us with:

- netCDF files and a complete list of all details on the model output including e.g., a brief description of the quantity
- native variable name
- units
- vertical grid definition
- other relevant information on the data set

CHAPTER 13

References

- Flato, G., Marotzke, J., Abiodun, B., Braconnot, P., Chou, S. C., Collins, W., Cox, P., Driouech, F., Emori, S., Eyring, V., Forest, C., Gleckler, P., Guilyardi, E., Jakob, C., Kattsov, V., Reason, C., and Rummukainen, M.: Evaluation of Climate Models. In: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, Stocker, T. F., D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex and P.M. Midgley (Ed.), Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2013.
- NCL (2014) The NCAR Command Language (Version 6.2.1) [Software]. (2014). Boulder, Colorado: UCAR/NCAR/CISL/VETS. <http://dx.doi.org/10.5065/D6WD3XH5>
- PEP8 (2001) <https://www.python.org/dev/peps/pep-0008/>
- Taylor, K. E., Stouffer, R. J., and Meehl, G. A.: An Overview of Cmp5 and the Experiment Design, B Am Meteorol Soc, 93, 485-498, 2012.
- XML <http://www.w3.org/TR/xml/www.gmail.com>

Part V

Annex A - tables

CHAPTER 14

More tables

Table 14 Directory structure of the ESMValTool sorted by file type.

<i>Namelist</i>	
nml/namelist_XyZ.xml	Namelist for specifying general parameters, input data and diagnostics to run.

<i>Configuration files</i>	
nml/cfg_XyZ/cfg_XyZ_*.typ	Configuration files for diagnostic scripts. The suffix “*.typ” specifies the language the routine is written in. Note: there is usually than one configuration script per diagnostic set.

<i>Scripts</i>	
main.py	Driver script controlling the overall program flow
diag_scripts/ <ul style="list-style-type: none"> • MyDiag.ncl • Sealce_polcon.ncl • SAMonsoon.ncl • etc. 	Directory containing all diagnostics called by the namelists. Supporting routines are placed in “diag_scripts/lib” under the subdirectory corresponding to the programming language used (NCL, Python, R).
aux/ <ul style="list-style-type: none"> • <diagnostic>/ 	Functions and procedures specific to a given diagnostic are stored in the subdirectory diag_scripts/aux/<diagnostic>.
lib/ <ul style="list-style-type: none"> • ncl/ <ul style="list-style-type: none"> – ensemble.ncl – regrid.ncl – statistics.ncl – style.ncl – ... • python/ <ul style="list-style-type: none"> – ensemble.py – regrid.py – statistics.py – style.py – ... • ... for other languages (e.g., R) 	Functions that are called by the diag_scripts, for example statistics.type collects all statistical functions in a single file. When adding a new function, it must be added to list in the header.
plot_scripts/ <ul style="list-style-type: none"> • ncl/ <ul style="list-style-type: none"> – plotting1.ncl – plotting2.ncl – ... • python/ <ul style="list-style-type: none"> – plotting1.py – plotting2.py • ... for other languages (e.g., R) 	Plotting routines; files should have an intuitive name for their purpose. Data to be plotted may be passed to them directly or via netCDF files.
interface_data/	Inter-process communication, e.g., between Python and NCL/R, is done by sourcing NCL/R specific files updated on the fly in this folder. These intermediate files are based on templates files for different languages.
interface_scripts/	Routines called from the workflow manager script “main.py”, mainly used to handle the control flow of the tool, e.g., parsing namelists, updating temporary files in the folder interface_data/, etc).

reformat_scripts/	Routines for checking or reformatting raw input data.
reformat_scripts/cmor/ <ul style="list-style-type: none"> CMOR_<var>.dat 	Contains the CMOR tables, defined as plain-text files CMOR_<var>.dat, where <var> is the variable's standard name. Additional tables can be added by the users, e.g., from http://www2-pcmdi.llnl.gov/cmor/tables/ .
reformat_scripts/default/ <ul style="list-style-type: none"> reformat_default_main.ncl reformat_default_func.ncl 	Contains the reformat routines, in two NCL scripts.
reformat_scripts/ECEARTH/ <ul style="list-style-type: none"> reformat_ECEARTH_main.ncl reformat_ECEARTH_func.ncl names_ECEARTH.dat make_lsm3d.sc 	Contains the EC-Earth/NEMO-specific reformat routines.
reformat_scripts/EMAC <ul style="list-style-type: none"> reformat_EMAC_main.ncl reformat_EMAC_func.ncl names_EMAC.dat 	Contains the EMAC-specific reformat routines.
reformat_scripts/fixes/ <ul style="list-style-type: none"> <project>_<model>_fixes.ncl 	Contains the user-defined, project- and model-specific fixes, defined as NCL scripts <project>_<model>_fixes.ncl. A template is also provided for the user to add more fixes.
reformat_scripts/obs/	Contains specific reformat routines for "cmorizing" observational data.
reformat_scripts/constants.ncl	Contains general-purpose functions and procedure, called by the default, the ECEARTH- and the EMAC-specific routines.
reformat_scripts/recognized_units.dat	Provides a list of possible alternative units to the CMOR standard and the corresponding conversion factor. Can be extended by the user.
reformat_scripts/recognized_vars.dat	Provides a list of possible alternative variable names to the CMOR standard names. Can be extended by the user.
reformat_scripts/variable_defs/	Declaration of variables, variable specific attributes and calculation of derived variables

<i>Data folders</i>	
	The data folders are specified in nml/namelist_*, and thus may be different from the defaults given here. These folders contain the output generated by the ESMValTool and are created on the fly if needed. Note that these folders do not need to be in the same directory as the source code. They can be arbitrarily specified in the namelist as path relative to the root path. Using symbolic links is another option to separate the actual data from the code.
climo/	Quality checked and derived netCDF files, reformatted from the original data.
plots/	Destination directory for graphics files.
work/	Miscellaneous files produced during run-time, e.g., optional netCDF output and references/acknowledgements.

Workflow of reformat routines

Control flow of reformat_default

The `reformat_default_main.ncl` script sets the global variables as defined in `reformat.py` (input and output paths, variable name and field, model name and ensemble, etc.) and then performs a list of operations calling various functions and procedures defined in `reformat_default_func.ncl`. The workflow is as follows:

- find grid type: the data can be defined on a standard rectilinear grid or on an irregular grid. In the latter case, the script does not modify the grid properties and additionally attaches the area field (the area weights) for the irregular grid to the output file. The location of the area file is typically defined as an entry in the namelist, for example by using the project class `CMIP5_gridfile` where the final entry is the full path to the area file, see [Table 5.2](#).
- read variable: the selected variable is read from the input file. If the variable is not found, the reading function checks for possible alternative variable names (as specified in `recognized_vars.dat`), before issuing an error message.
- apply project- and model-specific fixes: if a fixing procedure is found in the `fixes/` directory for the selected project and model, it is called at this point in order to apply the user-defined corrections to the data.
- create time-series: the variable is read for the selected time range (`start_year-end_year`) and a time-series is created.
- rank/field consistency: the consistency of variable's rank with the given field (T3M, T2Mz, T2Ms, etc.). A simple calculation of the zonal mean is performed in case a rank 4 variable is provided with T2?z field.
- check fill values: a default missing values is assigned if the variable does not have one. The function then looks for data values that might represent undefined missing values. Currently it considers: -999., -9999., -99999., -999999., 1.e15, -1.e34. Finally, the ESMValTool default missing values (1.e20) is assigned as a standard `_FillValue` to the variable.
- reformat time coordinate: the time coordinate is reformatted according to the CMOR standard. If a calendar attribute is not assigned, the standard is used. The consistency of the time-series with the selected time range is checked.
- reformat vertical coordinate (applies only to certain fields and to rectilinear grids): the vertical coordinate is assigned "Pa" units, converting from the most common alternative units (mbar, bar, hPa) if required. The ordering is set from top to bottom (monotonically decreasing).
- reformat latitude coordinate (applies only to certain fields and to rectilinear grids): the ordering is set from South to North (monotonically increasing).

- reformat longitude coordinate (applies only to certain fields and to rectilinear grids): the ordering is set from 0 to 360 degrees.
- check units: consistency of the variable's units with the CMOR standard is checked. The CMOR table for the selected variable must be available in the CMOR/ directory (an error message is issued otherwise). Units renaming and conversion can also be performed, if the corresponding information is given in `recognized_units.dat`.
- set variable attributes: the CMOR standard attributes are assigned to the selected variable. The corresponding CMOR table must be available in the CMOR/ directory (an error message is issued otherwise).
- write output file: the variable reformatted according to the CMOR standard is written in the selected output file.
- add info for irregular grids (applies only to irregular grids): the area file of the irregular grid is added, this file may later be used for averaging.

Control flow of reformat_ECEARTH

This reformat procedure can be used to convert raw EC-Earth/NEMO files to a format that complies with the ESMValTool requirements. It performs the following additional operations compared with the default workflow:

1. `find_name`: translate the EC-Earth/NEMO name to a CMOR name using the table `names_ECEARTH.dat`.
2. `sub_staggergrid`: determine grid type (T, U, V) and add that information to the filename.
3. `mask_land`: land points have the value 0 in the raw files, not a fill value (missing value). This routine sets land points (as in the landmask file) to fill values.
4. `rename_time`: rename time variable from EC-Earth name to standard name and remove the attribute `_Fill-Value`.
5. `rename_lev`: vertical coordinate name in raw files depends on grid, rename it to `lev`. Requires the external input table `names_ECEARTH.dat`.
6. `add_ijcoordvars`: add `i` and `j` variables and assign them as coordinate variables.
7. `convert_units`: unit conversions that cannot be handled by `check_units`.
8. `add_ECEARTHgrid_info`: add ECEARTH grid info (lat, lon, areacello and grid sizes) to the output.

Control flow of reformat_EMAC

The workflow is similar to the default case, but some additional operations specific to the EMAC model are performed in addition:

9. `find_messy_version`: the MESSy version with which the EMAC output has been generated is read from the data.
10. `find_EMAC_name`: the EMAC name of the selected variable is found from the table in `names_EMAC.dat` (an error message is issued if not defined). For complex variables (i.e., variables not directly available as EMAC output but derivable from other EMAC variables), a user-defined recipe can be provided in `reformat_scripts/EMAC/recipes/EMAC_recipes_xxx.ncl` to derive it.
11. `check_field_consistency`: reads from `names_EMAC.dat` file the list of allowed fields for the selected variable (for example is not possible to select total column ozone `toz` as `T3M` field).
12. `check_vertical_integration_type` (only for `T2?s` types): reads from `names_EMAC.dat` the option for the vertical coordinate (C for column integration, S for surface value).
13. `start_the_time_loop`: the EMAC output is assumed to be monthly-aggregated (monthly averages are optional). The data are read starting from January of the `start_year` to December of the `end_year`.
14. `extract_variable`: the selected variable is searched in the EMAC output. If multiple files for a given month/year combination contain the selected variable, the following priority list is applied: time coordinate matching the field type (monthly mean or daily output), data from `tracer_gp` and `tr_*` streams/channels, first file in the list. For complex variables, the corresponding user-defined recipe is called (`reformat_scripts/EMAC/recipes/EMAC_recipes_xxx.ncl`). For `T2?z` types, the data are interpolated on constant pressure levels (defined in `reformat_scripts/constants.ncl`).

15. create time series: within the time loop, a time series start_year-end_year is created.
16. reformat coordinates, check units, set variable attributes and write output: these operations are applied exactly as in the default case.

The user can extend the reformat_scripts/EMAC/recipes/EMAC_recipes_XXX.ncl in order to calculate additional (derived) variables not directly available in EMAC.

Part VI

Annex B - environment

16.1 Basics

The source code of the ESMValTool is hosted on GitHub. The following description gives an overview of the typical workflow and usage for implementing new diagnostics or technical changes into the ESMValTool. For general information on Git, see e.g. the online documentation at <https://www.git-scm.com/doc>.

There are *two* ESMValTool GitHub repositories available:

1. The **PUBLIC** GitHub repository is open to the public. The ESMValTool is released as open-source software under the Apache License 2.0. Use of the software constitutes acceptance of this license and terms. The PUBLIC ESMValTool repository is located at <https://github.com/ESMValGroup/ESMValTool>
2. The **PRIVATE** GitHub repository is restricted to the ESMValTool Development Team. This repository is only accessible to ESMValTool developers that have accepted the terms of use for the ESMValTool development environment. The use of the ESMValTool software and access to the private ESMValTool GitHub repository constitutes acceptance of these terms. *When you fork or copy this repository, you must ensure that you do not copy the PRIVATE repository into an open domain!* The PRIVATE ESMValTool repository for the ESMValTool development team is located at <https://github.com/ESMValGroup/ESMValTool-private>

All developments can be made in either of the two repositories. The creation of *FEATURE BRANCHES* (see below), however, is restricted to registered ESMValTool developers in both repositories. We encourage all developers to join the ESMValTool development team. Please contact the ESMValTool Core Development Team (Section 11.2) if you want to join the ESMValTool development team. The PRIVATE GitHub repository offers a central protected environment for ESMValTool developers who would like to keep their contributions undisclosed (e.g., unpublished scientific work, work in progress by PhD students) while at the same time benefiting from the possibilities of collaborating with other ESMValTool developers and having a backup of their work. *FEATURE BRANCHES* created in the PRIVATE repository are only visible to the ESMValTool development team but not to the public. The concept of a PRIVATE repository has proven to be very useful to efficiently share code during the development across institutions and projects in a common repository without having the contributions immediately accessible to the public.

Both, the PUBLIC and the PRIVATE repository, contain the following kinds of branches:

- *MASTER BRANCH* (official releases),
- *DEVELOPMENT BRANCH* (includes approved new contributions but version is not yet fully tested),
- *FEATURE BRANCH* (development branches for new features and diagnostics created by developers, the naming convention for *FEATURE BRANCHES* is <Project>_<myfeature>).

16.2 Access rights

- Write access to the *MASTER* and *DEVELOPMENT BRANCH* in both, the *PUBLIC* and the *PRIVATE* GitHub repositories, is restricted to the ESMValTool core development team.
- *FEATURE BRANCHES* in both the *PUBLIC* and the *PRIVATE* repository can be created by all members of the ESMValTool development team (i.e. members in the GitHub organization “ESMValGroup”). If needed, branches can be individually write-protected within each repository so that other developers cannot accidentally push changes to these branches.

The *MASTER BRANCH* of the *PRIVATE* repository will be regularly synchronized with the *MASTER BRANCH* of the *PUBLIC* repository (currently by hand). This ensures that they are identical at all times (see schematic in Figure 1). The recommended workflow for members of the ESMValTool development team is to create additional *FEATURE BRANCHES* in either the *PUBLIC* or the *PRIVATE* repository, see further instructions below.

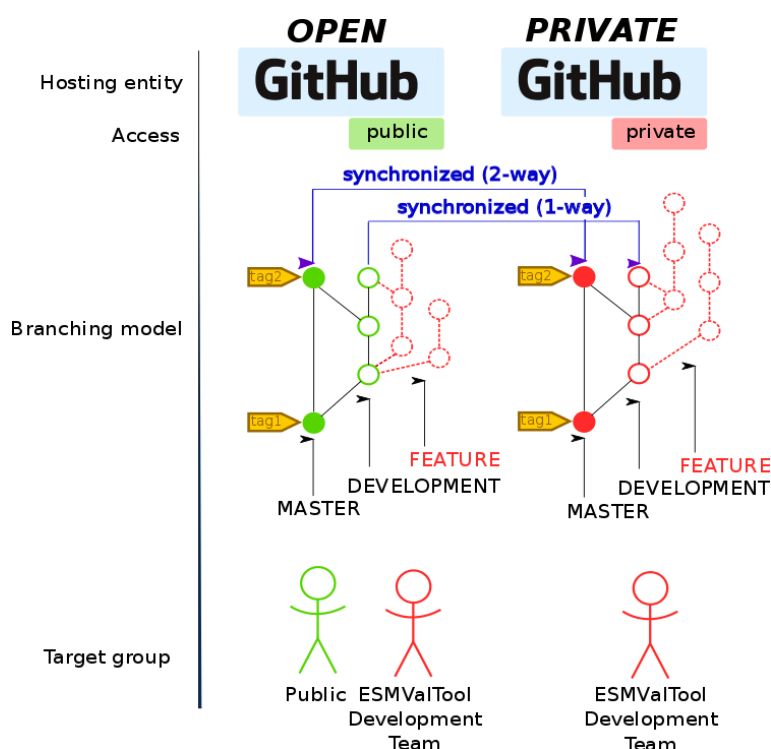


Figure 1: Schematic diagram of the ESMValTool GitHub repositories.

16.3 Workflow

The following description gives an overview of the typical workflow and usage for implementing new diagnostics or technical changes into the ESMValTool. The description assumes that your local development machine is running a Unix-like operating system. For a general introduction to Git tutorials such as, for instance, <https://www.git-scm.com/docs/gittutorial> are recommended.

16.3.1 Getting started

First make sure that you have Git installed on your development machine. On shared machines, software is usually installed using the environment modules. Try e.g.

```
module avail git
```

if this is the case. You can ask your system administrator for assistance. You can test this with the command:

```
git --version
```

In order to properly identify your contributions to the ESMValTool you need to configure your local Git with some personal data. This can be done with the following commands:

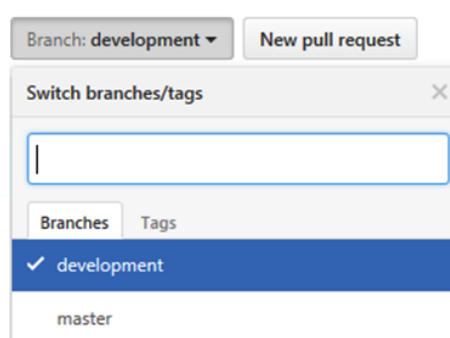
```
git config --global user.name "YOUR NAME"
git config --global user.email "YOUR EMAIL"
```

Note: For working on GitHub you need to create an account and login to <https://github.com/>.

16.3.2 Working with the ESMValTool GitHub Repositories

As a member of the ESMValTool development team you can create *FEATURE BRANCHES* in the PUBLIC as well as in the PRIVATE repository. We encourage all ESMValTool developers to use the following workflow for long-lived developments (>2 weeks).

- Login to GitHub.com
- On GitHub, go to the website of the ESMValTool repository (<https://github.com/ESMValGroup/ESMValTool-private> or <https://github.com/ESMValGroup/ESMValTool>)
- Click on the button create *FEATURE BRANCH*
- Select the “*DEVELOPMENT*” *BRANCH* and create a new *FEATURE BRANCH* for the diagnostic/feature you want to implement. Please follow the following naming convention for your new *FEATURE BRANCH*: <Project>_<myfeature>.



- Click the button “Clone or Download” and copy the URL shown there
- Open a terminal window and go to the folder where you would like to store your local copy of the ESMValTool source
- Type git clone, and paste the URL:

```
git clone <URL_FROM_CLIPBOARD>
```

This will clone the ESMValTool repository at GitHub to a local folder. You can now query the status of your local working copy with:

```
git status
```

You will see that you are on a branch called master and your local working copy is up to date with the remote repository. With

```
git branch --all
```

you can list all available remote and local branches. Now switch to your feature branch by:

```
git checkout <NAME_OF_YOUR_FEATURE_BRANCH>
```

You can now start coding. To check your current developments you can use the command

```
git status
```

You can add new files and folders that you want to have tracked by Git using:

```
git add <NEW_FILE|FOLDER>
```

Commit your tracked changes to your local working copy via:

```
git commit -m "YOUR COMMIT MESSAGE"
```

You can inspect your changes with (use `man git-log` for all options):

```
git log
```

To share your work and to have an online backup, push your local development to your *FEATURE BRANCH* on GitHub:

```
git push origin <YOUR_FEATURE_BRANCH>
```

Note: An overview on Git commands and best practices can be found e.g. here: <https://zeroturnaround.com/rebellabs/git-commands-and-best-practices-cheat-sheet/>

16.3.3 Pull requests

Once your development is completely finished, go to the GitHub website of the ESMValTool repository and switch to your *FEATURE BRANCH*. You can then initiate a pull request by clicking on the button “New pull request”. Select the *DEVELOPMENT BRANCH* as “base branch” and click on “Create pull request”. Your pull request will then be tested, discussed and implemented into the *DEVELOPMENT BRANCH* by the ESMValTool Core Development Team.

Attention: Before creating a pull request, please make sure all requirements listed in Sections 9 and 10 are fully met (see also checklist in Table 9.11).

16.3.4 GitHub issues

In case you encounter a bug or if you have a feature request or something similar you can open an issue on the PUBLIC ESMValTool GitHub repository.

16.4 General do-s and don't-s

16.4.1 Do-s

- Create a *FEATURE BRANCH* and use exclusively this branch for developing the ESMValTool. The naming convention for *FEATURE BRANCHES* is <Project>_<myfeature>.
- Comment your code as much as possible and in English.
- Use short but self-explanatory variable names (e.g., `model_input` and `reference_input` instead of `xm` and `xr`).

- Consider a modular/functional programming style. This often makes code easier to read and deletes intermediate variables immediately. If possible, separate diagnostic calculations from plotting routines.
- Consider reusing or extending existing code. General-purpose code can be found in `diag_scripts/lib/` and in `plot_scripts/`.
- Comment all switches and parameters including a list of all possible settings/options in the header section of your code (see also Section 9.2).
- Use templates for namelists (Section 5.5) and diagnostics (Section 9.2) to help with proper documentation.
- Keep your *FEATURE BRANCH* regularly synchronized with the *DEVELOPMENT BRANCH* (git merge).
- Keep developments / modifications of the ESMValTool framework / backend / basic structure separate from developments of diagnostics by creating different *FEATURE BRANCHES* for these two kinds of developments. Create *FEATURE BRANCHES* for changes / modifications of the ESMValTool framework only in the *PUBLIC* repository.

16.4.2 Don't-s

- Do not use other programming languages than the ones currently supported (NCL, Python, R). Contact the Core Development Team (Section 11.2) if you wish to use another language, but remember that only open-source languages are supported by the ESMValTool.
- Do not develop without proper version control (see do-s above).
- Avoid large (memory, disk space) intermediate results. Delete intermediate files/variables or see modular/functional programming style.
- Do not use hard-coded pathnames or filenames.
- Do not mix developments / modifications of the ESMValTool framework and developments / modifications of diagnostics in the same *FEATURE BRANCH*.

Part VII

Annex C - namelists

CHAPTER 17

Introduction

The following pages give an overview on the available diagnostics and metrics packages implemented into the ESMValTool v1.1. The overview contains a brief description of the diagnostic/metric, a list of available user settings, information on the observational data used, references, and example plots.

For further information on the diagnostics and performance metrics of ESMValTool v1.0, see also

- Eyring et al., ESMValTool (v1.0) – a community diagnostic and performance metrics tool for routine evaluation of Earth System Models in CMIP, *Geosci. Model Dev.*, 9, 1747-1802, 2016. [doi:10.5194/gmd-9-1747-2016](https://doi.org/10.5194/gmd-9-1747-2016).

Diagnostics and observational data sets newly implemented into the ESMValTool v1.1 are described in

- Lauer et al., Benchmarking CMIP5 models with a subset of ESA CCI Phase 2 data using the ESMValTool, *Remote Sens. Environ.* (2017), [doi:10.1016/j.rse.2017.01.007](https://doi.org/10.1016/j.rse.2017.01.007)

18.1 Overview

The aerosol diagnostics currently implemented allow for three kinds of comparisons:

- *station data*: concentrations of various aerosol species (aerosol sulfate, aerosol nitrate, aerosol ammonium, black carbon, organic carbon, PM_{2.5} and PM₁₀) are compared with observational data from station networks (IMPROVE and CASTNET for North America, EMEP for Europe and EANET for Asia). Aerosol optical depth (AOD) at 550 nm is compared with AERONET data. The comparison between model and observations is performed considering all available observational data in the selected time period, on a monthly-mean basis. The model data is extracted in the grid boxes where the respective observational stations are located (co-located model and observational data). This diagnostic produces three types of plot: time series (model/observations vs. time), scatter plot (model versus observations) and a map plot (model as contour map, observations as dots using the same color coding).
- *satellite data*: AOD at 550 nm is compared with satellite data (MODIS and MISR). This diagnostic produces contour map plots of the AOD of model and observations, as well as difference plot between model and observations.
- *vertical profiles, size distributions*: vertical profiles of aerosol (mass and number) concentrations and of aerosol size distributions are compared to observational data from aircraft campaigns and ground based stations. The model data are extracted based on the campaign/station location (lat-lon box) and time period (on a climatological basis, i.e. selecting the same days/months, but regardless of the year). The basic statistics is calculated on the selected model data (mean, standard deviation, median, 5-10-25-75-90-95 percentiles) and compared to the same quantities from the observations (where available). This diagnostics supports monthly-mean, daily-mean and instantaneous input data, with the latter two providing a more robust evaluation. For this diagnostic, rather specific variables are required (i.e., aerosol number concentration for particles with diameter larger than 14 nm) to match the properties of the instruments used during the campaign. New CMOR variables have been added and corresponding EMAC recipes have been defined.

18.2 Available namelists and diagnostics

Namelists are stored in nml/

- namelist_aerosol_CMIP5.xml
- namelist_aerosol_EMAC.xml

Diagnostics are stored in diag_scripts/

- aerosol_stations.ncl
- aerosol_satellite.ncl
- aerosol_profiles.ncl

18.3 User settings

User setting files (cfg files) are stored in nml/cfg_aerosol/CMIP5/ and nml/cfg_aerosol/EMAC/:

! NOTE ! The cfg files may contain hard coded pathnames (attribute “datapath”, see below) for the observational data that have to be set by the user!

1. aerosol_stations.ncl (AERONET, CASTNET, EANET, EMEP, IMPROVE)

- cfg_aerosol_stations_AERONET.ncl
- cfg_aerosol_stations_CASTNET.ncl
- cfg_aerosol_stations_EANET.ncl
- cfg_aerosol_stations_EMEP.ncl
- cfg_aerosol_stations_IMPROVE.ncl
- cfg_aerosol_tsline.ncl

2. aerosol_satellites.ncl (ESACCI, MISR, MODIS)

- satellite_ESACCI-AEROSOL.ncl
- cfg_aerosol_satellite_MISR.ncl
- cfg_aerosol_satellite_MODIS.ncl

Required diag_script_info attributes

- ref_model: name of reference data set (e.g., “ESACCI-AEROSOL”)
- styleset: style (“DEFAULT”)
- regrid_method: regriding method (coarsest, finest, ref)
- range_option: time range option (0: model own time range, 1: intersection between models)

3. aerosol_profiles.ncl (ACCESS, ACCESS-2, Asmi11, CIRRUS, CONCERT, CR-AVE, DC3, HIPPO, IMPROVE, INCA, LACE, Putaud, SALTRACE, TC4, Texas, UCN-Pacific)

- cfg_aerosol_profiles_ACCESS.ncl
- cfg_aerosol_profiles_ACCESS-2.ncl
- cfg_aerosol_profiles_Asmi11.ncl
- cfg_aerosol_profiles_CIRRUS.ncl
- cfg_aerosol_profiles_CONCERT.ncl
- cfg_aerosol_profiles_CR-AVE.ncl
- cfg_aerosol_profiles_DC3.ncl
- cfg_aerosol_profiles_HIPPO.ncl
- cfg_aerosol_stations_IMPROVE.ncl
- cfg_aerosol_profiles_INCA.ncl
- cfg_aerosol_profiles_LACE.ncl
- cfg_aerosol_profiles_Putaud.ncl

- `cfg_aerosol_profiles_SALTRACE.ncl`
- `cfg_aerosol_profiles_TC4.ncl`
- `cfg_aerosol_profiles_Texas.ncl`
- `cfg_aerosol_profiles_UCN-Pacific.ncl`

Required diag_script_info attributes

- `campaign`: name of measurement campaign (e.g., INCA, LACE, Texas)
- `styleset`: "DEFAULT"
- **`datapath`: path for campaign data**
- `cc_threshold`: cloud cover threshold (%)
- `summary_plot`: create a summary with the median of all flights on a single plot
- `exclude_flights`: exclude specific flights from the summary plot

18.4 Variables

CMIP5

- `sconcbc`, `sconcoa`, `sconco4`
- `od550aer`, `od550lt1aer`, `abs550aer`, `od870aer`

EMAC

- `conccnmode`, `conccnmodeSTP`, `diamcnmode`, `cl`
- `conccnd5`, `conccnd10`, `conccnSTPd3`, `conccnSTPd5`, `conccnSTPd14`, `conccnSTPd120`
- `mmrbc`, `mmraer`, `mmrbcfree`
- `sconco4`, `sconco3`, `sconcnh4`, `sconcbc`, `sconcoa`, `sconco4`, `sconco4`, `sconco4`, `sconco4`, `sconco4`
- `od550aer`

18.5 Observations and reformat scripts

Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

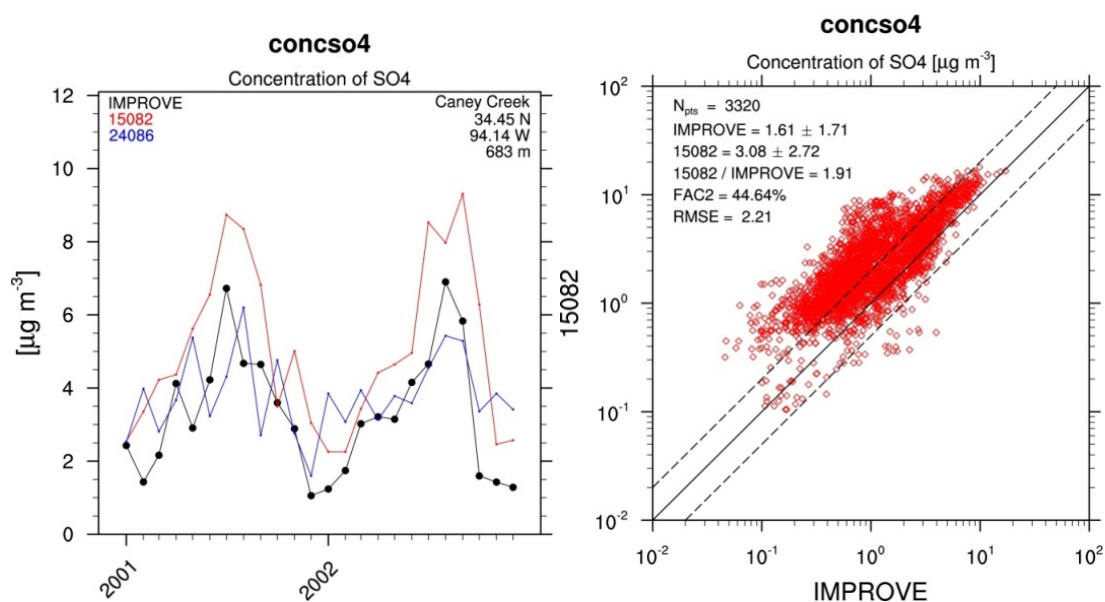
- AERONET (`reformat_scripts/obs/reformat_obs_AERONET.ncl`)
- CASTNET (`reformat_scripts/obs/reformat_obs_CASTNET.ncl`)
- EANET (`reformat_scripts/obs/reformat_obs_EANET.ncl`)
- EMEP (`reformat_scripts/obs/reformat_obs_EMEP.ncl`)
- IMPROVE (`reformat_scripts/obs/reformat_obs_IMPROVE.ncl`)
- MODIS (`obs4mips`)
- MISR (`obs4mips`)
- ESACCI-AEROSOL (`reformat_scripts/obs/reformat_obs_ESACCI-AEROSOL.ncl`)
- ACCESS (`reformat_scripts/obs/reformat_obs_ACCESS.ncl`)
- ACCESS-2 (`reformat_scripts/obs/reformat_obs_ACCESS-2.ncl`)
- Asmi11 (`reformat_scripts/obs/reformat_obs_Asmi11.ncl`)
- CIRRUS (`reformat_scripts/obs/reformat_obs_CIRRUS.ncl`)

- CONCERT (reformat_scripts/obs/reformat_obs_CONCERT.ncl)
- CR-AVE (reformat_scripts/obs/reformat_obs_CR-AVE.ncl)
- DC3 (reformat_scripts/obs/reformat_obs_DC3.ncl)
- HIPPO (reformat_scripts/obs/reformat_obs_HIPPO.ncl)
- INCA (reformat_scripts/obs/reformat_obs_INCA.ncl)
- LACE (reformat_scripts/obs/reformat_obs_LACE.ncl)
- Putaud (reformat_scripts/obs/reformat_obs_Putaud.ncl)
- SALTRACE (reformat_scripts/obs/reformat_obs_SALTRACE.ncl)
- TC4 (reformat_scripts/obs/reformat_obs_TC4.ncl)
- Texas (reformat_scripts/obs/reformat_obs_Texas.ncl)
- UCN-Pacific (reformat_scripts/obs/reformat_obs_UCN-Pacific.ncl)

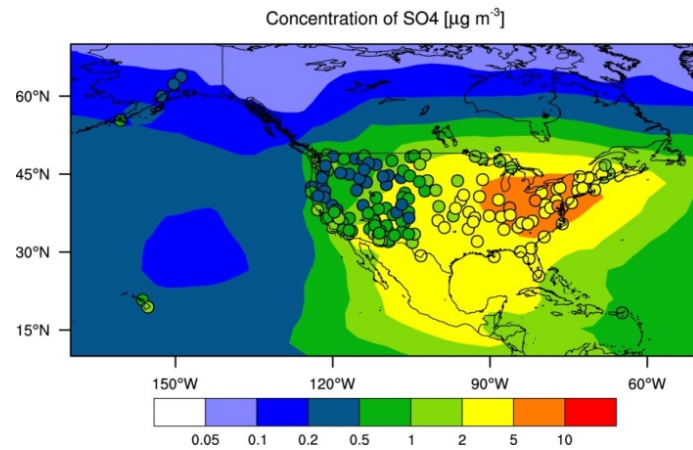
18.6 References

- Aquila, V. et al., MADE-in: a new aerosol microphysics submodel for global simulation of insoluble particles and their mixing state. *Geosci. Model Dev.* 4 (2), 325-355 (2011).
- Lauer, A. et al., Simulating aerosol microphysics with the EMAC/MADE GCM - Part I: Model description and comparisons with observations. *Atmos. Chem. Phys.* 5 (12), 3251-3276 (2005).
- Righi, M. et al., The global impact of the transport sectors on atmospheric aerosol: Simulations for year 2000 emissions. *Atmos. Chem. Phys.* 13 (19), 9939-9970 (2013).

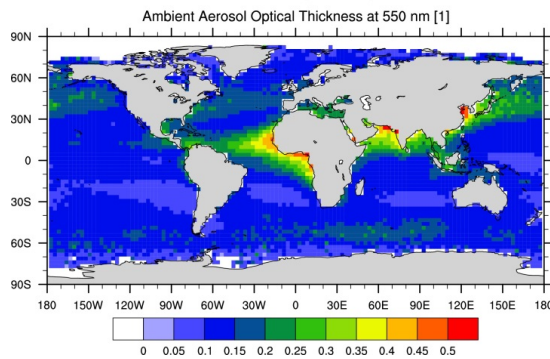
18.7 Example plots



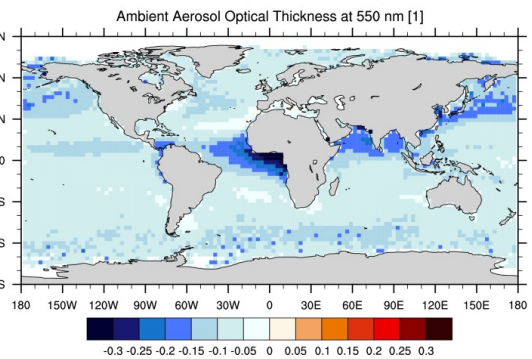
concs04 (24086 - IMPROVE)



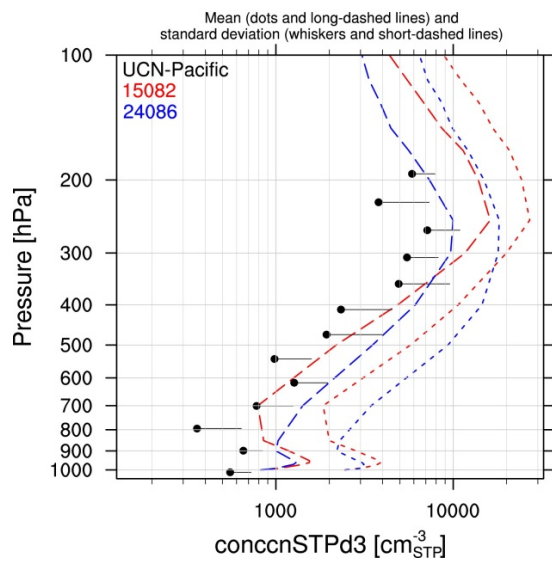
od550aer (MODIS)



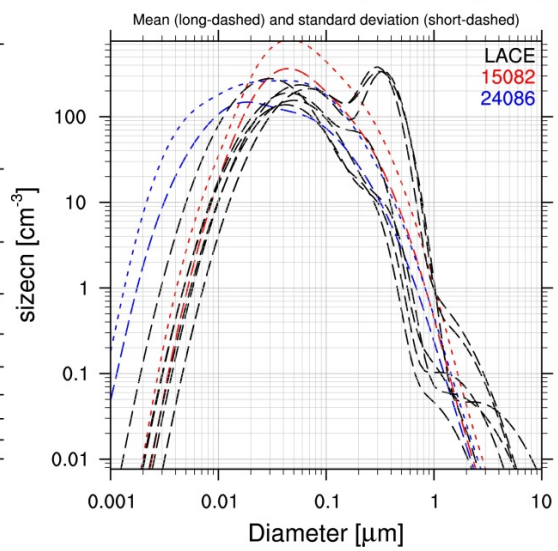
od550aer (24086 - MODIS)



UCN-Pacific - 20N-70N



LACE - Lower Free Troposphere (4 km)



AutoAssess Radiation RMS errors

19.1 Overview

The goal of this effort is to code up routines of calculate RMS errors in AutoAssess Radiation and draw a summary plot, so that errors in two models or versions simulations can be compared.

19.2 Available Namelists and Diagnostics

Namelists are stored in nml/

- namelist_AutoAssess_radiation_rms_Amon_all.xml
- namelist_AutoAssess_radiation_rms_cfMon_all.xml

Diagnostics are stored in diag_scripts/

AutoAssess radiation calculates the spatial RMS errors of diagnostics of radiation budget, cloud and hydrological cycle. Specifically,

- TOA Net Incoming Total Radiation
- TOA Net Incoming Shortwave Radiation
- TOA Outgoing Shortwave Radiation
- TOA Outgoing Longwave Radiation
- TOA Outgoing Clear-Sky Shortwave Radiation
- TOA Outgoing Clear-Sky Longwave Radiation
- TOA Shortwave Cloud Radiative Effect
- TOA Longwave Cloud Radiative Effect
- Surface Net Downwelling Shortwave Radiation
- Surface Net Downwelling Longwave Radiation
- Surface Downwelling Clear-Sky Longwave Radiation
- ISCCP High Level Intermediate Cloud Fraction

- ISCCP High Level Thick Cloud Fraction
- ISCCP Middle Level Intermediate Cloud Fraction
- ISCCP Middle Level Thick Cloud Fraction
- ISCCP Low Level Intermediate Cloud Fraction
- ISCCP Low Level Thick Cloud Fraction
- Water Vapor Path
- Precipitation

Following variables are defined so that they can be evaluated after their COSP outputs become available.

- CCI Total Cloud Fraction
- CCI High Level Intermediate Cloud Fraction
- CCI High Level Thick Cloud Fraction
- CCI Middle Level Intermediate Cloud Fraction
- CCI Middle Level Thick Cloud Fraction
- CCI Low Level Intermediate Cloud Fraction
- CCI Low Level Thick Cloud Fraction
- CCI Total Liquid Cloud Area Fraction
- CCI Mean Cloud Top Pressure

The main plot is a summary plot which shows normalized RMS errors of different diagnostics in a model (version) relative to those in another model (version).

19.3 Specific Routines

See the individual namelists for specific routines.

Summary Plot

A python program which draw a summary plot: `plot_scripts/python/plot_norm_ac_ideal.py`

A script to run the program: `plot_scripts/python/AutoAssess_radiation_summary/Plot_AutoAssess_radiation_summary.csh`
(Please follow instructions written in the script to obtain a summary file with RMS errors for multi variables.)

Observational uncertainty can be calculated using the second reference data. See below how to calculate observational uncertainty for other variables.

Observational Uncertainty

Observational uncertainty is defined by the RMS difference of two reference dataset. Area mean of this RMS difference can be calculated by using the `AutoAssess_radiation` programs, giving two reference data as input to the program. This value can be plotted as an uncertainty bar for each variable in the summary plot. As an example, the file `radiation_obs2_cmipname_ANN.csv.example` contains the RMS differences calculated with the example namelist `nml/namelist_AutoAssess_radiation_rms_Amon_all_obs.xml`.

The following steps are required to obtain a summary plot with observational uncertainty bars:

- Calculate the RMS difference of two reference dataset
 - An example namelist file: `nml/namelist_AutoAssess_radiation_rms_Amon_all_obs.xml`
- You get csv files for each diagnostics with the RMS difference value in each file: `summary_global_<observational data name for the diagnostic>.csv` (e.g. `summary_global_CERES-EBAF.csv`)

- Create a file 'radiation_obs2_cmipname_ANN.csv' under the directory 'plot_scripts/python/AutoAssess_radiation_summary' and add the diagnostics names with their RMS difference values
 - There is a template file:
plot_scripts/python/AutoAssess_radiation_summary/radiation_obs2_cmipname_ANN.csv.template
 - Edit the file (with your favourite editor) by replacing the undefined value to the calculated RMS difference value
 - If the RMS difference value is not available, please set undefined value '-10000', instead
- Run Plot_AutoAssess_radiation_summary.csh

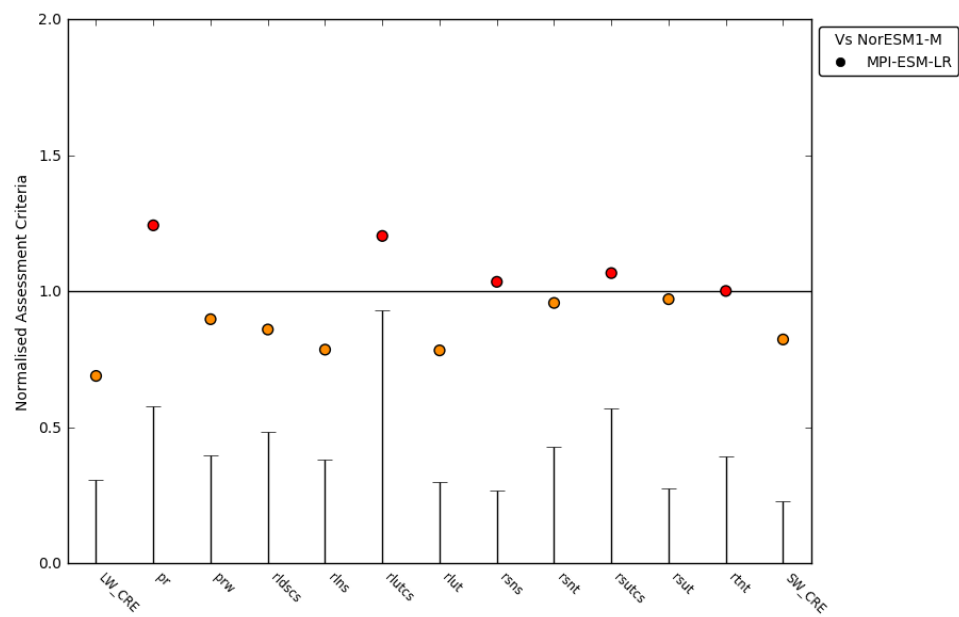
19.4 Variables

- rlut (atmos, monthly mean, longitude, latitude, time)
- rlutcs (atmos, monthly mean, longitude, latitude, time)
- rsut (atmos, monthly mean, longitude, latitude, time)
- rsutcs (atmos, monthly mean, longitude, latitude, time)
- rsdt (atmos, monthly mean, longitude, latitude, time)
- rsds (atmos, monthly mean, longitude, latitude, time)
- rsus (atmos, monthly mean, longitude, latitude, time)
- rlds (atmos, monthly mean, longitude, latitude, time)
- rlus (atmos, monthly mean, longitude, latitude, time)
- rldscs (atmos, monthly mean, longitude, latitude, time)
- prw (atmos, monthly mean, longitude, latitude, time)
- pr (atmos, monthly mean, longitude, latitude, time)
- clisccp (atmos, monthly mean, longitude, latitude, time)

19.5 Observations and Scripts

- Reference data (and the second reference data which are currently available in a readable netCDF format to the program to estimate the observational uncertainty)
- obs4mips CERES-EBAF L3B Ed2-8 (CERES CERES_SYN1deg-Month Terra-Aqua-MODIS Ed3A)
- obs4mips SSMI L3 RSS (obs4mips SSMI-MERIS L3 V1-00)
- obs4mips GPCP-SG L3 v2.2
- ana4mips JRA-55 Amon reanalysis
- obs4mips ISCCP_L3_V1.0
- Global mean of the RMS difference of variables are in
scripts/python/AutoAssess_radiation_summary/radiation_obs2_cmipname_ANN.csv.new

19.6 Example Plots



20.1 Overview

This diagnostic calculates biases of long-term climatological annual means of total runoff, precipitation and evapotranspiration for 12 large-scale catchments on different continents and climates. For total runoff, catchment averaged model values are compared to climatological GRDC station observations of river runoff (Dümenil Gates et al., 2000). Due to the incompleteness of these station data, a year-to-year correspondence of data cannot be achieved in a generalized way, so that only climatological data are considered, such it has been done in Hagemann et al. (2013). For precipitation, catchment-averaged WFDEI precipitation (Weedon et al., 2014) data from 1979-2010 is used as reference. For evapotranspiration, observations are estimated using the difference of the above mentioned precipitation reference minus the climatological GRDC river runoff.

20.2 Available namelists and diagnostics

Namelists are stored in nml/

- namelist_runoff_et.xml

Diagnostics are stored in diag_scripts/

- catchment_analysis_val.py

20.3 User settings

None.

20.4 Variables

- mrro (land, annual mean, longitude latitude time)
- evspsbl (atmos, annual mean, longitude latitude time)
- pr (atmos, annual mean, longitude latitude time)

20.5 Observations and reformat scripts

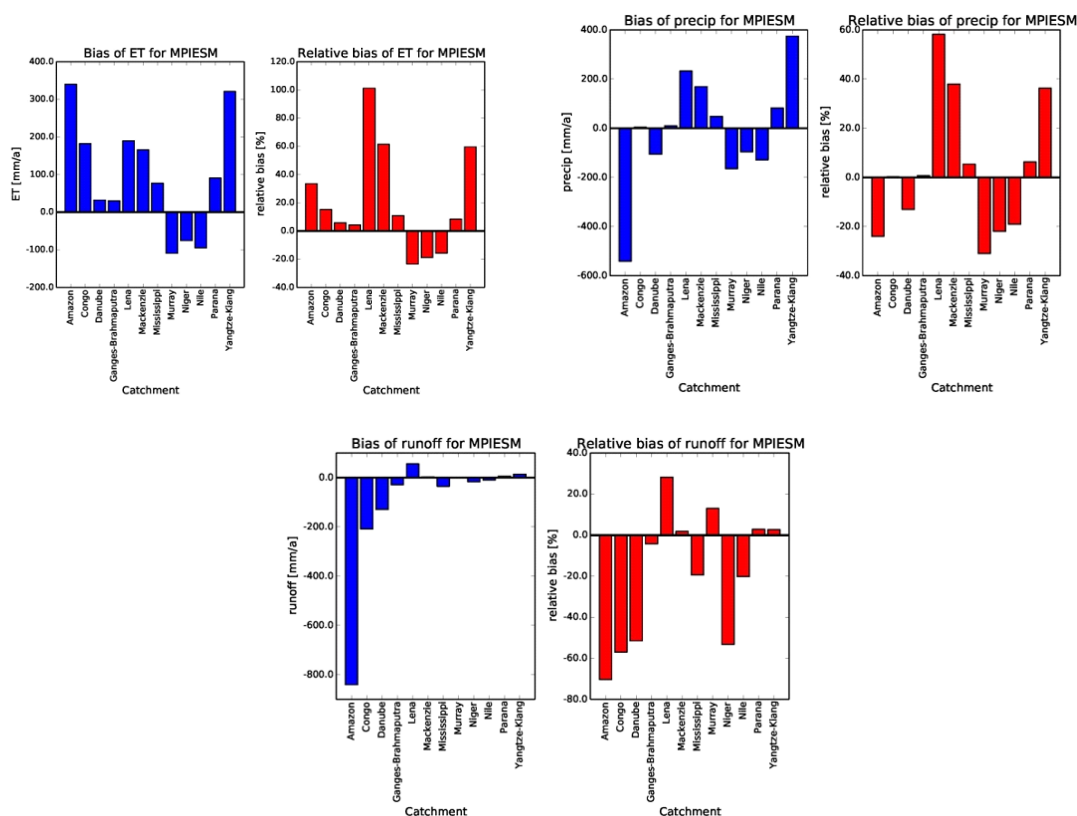
The observational data (big_catchments.nc) are provided with the ESMValTool and are located in `diag_scripts/aux/catchment_analysis/`.

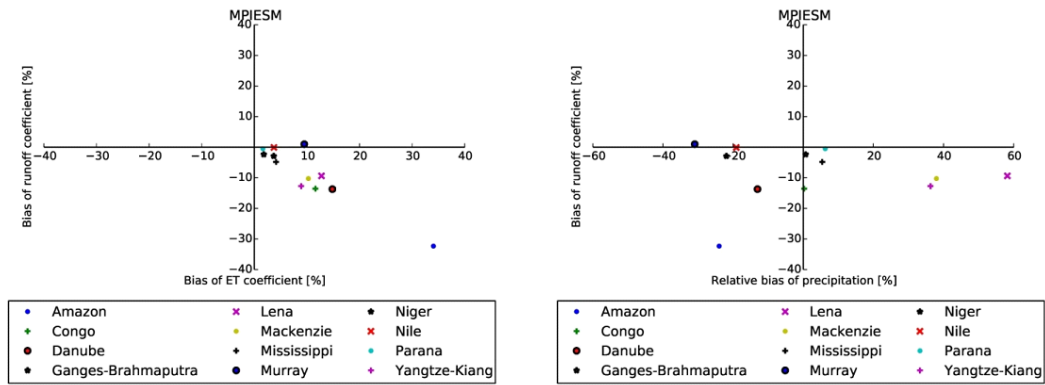
- Climatological river runoff from various GRDC stations
- WFDEI precipitation (Weedon et al., 2014).

20.6 References

- Dümenil Gates, L., S. Hagemann and C. Golz, 2000: Observed historical discharge data from major rivers for climate model validation. Max Planck Institute for Meteorology Report 307, Hamburg, Germany.
- Hagemann, S., A. Loew, A. Andersson, 2013: Combined evaluation of MPI-ESM land surface water and energy fluxes. *J. Adv. Model. Earth Syst.*, 5: 259-286, doi: 10.1029/2012MS000173.
- Weedon, G. P., G. Balsamo, N. Bellouin, S. Gomes, M. J. Best, and P. Viterbo (2014), The WFDEI meteorological forcing data set: WATCH Forcing Data methodology applied to ERA-Interim reanalysis data, *Water Resour. Res.*, 50, 75057514, doi: 10.1002/2014WR015638.

20.7 Example plots





21.1 Overview

The namelist `namelist_lauer13jclim.xml` computes the climatology and interannual variability of climate relevant cloud variables such as cloud radiative forcing (CRE), liquid (lwp) and ice water path (iwp), cloud amount (clt), and total precipitation (pr) reproducing some of the evaluation results of Lauer and Hamilton (2013). The standard namelist includes a comparison of the geographical distribution of multi-year average cloud parameters from individual models and the multi-model mean with satellite observations. Taylor diagrams are generated that show the multi-year annual or seasonal average performance of individual models and the multi-model mean in reproducing satellite observations. The diagnostic also facilitates the assessment of the bias of the multi-model mean and zonal averages of individual models compared with satellite observations. Interannual variability is estimated as the relative temporal standard deviation from multi-year timeseries of data with the temporal standard deviations calculated from monthly anomalies after subtracting the climatological mean seasonal cycle.

21.2 Available namelists and diagnostics

Namelists are stored in `nml/`

- `namelist_lauer13jclim.xml`

Diagnostics are stored in `diag_scripts/`

- `clouds.ncl`: global maps of (multi-year) annual means including multi-model mean
- `clouds_bias.ncl`: global maps of the multi-model mean and the multi-model mean bias
- `clouds_interannual`: global maps of the interannual variability
- `clouds_isccp`: global maps of multi-model mean minus observations + zonal averages of individual models, multi-model mean and observations
- `clouds_taylor.ncl`: taylor diagrams

21.3 User settings

User setting files (cfg files) are stored in `nml/cfg_cloud/`

1. clouds.ncl: cfg_clouds.ncl

Required diag_script_info attributes

- grid: grid for regridding (coarsest, finest, ref, 1°x1°) in case calculation of the multi-model mean or difference plots are requested
- ref_model: name of reference data set

Optional diag_script_info attributes

- calcmm: calculate multi-model mean (True, False)
- embracesetup: True = 2 plots per line, False = 4 plots per line (default)
- showdiff: calculate and plot differences (True, False)
- timemean: time averaging, i.e. “seasonal” (DJF, MAM, JJA, SON), “annual” (annual mean)

Required variable_info attributes

- long_name: description of variable
- map_Levels: contour levels for plotting
- units: variable units

Color tables

- variable “lwp”: diag_scripts/lib/ncl/rgb/qcm3.rgb

2. clouds_bias.ncl: cfg_clouds_bias.ncl

Required diag_script_info attributes

- grid: grid for regridding (coarsest, finest, ref) in case calculation of the multi-model mean is requested
- ref_model: name of reference data set
- valid_fraction: minimum fraction of valid data points (for creating a mask)

Optional diag_script_info attributes

- ncdf: enable to output to netCDF; either use “default” or give a full file name
- projection: map projection, e.g., Mollweide, Mercator (see <http://www.ncl.ucar.edu/Document/Graphics/Resources/mp.shtml#mpProjection> for available projections)
- timemean: time averaging, i.e. “seasonalclim” (DJF, MAM, JJA, SON), “annualclim” (annual mean)

Required variable_info attributes

- long_name: description of variable
- map_diff_Levels: contour levels for difference plot
- units: variable units

Color tables

- variable “tas”: diag_scripts/lib/ncl/rgb/ipcc-tas.rgb, diag_scripts/lib/ncl/rgb/ipcc-tas-delta.rgb
- variable “pr-mmday”: diag_scripts/lib/ncl/rgb/ipcc-precip.rgb, diag_scripts/lib/ncl/rgb/ipcc-precip-delta.rgb

3. clouds_interannual.ncl: cfg_clouds_interannual.ncl

Required diag_script_info attributes

- colormap: e.g., WhiteBlueGreenYellowRed, rainbow
- grid: grid for regridding (coarsest, finest, ref, 1°x1°) in case calculation of the multi-model mean is requested
- models_to_skip: name(s) of data set(s) to be skipped when calculating the multi-model mean

- `projection`: map projection, e.g., Mollweide, Mercator (see <http://www.ncl.ucar.edu/Document/Graphics/Resources/mp.shtml#mpProjection> for available projections)
- `ref_model`: specifies model that should be taken as “reference” when regridding to “ref” (only required if `grid = “ref”`)

Optional diag_script_info attributes

- `calcm`: calculate multi-model mean (True, False)
- `extrafiles`: write plots for individual models to separate files (True, False)

Color tables

- variable “lwp”: `diag_scripts/lib/ncl/rgb/qcm3.rgb`

4. `clouds_ipcc.ncl: cfg_clouds_ipcc.ncl`

Required diag_script_info attributes

- `grid`: grid for regridding (coarsest, finest, ref, 1x1)
- `ref_model`: name of reference data set
- `valid_fraction`: minimum fraction of valid data points (for creating a mask)
- `projection`: map projection, e.g., Mollweide, Mercator (see <http://www.ncl.ucar.edu/Document/Graphics/Resources/mp.shtml#mpProjection> for available projections)

Optional diag_script_info attributes

- `mask_ts_sea_ice`: True = mask $T < 272$ K as sea ice (only for variable “ts”), False = no additional grid cells masked for variable “ts”
- `models_to_skip`: name(s) of data set(s) to be skipped when calculating the multi-model mean (usually all observations)
- `ncdf`: enable to output to netCDF; either use “default” or give a full file name
- `styleset`: “CMIP5”, “DEFAULT”
- `timemean`: time averaging, i.e. “seasonalclim” (DJF, MAM, JJA, SON), “annualclim” (annual mean)

Required variable_info attributes

- `long_name`: description of variable
- `map_Levels`: contour levels for plotting
- `units`: variable units

Color tables

- variables “pr”, “pr-mmday”: `diag_scripts/lib/ncl/rgb/ipcc-precip-delta.rgb`

5. `clouds_taylor.ncl: cfg_clouds_taylor.ncl`

Required diag_script_info attributes

- `grid`: grid for regridding (coarsest, finest, ref, 1x1)
- `valid_fraction`: minimum fraction of valid data points (for creating a mask)

Optional diag_script_info attributes

- `calcm`: calculate multi-model mean (True, False)
- `models_to_skip`: name(s) of data set(s) to be skipped when calculating the multi-model mean (usually all observations)
- `embracelegend`: False (default) = include legend in plot, max. 2 columns with model names in legend; True = write extra file with legend, max. 7 model names per column in legend, alternative observational dataset(s) will be plotted as a red star and labeled “altern. ref. dataset” in legend (only if dataset is of class “OBS”)

- `estimate_obs_uncertainty`: True = estimate observational uncertainties from mean values (assuming fractions of obs. RMSE from documentation of the obs data); only available for “CERES-EBAF”, “MODIS”, “MODIS-L3”; False = do not estimate obs. uncertainties from mean values
- `mask_ts_sea_ice`: True = mask $T < 272$ K as sea ice (only for variable “ts”); False = no additional grid cells masked for variable “ts”
- `styleset`: “CMIP5”, “DEFAULT” (if not set, CLOUDS_TAYLOR will create a color table and symbols for plotting)
- `ref_model`: name of reference data set (if this attribute is not set the variable attribute “ref_model” defined in the namelist is used; see below); note: if neither `diag_script_info@ref_model` nor the variable attribute `ref_model` is set, an error message is issued
- `timemean`: time averaging, i.e. “seasonalclim” (DJF, MAM, JJA, SON), “annualclim” (annual mean)

Optional variable attributes (defined in namelist)

- `ref_model`: name of reference data set; note: if `diag_script_info@ref_model` is defined, `diag_script_info@ref_model` will be used and the variable attribute `ref_model` will be ignored

21.4 Variables

- `clwvi` (atmos, monthly mean, longitude latitude time)
- `clivi` (atmos, monthly mean, longitude latitude time)
- `clt` (atmos, monthly mean, longitude latitude time)
- `pr` (atmos, monthly mean, longitude latitude time)
- `rlut`, `rlutcs` (atmos, monthly mean, longitude latitude time)
- `rsut`, `rsutcs` (atmos, monthly mean, longitude latitude time)

21.5 Observations and reformat scripts

Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

- CERES-EBAF (obs4mips) - CERES TOA radiation fluxes (used for calculation of cloud forcing)
- GPCP-SG (obs4mips) - Global Precipitation Climatology Project total precipitation
- MODIS (obs4mips) - MODIS total cloud fraction
- MODIS-CFMIP - MODIS ice water path climatology, originally created for CFMIP, references: King et al. (2003), IEEE Trans. Geosci. Remote Sens.; Pincus et al. (2012), J. Clim.

Reformat script: `reformat_scripts/obs/reformat_obs_MODIS-CFMIP.ncl`

- UWisc - University of Wisconsin-Madison liquid water path climatology, based on satellite observations from TMI, SSM/I, and AMSR-E, reference: O'Dell et al. (2008), J. Clim.

Reformat script: `reformat_scripts/obs/reformat_obs_UWisc.ncl`

21.6 References

- King, M. D., et al. (2003), Cloud and aerosol properties, precipitable water, and profiles of temperature and water vapor from MODIS, IEEE Trans. Geosci. Remote Sens., 41, 442-458, doi: 10.1109/TGRS.2002.808226.

- Lauer A., and K. Hamilton (2013), Simulating clouds with global climate models: A comparison of CMIP5 results with CMIP3 and satellite data, *J. Clim.*, 26, 3823-3845, doi: 10.1175/JCLI-D-12-00451.1.
- O'Dell, C. W., F. J. Wentz, and R. Bennartz (2008), Cloud liquid water path from satellite-based passive microwave observations: A new climatology over the global oceans, *J. Clim.*, 21, 1721-1739, doi:10.1175/2007JCLI1958.1.
- Pincus, R., S. Platnick, S. A. Ackerman, R. S. Hemler, Robert J. Patrick Hofmann (2012), Reconciling simulated and observed views of clouds: MODIS, ISCCP, and the limits of instrument simulators. *J. Climate*, 25, 4699-4720, doi: 10.1175/JCLI-D-11-00267.1.

21.7 Example plots

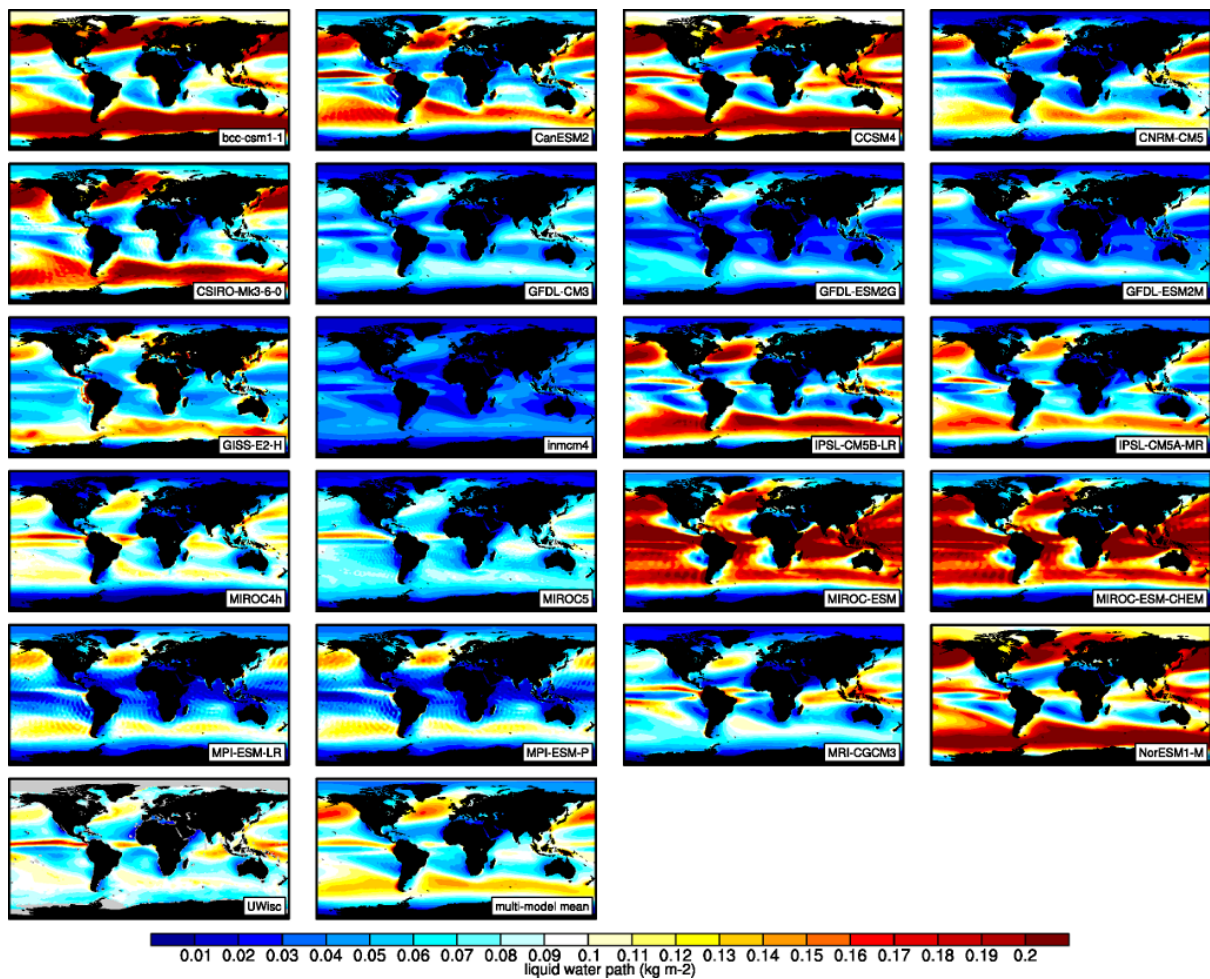


Figure 1: The 20-yr average LWP (1986-2005) from the CMIP5 historical model runs and the multi-model mean in comparison with the UWisc satellite climatology (1988-2007) based on SSM/I, TMI, and AMSR-E (O'Dell et al. 2008).

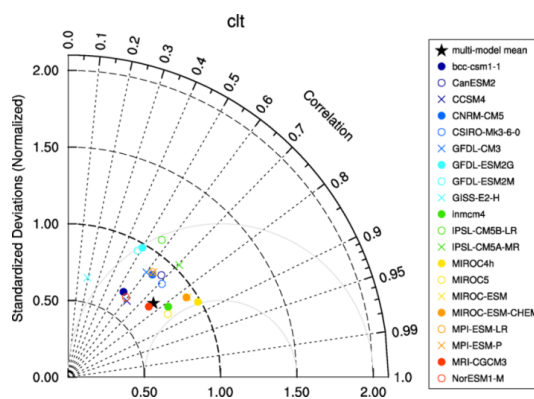


Figure 2: Taylor diagram showing the 20-yr annual average performance of CMIP5 models for total cloud fraction as compared to MODIS satellite observations.

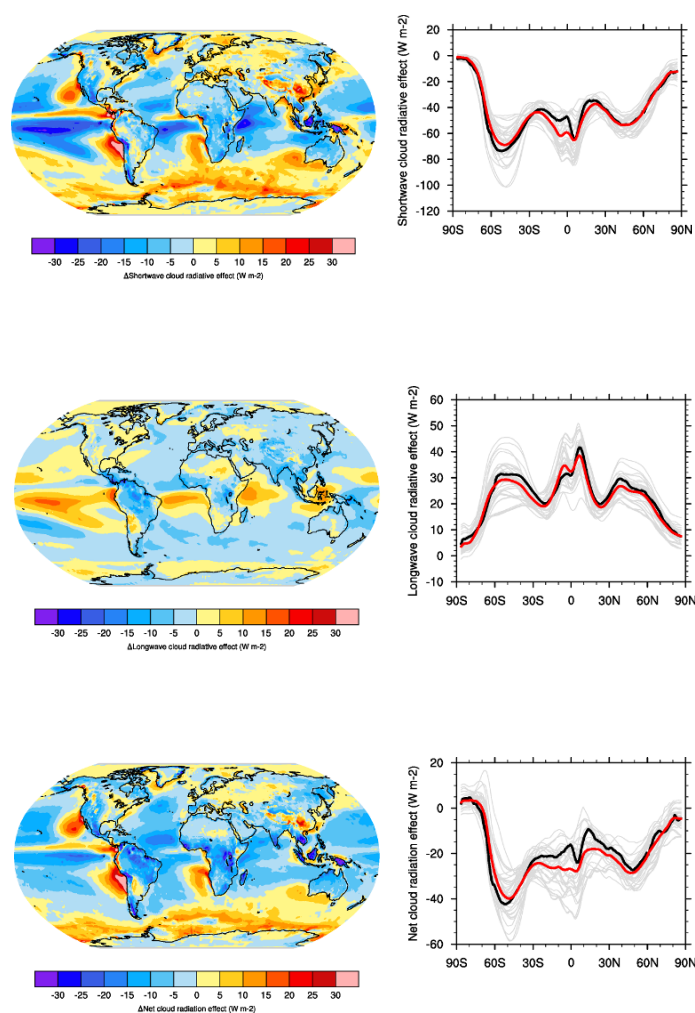


Figure 3: 20-year average (1986–2005) annual mean cloud radiative effects of CMIP5 models against the CERES EBAF (2001–2012). Top row shows the shortwave effect; middle row the longwave effect, and bottom row the net effect. Multi-model mean biases against CERES EBAF are shown on the left, whereas the right panels show zonal averages from CERES EBAF (thick black), the individual CMIP5 models (thin gray lines) and the multi-model mean (thick red line). Similar to Figure 9.5 of Flato et al. (2013).

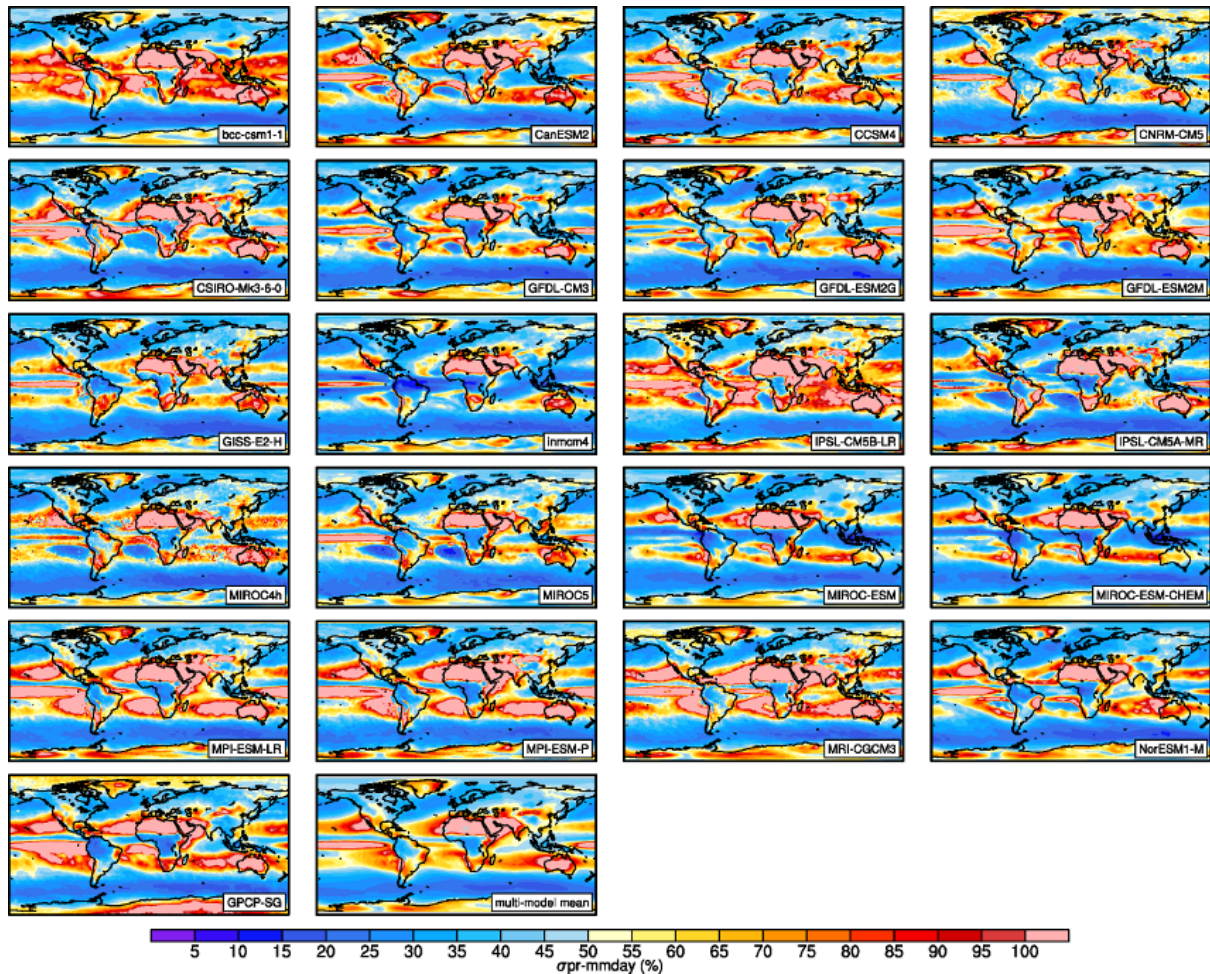


Figure 4: Interannual variability of modeled and observed (GPCP) precipitation rates estimated as relative temporal standard deviation from 20 years (1986-2005) of data. The temporal standard deviations are calculated from monthly anomalies after subtracting the climatological mean seasonal cycle.

Cloud Regime Error Metric (CREM)

22.1 Overview

The radiative feedback from clouds remains the largest source of uncertainty in determining the climate sensitivity. Traditionally, cloud has been evaluated in terms of its impact on the mean top of atmosphere fluxes. However it is quite possible to achieve good performance on these criteria through compensating errors, with boundary layer clouds being too reflective but having insufficient horizontal coverage being a common example (e.g., Nam et al., 2012). Williams and Webb (2009) (WW09) propose a Cloud Regime Error Metric (CREM) which critically tests the ability of a model to simulate both the relative frequency of occurrence and the radiative properties correctly for a set of cloud regimes determined by the daily mean cloud top pressure, cloud albedo and fractional coverage at each grid-box. WW09 describe in detail how to calculate their metrics and we have included the CREMpd metric from their paper in ESMValTool, with clear references in the lodged code to tables in their paper. This has been applied to those CMIP5 models who have submitted the required diagnostics for their AMIP simulation (see Figure 8 below). As documented by WW09, a perfect score with respect to ISCCP would be zero. WW09 also compared MODIS/ERBE to ISCCP in order to provide an estimate of observational uncertainty. This was found to be 0.96 and this is marked on Figure 8, hence a model with a CREM similar to this value could be considered to have an error comparable with observational uncertainty, although it should be noted that this does not necessarily mean that the model lies within the observations for each regime. A limitation of the metric is that it requires a model to be good enough to simulate each regime. If a model is that poor that the simulated frequency of occurrence of a particular regime is zero, then a NaN will be returned from the code and a bar not plotted on the figure for that model.

22.2 Available namelists and diagnostics

Namelists are stored in nml/

- `namelist_williams09climdyn_CREM.xml`

Diagnostics are stored in diag_scripts/

- `ww09_ESMValTool.py`

22.3 User settings

None.

22.4 Variables

- albiscpp (atmos, daily mean, longitude latitude time)
- cltiscpp (atmos, daily mean, longitude latitude time)
- pctiscpp (atmos, daily mean, longitude latitude time)
- rlut (atmos, daily mean, longitude latitude time)
- rlutcs (atmos, daily mean, longitude latitude time)
- rsut (atmos, daily mean, longitude latitude time)
- rsutcs (atmos, daily mean, longitude latitude time)
- sic (atmos, daily mean, longitude latitude time)
- snv (atmos, daily mean, longitude latitude time)

If snv is not available then snw can be used instead. For AMIP simulations, sic is often not submitted as it's a boundary condition and effectively the same for every model. In this case the same daily sic data set can be used for each model.

Note: in case of using sic data from a different model (AMIP), it has to be checked by the user that the calendar definitions of all data sets are compatible, in particular whether leap days are included or not.

22.5 Observations and reformat scripts

All observational data have been pre-processed and included within the routine. These are ISCCP, ISCCP-FD, MODIS, ERBE. No additional observational data are required at runtime.

22.6 References

- Williams, K.D. and Webb, M.J.: A quantitative performance assessment of cloud regimes in climate models. Clim. Dyn. 33, 141-157, doi: 10.1007/s00382-008-0443-1, 2009.

22.7 Example plots

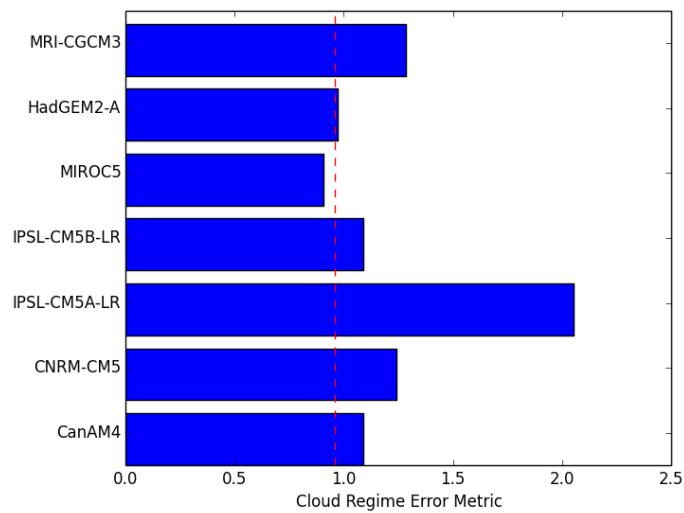


Figure 1: Cloud Regime Error Metrics (CREMpd) from William and Webb (2009) applied to those CMIP5 AMIP simulations with the required data in the archive. A perfect score with respect to ISCCP is zero; the dashed red line is an indication of observational uncertainty.

Diurnal cycle of convection

23.1 Overview

The diurnal cycle of precipitation and radiative fluxes is closely related to the representation of boundary-layer turbulence and cumulus clouds, as well as to the triggering criteria of the deep convection scheme, which will give the onset of precipitation, and its closure, which in turn determines its intensity and duration. Associated clouds will have a different radiative impact given their timing of occurrence within the day. Thus, the evaluation of the representation of the diurnal cycle of precipitation and radiative fluxes gives interesting insights into the progress achieved in the design and setup of physical parameterizations.

This diagnostic evaluates the representation of the diurnal cycle of precipitation over specific regions in the Tropics. The diurnal cycle is known to be different over land and ocean. Over land, the semi-arid Sahel region is contrasted with humid regions (Amazonia) as well as with the two monsoon regions West-Africa and India. Over the ocean, the diagnostic focuses on the Gulf of Guinea, the Indian Ocean and the East and West Equatorial Pacific. A harmonic analysis is done to provide global maps of the timing and amplitude of maximum rainfall with 3-hourly TRMM observations used as a reference.

Optionally, the analysis can be extended to include the diurnal cycle of radiative fluxes at the top of the atmosphere and at the surface (LW and SW, total and clear-sky components). 3-hourly CERES data are used as a reference derived from measurements at the top of the atmosphere and computed using a radiative transfer model at the surface.

23.2 Available namelists and diagnostics

Namelists are stored in `nml/`

- `namelist_DiurnalCycle_box_pr.xml`
- `namelist_DiurnalCycle_box_SFCflux.xml`
- `namelist_DiurnalCycle_box_TOAflux.xml`
- `namelist_DiurnalCycle_harmonic.xml`

Diagnostics are stored in `diag_scripts/`

The following diagnostic script is called by `namelist_DiurnalCycle_harmonic.xml`:

- `DiurnalCycle_precip_harmonic.ncl`: computes the amplitude, phase and percentage of the variance of the first harmonic of the mean diurnal cycle of precipitation by season over the globe. 3-hr mean precipitation outputs are used and interpolated onto a common 2.5°x2.5° grid.

The following diagnostic script is called by `namelist_DiurnalCycle_box_pr.xml`, `namelist_DiurnalCycle_box_SFCflux.xml`, `namelist_DiurnalCycle_box_TOAflux.xml`:

- `DiurnalCycle_box.ncl`: computes the mean diurnal cycle of precipitation and available radiative fluxes over specific regions and seasons. Regions include Sahel (Sahel, 12°N-23°N, 10°E-10°W), West-Africa (WestAf, 4°N-10°N, 8°E-8°W), Gulf of Guinea (GoG, 2°S-4°N, 8°E-8°W), Amazonia (Amazon, 15°S-0°, 70°W-50°W), India (India, 12°N-20°N, 75°E-80°E), Indian Ocean (IO, 10°S-5°N, 70°E-90°E), East-Equatorial Pacific (EEP, 0°-10°N, 110°W-90°W) and West-Equatorial Pacific (WEP, 0°-20°N, 130°E-150°E). Seasons include DJF, MAM, JJA and SON.

23.3 User settings

User setting files (cfg files) are stored in `nml/cfg_DiurnalCycle/`

1. `namelist_DiurnalCycle_harmonic.xml`

Required diag_script_info attributes

- `season`: season (DJF, MAM, JJA, SON)
- `destgrid`: common grid for interpolation ("2.5x2.5")
- `latrange_basic`: latitude range of box
- `lonrange_basic`: longitude range of box
- `cn_levels_amplitude_basic`: contour levels for amplitude plots
- `cn_levels_phase_basic`: contour levels for phase plots
- `cn_levels_var_basic`: contour levels for percentage of variance
- `styleset`: "CMIP5", "DEFAULT"
- `my_region`: label for region

2. `namelist_DiurnalCycle_box_pr.xml`, `namelist_DiurnalCycle_box_TOAflux.xml`

`namelist_DiurnalCycle_box_SFCflux.xml`,

Required diag_script_info attributes

- `season`: season (DJF, MAM, JJA, SON)
- `latrange`: latitude range of box
- `lonrange`: longitude range of box
- `styleset`: "CMIP5", "DEFAULT"
- `box`: label for region
- `multi_model_mean`: calculate multi-model mean ("y", "n")
- `supporting_gridlines`: display supporting gridline ("y", "n")
- `x_gridlines`: display gridline along x-axis ("y", "n")
- `y_gridlines`: display gridline along y-axis ("y", "n")

23.4 Variables

- `pr` (atmos, 3hr mean, longitude latitude time)

- rsds (atmos, 3hr mean, longitude latitude time)
- rsdscs (atmos, 3hr mean, longitude latitude time)
- rlds (atmos, 3hr mean, longitude latitude time)
- rldscs (atmos, 3hr mean, longitude latitude time)
- rsus (atmos, 3hr mean, longitude latitude time)
- rsuscs (atmos, 3hr mean, longitude latitude time)
- rlus (atmos, 3hr mean, longitude latitude time)
- rlut (atmos, 3hr mean, longitude latitude time)
- rlutcs (atmos, 3hr mean, longitude latitude time)
- rsut (atmos, 3hr mean, longitude latitude time)
- rsutcs (atmos, 3hr mean, longitude latitude time)

23.5 Observations and reformat scripts

Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

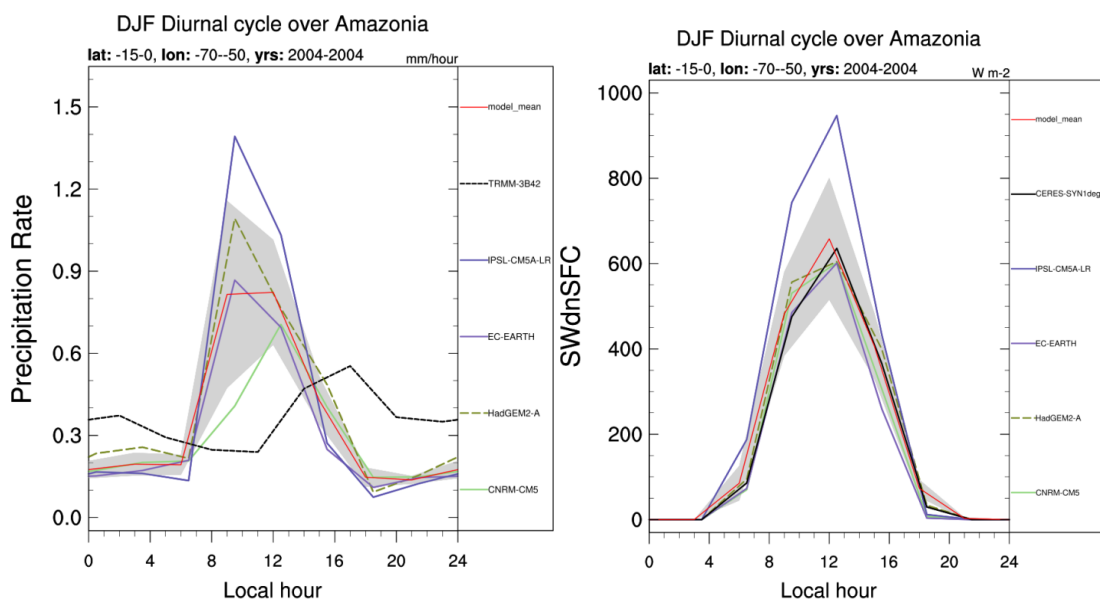
- TRMM-3B42, 3-hr, $0.25^\circ \times 0.25^\circ$: pr (obs4mips)
- CERES-SYN1deg, 3-hr, $1^\circ \times 1^\circ$: rsds, rsdscs, rlds, rldscs, rsus, rsuscs, rlus, rsut, rsutcs, rlut, rlutcs

Reformat scripts: reformat_scripts/obs/reformat_obs_CERES-SYN1deg-SFC.bash,
reformat_scripts/obs/reformat_obs_CERES-SYN1deg-TOA.bash

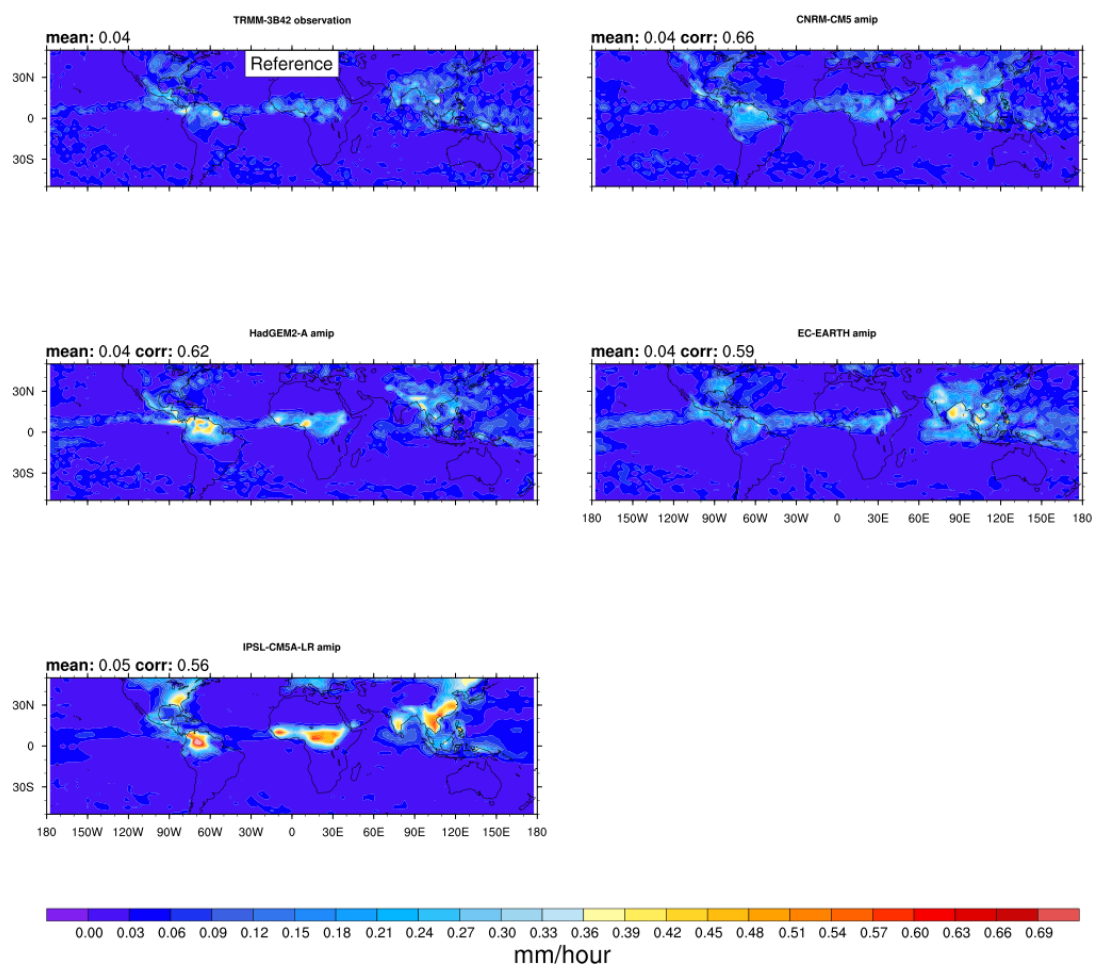
23.6 References

None.

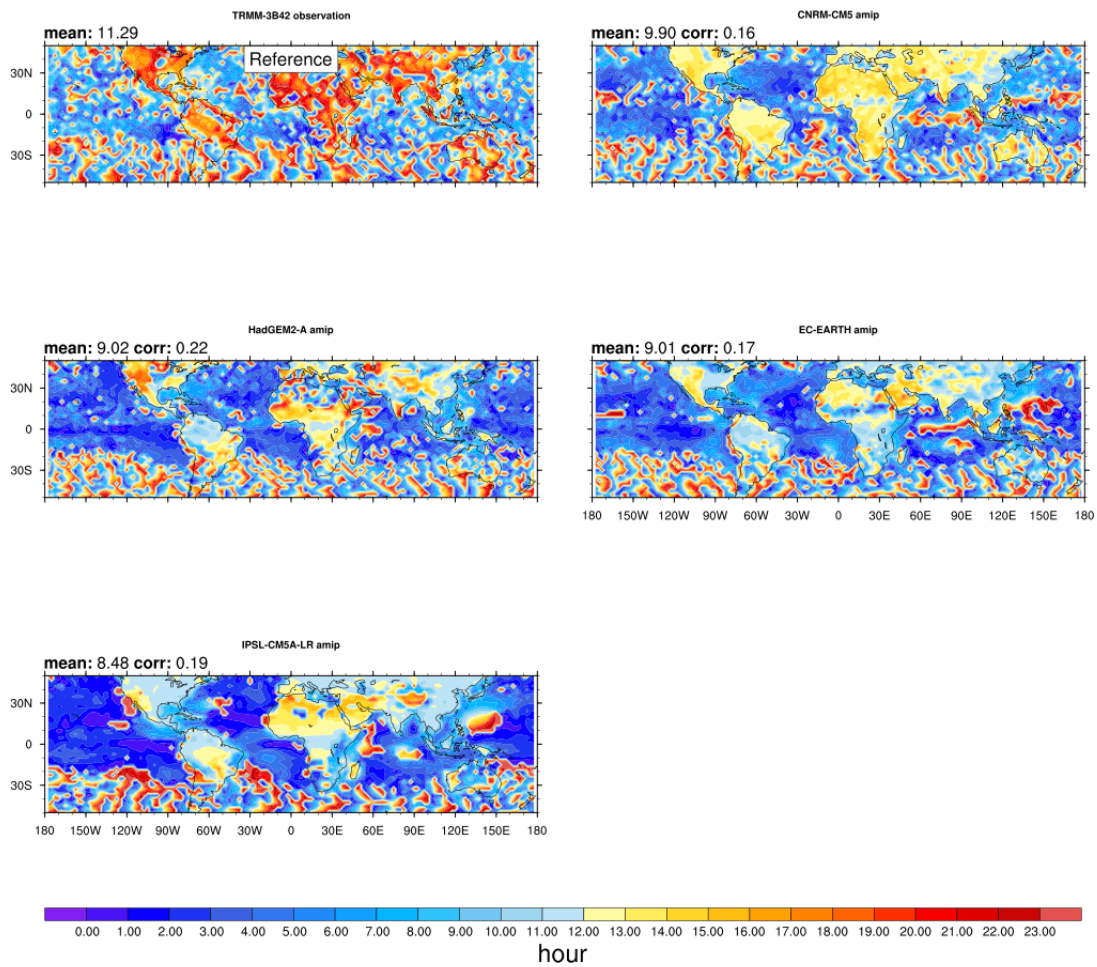
23.7 Example plots



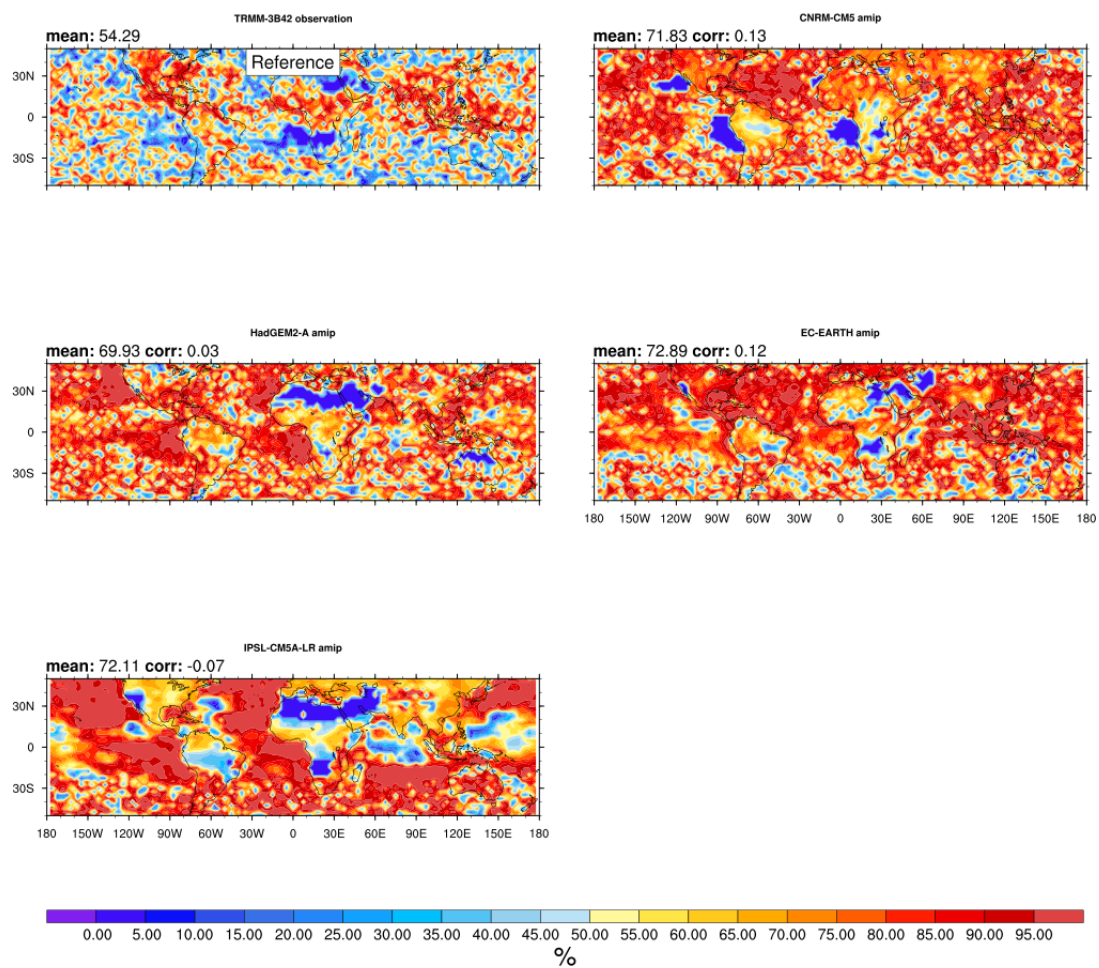
JJA-amplitude of the 1st harmonic of the mean diurnal cycle of Precipitation Rate



JJA-phase of the 1st harmonic of the mean diurnal cycle of Precipitation Rate



JJA-var of the 1st harmonic of the mean diurnal cycle of Precipitation Rate



Emergent constraints on carbon cycle feedbacks

24.1 Overview

Figures from Wenzel et al. (2014) are reproduced with `namelist_wenz14jgr.xml`. Variables relevant for the carbon cycle-climate feedback such as near surface air temperature (`tas`), net biosphere productivity (`nbp`) and (`fgco2`) are analyzed for coupled (`1pctCO2`; here the carbon cycle is fully coupled to the climate response) and uncoupled (`esmFixCLim1`; here the carbon cycle is uncoupled to the climate response) simulations. The standard namelist includes a comparison of cumulated `nbp` from coupled and uncoupled simulations and includes a set of routines to diagnose the long-term carbon cycle-climate feedback parameter (`GammaLT`) from an ensemble of CMIP5 models. Also included in the namelist is a comparison of the interannual variability of `nbp` and `fgco2` for historical simulations used to diagnose the observable sensitivity of `CO2` to tropical temperature changes (`GammaIAV`). As a key figure of this namelist, the diagnosed values from the models `GammaLT` vs. `GammaIAV` are compared in a scatter plot constituting an emergent constraint.

24.2 Available namelists and diagnostics

Namelists are stored in `nml/`

- `namelist_wenzel14jgr.xml`

Diagnostics are stored in `diag_scripts/`

- `tsline.ncl`: time line plots of annual means for spatial averages
- `carbon_corr_2var.ncl`: scatter plot of annual mean anomalies of two different variables; diagnosing and saving `GammaIAV`
- `carbon_constraint.ncl`: scatter plot of `GammaLT` vs. `GammaIAV` + line plot of probability density functions; diagnosing `GammaLT`

24.3 User settings

User setting files (`cfg` files) are stored in `nml/cfg_carbon/`

1. `tsline` (`cfg_carbon_line.ncl`, `cfg_carbon_line_h.ncl`)

Required `diag_script_info` attributes

- styleset: "CMIP5"
 - ts_minlat: minimum latitude for area-averaging
 - ts_maxlat: maximum latitude for area-averaging
 - ts_minlon: minimum longitude for area-averaging
 - ts_maxlon: maximum longitude for area-averaging
 - multi_model_mean: True for multi-model mean calculation
 - ts_maxyear: last year (time range)
 - ts_minyear: first year (time range)
 - time_avg: "yearly" (currently, only yearly is available)
 - ts_anomaly: calculates anomalies with respect to the first 10-year average (anom, noanom)
 - area_opper: type of area operation (average, sum)
2. ts_line (cfg_carbon_line_h.ncl)
- Required diag_script_info attributes
- ts_detrend: detrend time series (detr, nodetr)
3. create_co2_flux.ncl, carbon_corr_2var.ncl
- Required diag_script_info attributes
- styleset: "CMIP5"
 - con_latrange: 2-element array of latitudes for time series plots if 2D or 3D
 - con_lonrange: 2-element array of pressure levels for time series plots if 2D or 3D
 - con_units: label string for units, e.g., "GtC/K"
 - pcBGC: label for CMIP5 experiment (for creating filename mask), e.g., "esmFixClim1"
 - reg_models: models to be included in emergent constraint (n-element array or (/ "all" /)); **note: has to be consistent with the list of models specified in the namelist!**

24.4 Variables

- tas (atmos, monthly mean, longitude latitude time)
- co2 (atmos, monthly mean, plev longitude latitude time)
- nbp (land, monthly mean, longitude latitude time)
- fgco2 (ocean, monthly mean, longitude latitude time)

24.5 Observations and reformat scripts

Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

- GCP - Global Carbon Budget including land (nbp) and ocean (fgco2) carbon fluxes

Reformat script: reformat_scripts/obs/reformat_obs_GCP.ncl

- NCEP - National Centers for Environmental Prediction reanalysis data for near surface temperature

Reformat script: reformat_scripts/obs/reformat_obs_NCEP.ncl

24.6 References

- Cox, P. M., D. B. Pearson, B. B. Booth, P. Friedlingstein, C. C. Huntingford, C. D. B. Jones, and C. M. Luke, 2013, Sensitivity of tropical carbon to climate change constrained by carbon dioxide variability, *Nature*, 494(7437), 341–344. doi: 10.1038/nature11882
- Wenzel, S., P. M. Cox, V. Eyring, and P. Friedlingstein, 2014, Emergent Constraints on Climate Carbon Cycle Feedbacks in the CMIP5 Earth System Models, *JGR Biogeoscience*, 119(5), doi: 2013JG002591.

24.7 Example plots

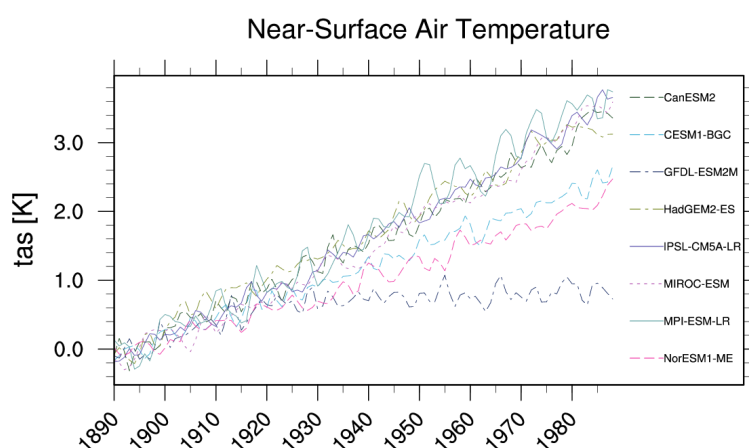


Figure 1: Time series of tropical (30°S - 30°N) mean near surface temperature (tas) change between year 30 and year 110 for the CMIP5 models simulated with prescribed CO₂ (1%/yr CO₂ increase) coupled simulation (1pctCO₂).

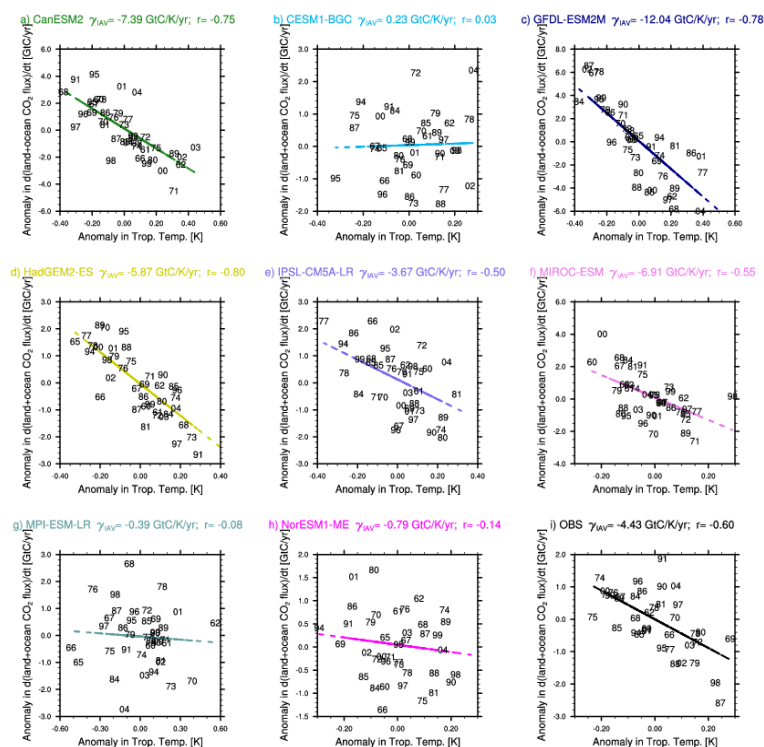


Figure 2: Correlations between the interannual variability of global co2flux (nbpc+fgco2) and tropical temperature for the individual CMIP5 models using esmHistorical simulations, and for observations.

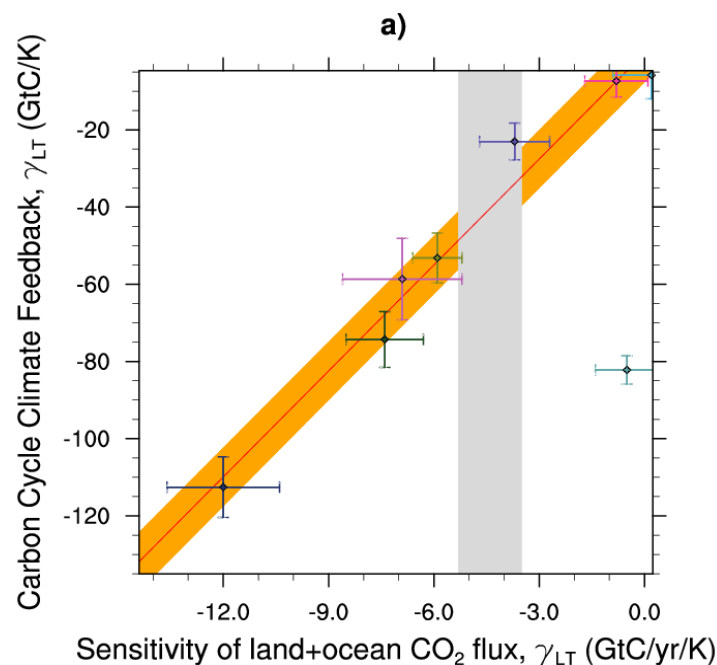


Figure 3: Carbon cycle-climate feedback of tropical land carbon vs. the sensitivity of co2flux to interannual temperature variability in the tropics (30°S-30°N). The red line shows the linear best fit of the regression together with the prediction error (orange shading) and the gray shading shows the observed range.

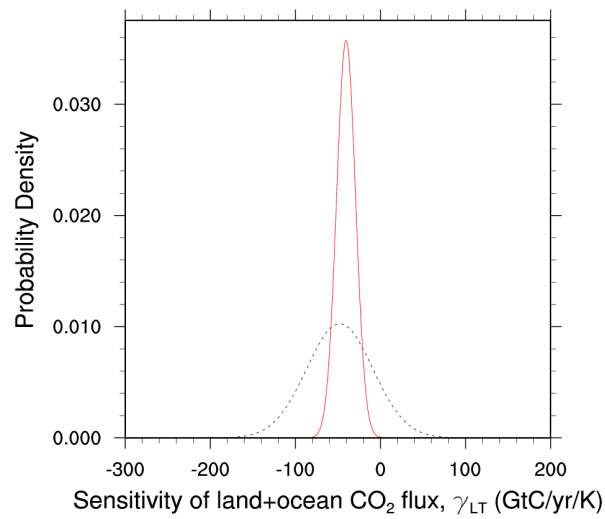


Figure 4: Probability Density Functions for the pure CMIP5 ensemble (black dashed) and after applying the observed constraint to the models (red solid)

25.1 Overview

Namelist for creating the 20 figures from Lauer et al. (2017) using the European Space Agency's Climate Change Initiative (ESA CCI) data sets for sea surface temperature, sea ice, cloud, soil moisture, land cover, aerosol, ozone, and greenhouse gases (CO₂). This namelist demonstrates the value of the ESA CCI data for model evaluation providing an overview on the possible applications of the new data to evaluating CMIP models.

25.2 Available namelists and diagnostics

Namelists are stored in nml/

- namelist_lauer17rse.xml

Diagnostics are stored in diag_scripts/

- aerosol_stations.ncl: comparison of ESA CCI aerosol with AeroNet and MODIS
- clouds.ncl: global maps of (multi-year) annual means including multi-model mean
- clouds_interannual.ncl: global maps of interannual variability of cloud properties
- clouds_ipcc.ncl: maps of multi-model mean bias and zonal averages
- clouds_taylor.ncl: taylor diagrams
- eyring13jgr_fig01.ncl: calculates seasonal cycles of zonally averaged total ozone columns.
- eyring13jgr_fig02.ncl: time series of area-weighted total ozone from 1960-2005 for the annual mean averaged over the global domain (90°S-90°N), Tropics (25°S-25°N), northern mid-latitudes (35°N-60°N), southern mid-latitudes (35°S-60°S), and the March and October mean averaged over the Arctic (60°N-90°N) and the Antarctic (60°S-90°S).
- eyring13jgr_fig04.ncl: climatological annual mean tropospheric ozone columns (geographical distribution).
- lc_ESACCI.py: ESA CCI land cover diagnostics including global maps of grass and cropland cover and of forest and shrub cover
- perfmetrics_grading.ncl: calculates grades according to a given metric with different options for normalization. It requires fields precalculated by perfmetrics_main.ncl (see *Performance metrics for essential climate parameters*).

- `perfmetrics_grading_collect.ncl`: collects results from metrics previously calculated by `perfmetrics_grading.ncl` and passes them to the plotting functions (see *Performance metrics for essential climate parameters*).
- `perfmetrics_main.ncl`: calculates and (optionally) plots annual/seasonal cycles, zonal means, lat-lon fields and time-lat-lon fields from input monthly 2-d or 3-d (“T2M”, “T3Ms”) data. The calculated fields can be also plotted as difference w.r.t. a given reference model. They are also used as input to calculate grading metrics (see `perfmetrics_grading.ncl`) (see *Performance metrics for essential climate parameters*).
- `SeaIce_polcon.ncl`: polar stereographic plots of sea ice concentration (= sea ice area fraction) and extent (grid cells with a sea ice concentration of at least 15%) for individual models or observational data sets, for Arctic and Antarctic regions with flexible paneling of the individual plots. The edges of sea ice extent can be highlighted via an optional red line.
- `SeaIce_polcon_diff.ncl`: polar stereographic plots of sea ice area concentration difference between individual models and reference data (e.g., an observational data set) for both Arctic and Antarctic with flexible paneling of the individual plots. All data are regridded to a common grid (1°x1°) before comparison.
- `SeaIce_tsline.ncl`: time series line plots of total sea ice area and extent (accumulated) for northern and southern hemispheres with optional multi-model mean and standard deviation. One value is used per model per year, either annual mean or the mean value of a selected month.
- `sm_ESACCI.py`: ESA CCI soil moisture diagnostics including global maps of temporal trend in soil moisture and percentile maps for soil moisture
- `sst_ESACCI.py`: ESA CCI SST diagnostics including global maps of absolute and relative SST bias and time series of mean SST for different ocean basins
- `tsline.ncl`: time line plots of annual means for spatial averages
- `vpline.ncl`: produces vertical profiles according to Eyring et al. (2006) Figure 5 upper panels following `eyring06jgr_fig05.ncl`.

25.3 User settings

User setting files (cfg files) are stored in `nml/cfg_lauer17rse/`

- `cfg_aerosol_stations_AERONET.ncl`
- `cfg_carbon_line_3030.ncl`
- `cfg_carbon_line_3060.ncl`
- `cfg_carbon_line_6030.ncl`
- `cfg_carbon_line_h.ncl`
- `cfg_clouds_err.ncl`
- `cfg_clouds_interannual_esa.ncl`
- `cfg_clouds_ipcc.ncl`
- `cfg_clouds.ncl`
- `cfg_clouds_taylor_esa.ncl`
- `cfg_clouds_taylor_esa-sic.ncl`
- `cfg_dummy.conf`
- `cfg_esacci_vpline.ncl`
- `cfg_eyring13jgr_fig01.ncl`
- `cfg_eyring13jgr_fig01_NIWA.ncl`
- `cfg_eyring13jgr_fig02.ncl`

- `cfg_eyring13jgr_fig04.ncl`
- `cfg_lc_ESACCI.py`
- `cfg_perfmetrics_grading_collect.ncl`
- `cfg_perfmetrics_grading_RMSD_200_glob.ncl`
- `cfg_perfmetrics_grading_RMSD_400_glob.ncl`
- `cfg_perfmetrics_grading_RMSD_500_glob.ncl`
- `cfg_perfmetrics_grading_RMSD_850_glob.ncl`
- `cfg_perfmetrics_grading_RMSD_all_glob_aero.ncl`
- `cfg_perfmetrics_grading_RMSD_all_glob.ncl`
- `cfg_perfmetrics_grading_RMSD_all_glob_sm.ncl`
- `cfg_perfmetrics_grading_RMSD_all_glob_toz.ncl`
- `cfg_perfmetrics_grading_RMSD_all_glob_ts.ncl`
- `cfg_perfmetrics_grading_RMSD_all_glob_xco2.ncl`
- `cfg_perfmetrics_grading_RMSD_all_NHpolar_sic.ncl`
- `cfg_perfmetrics_grading_RMSD_all_SHpolar_sic.ncl`
- `cfg_perfmetrics_grading_RMSD_all_SHpolar_toz.ncl`
- `cfg_perfmetrics_latlon_annualclim_all_glob_aerosol.ncl`
- `cfg_perfmetrics_latlon_annualclim_all_glob.ncl`
- `cfg_SeaIce_NH.ncl`
- `cfg_SeaIce_SH.ncl`
- `cfg_sm_ESACCI.py`
- `cfg_sst_ESACCI_fig3.py`
- `cfg_sst_ESACCI_fig4.py`

25.4 Variables

- `abs550aer`
- `clt, cltStderr`
- `grassNcropFrac`
- `hus`
- `LW_CRE`
- `od550aer, od550aerStderr`
- `od550lt1aer`
- `od870aer, od870aerStderr`
- `pr`
- `rlut, rsut`
- `shrubNtreeFrac`
- `sic`
- `sm, smStderr`

- SW_CRE
- ta
- tas
- tos
- toz, tozStderr
- tro3prof
- ts, tsStderr
- ua, va
- xco2, xco2Stderr
- zg

25.5 Observations and reformat scripts

Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

- AIRS (hus): obs4mips
- BDBP (tro3prof): reformat_scripts/obs/reformat_obs_BDBP.ncl
- CERES-EBAF (LW_CRE, rlut, rsut, SW_CRE): obs4mips
- CLARA-A2 (clt): *contact ESMValtool development team*
- ERA-Interim (hus, ta, tas, ua, va, zg): reformat_scripts/obs/reformat_obs_ERA-Interim.ncl, reformat_scripts/obs/reformat_obs_ERA-Interim_surffluxes.ncl
- ESACCI-AEROSOL (abs550aer, od550aer, od550aerStderr, od550lt1aer, od870aer, od870aerStderr): reformat_scripts/obs/reformat_obs_ESACCI-AEROSOL.ncl
- ESACCI-CLOUD (clt, cltStderr): reformat_scripts/obs/reformat_obs_ESACCI-CLOUD.ncl
- ESACCI-GHG (xco2, xco2Stderr): reformat_scripts/obs/reformat_obs_ESACCI-GHG.csh
- ESACCI-LANDCOVER (grassNcropFrac, shrubNtreeFrac): reformat_scripts/obs/reformat_obs_ESACCI-LANDCOVER.py
- ESACCI-OZONE (toz, tozStderr, tro3prof): reformat_scripts/obs/reformat_obs_ESACCI-OZONE.ncl, reformat_scripts/obs/reformat_obs_ESACCI-OZONE_LP.ncl
- ESACCI-SIC (sic): reformat_scripts/obs/reformat_obs_ESACCI-sic.ncl
- ESACCI-SOILMOISTURE (sm, smStderr): reformat_scripts/obs/reformat_obs_ESACCI-SOILMOISTURE.ncl
- ESACCI-SST (tos, ts, tsStderr): reformat_scripts/obs/reformat_obs_ESACCI-SST.ncl
- GPCP-SG (pr): obs4mips
- HadISST (ts): reformat_scripts/obs/reformat_obs_HadISST.ncl
- MODIS-L3-C6 (clt, od550aer): reformat_scripts/obs/reformat_obs_MODIS-L3-C6.ncl
- NCEP (ta, tas, ua, va, zg): reformat_scripts/obs/reformat_obs_NCEP.ncl
- NIWA (toz): reformat_scripts/obs/reformat_obs_NIWA.ncl
- NSIDC-NT (sic): reformat_scripts/obs/reformat_obs_NSIDC.ncl
- PATMOS (clt): *contact ESMValtool development team*

25.6 References

- Lauer, A., V. Eyring, M. Righi, M. Buchwitz, P. Defourny, M. Evaldsson, P. Friedlingstein, R. de Jeuf, G. de Leeuw, A. Loew, C. J. Merchant, B. Müller, T. Popp, M. Reuter, S. Sandven, D. Senftleben, M. Stengel, M. Van Roozendaal, S. Wenzel, and U. Willén: Benchmarking CMIP5 models with a subset of ESA CCI Phase 2 data using the ESMValTool, Remote Sensing of Environment, <http://dx.doi.org/10.1016/j.rse.2017.01.007>, 2017.

25.7 Example plots

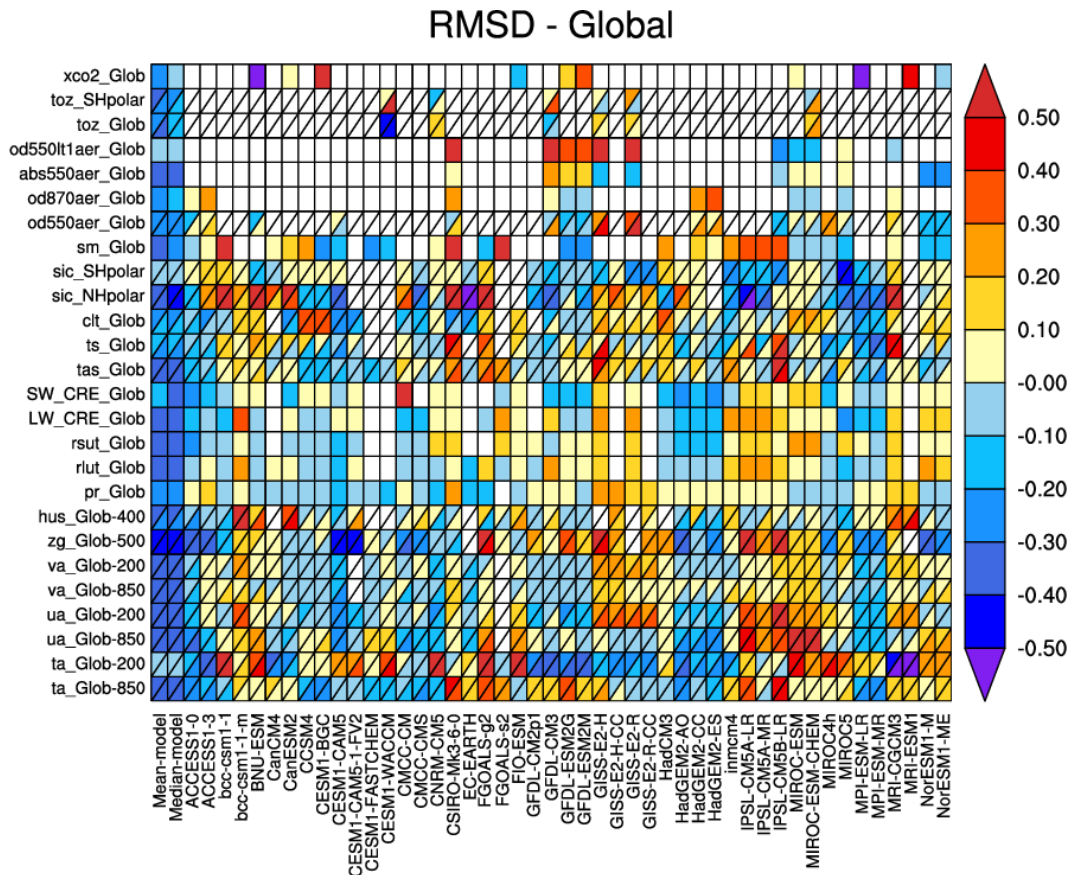


Figure 1: Relative space-time root-mean-square deviation (RMSD) calculated from the climatological seasonal cycle of the CMIP5 simulations (Lauer et al. 2017, Fig. 1).

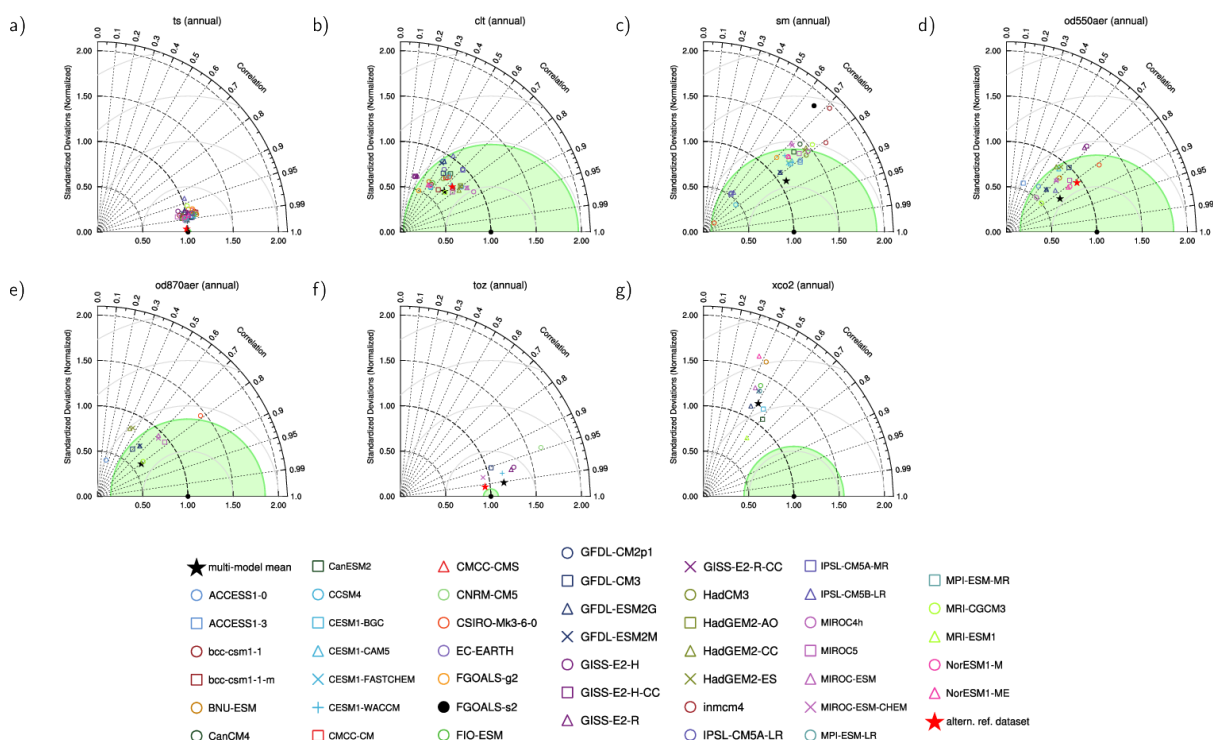


Figure 2: Extended Taylor diagrams showing the multi-year annual average performance of CMIP5 models in comparison with ESA CCI data (Lauer et al. 2017, Fig. 2).

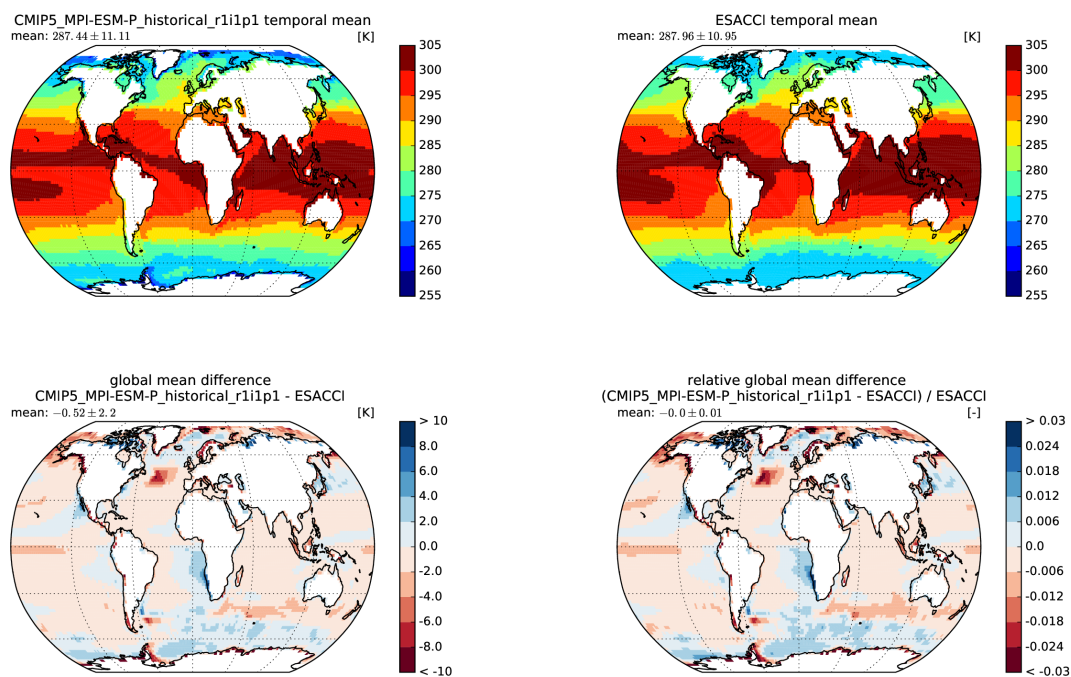


Figure 3: Temporal means of SST in K for the ESA CCI data set (top right) and the CMIP5 model MPI-ESM (top left) as well as absolute (bottom left) and relative differences (bottom right) (Lauer et al. 2017, Fig. 3).

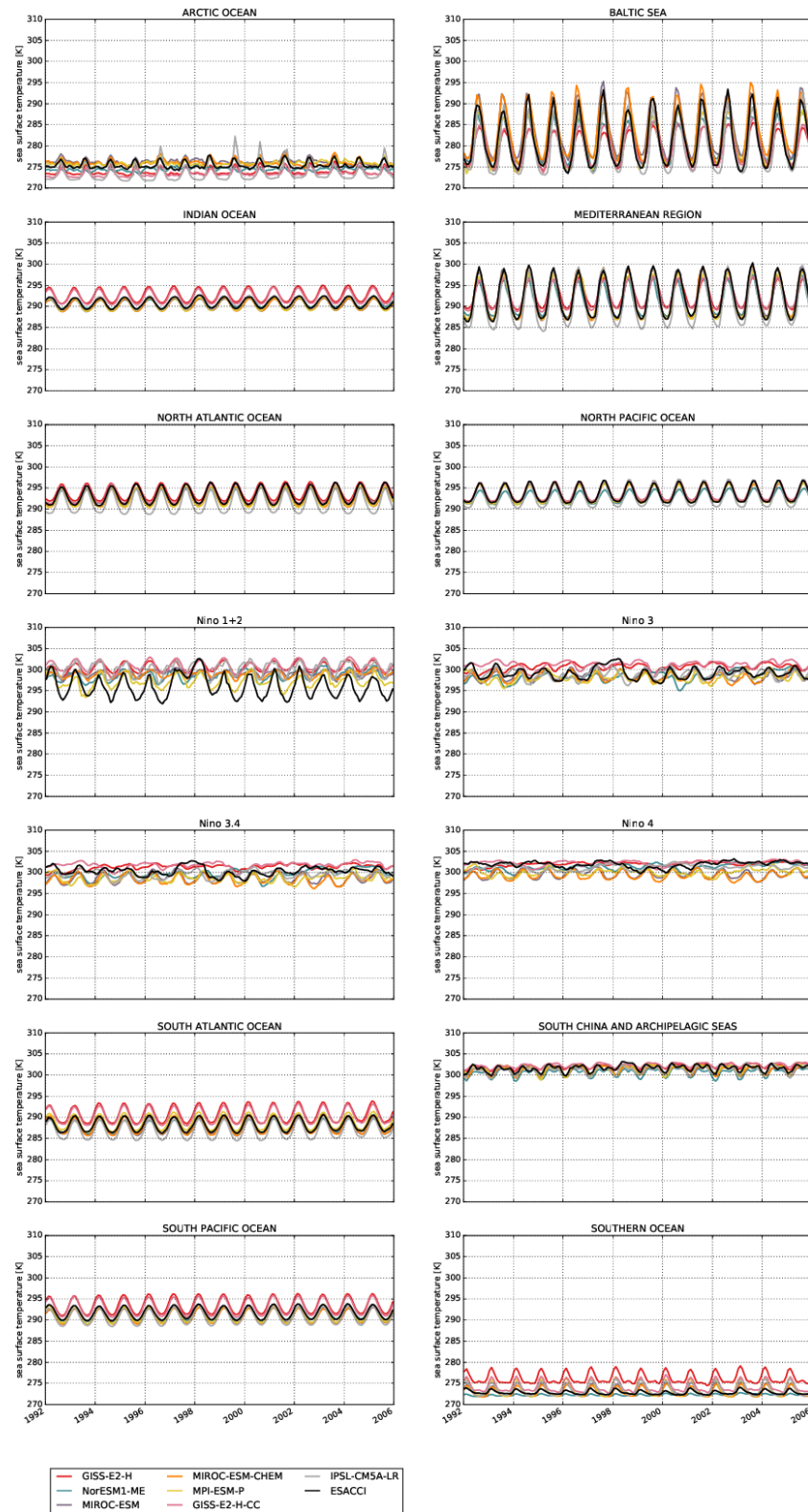


Figure 4: Time series of SST for different ocean basins from 7 CMIP5 models compared with the ESA CCI SST data (Lauer et al. 2017, Fig. 4).

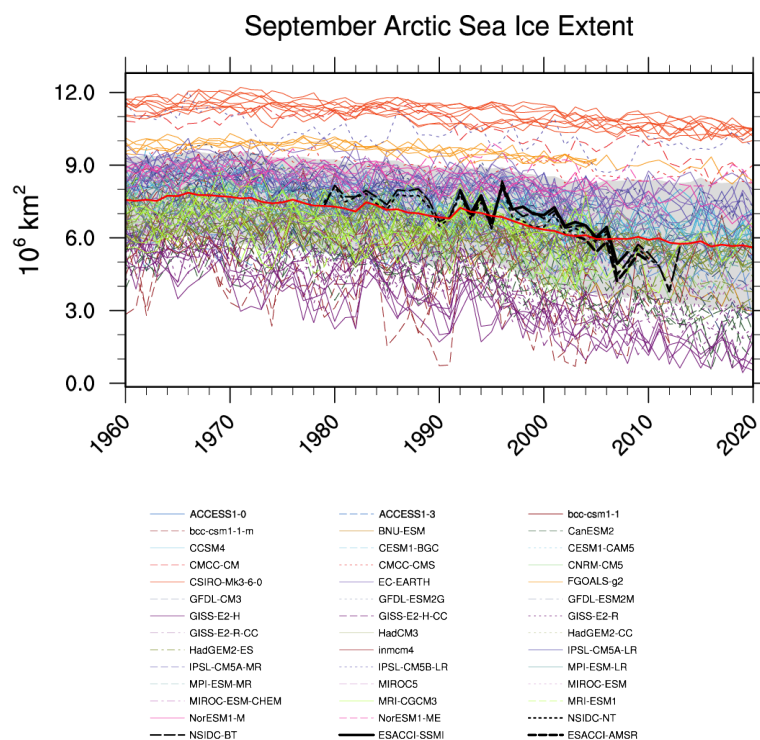


Figure 5: Evolution (1960-2020) of September Arctic sea ice extent in million km^2 from the CMIP5 models (colored lines) and from observations (thick black lines) (Lauer et al. 2017, Fig. 5).

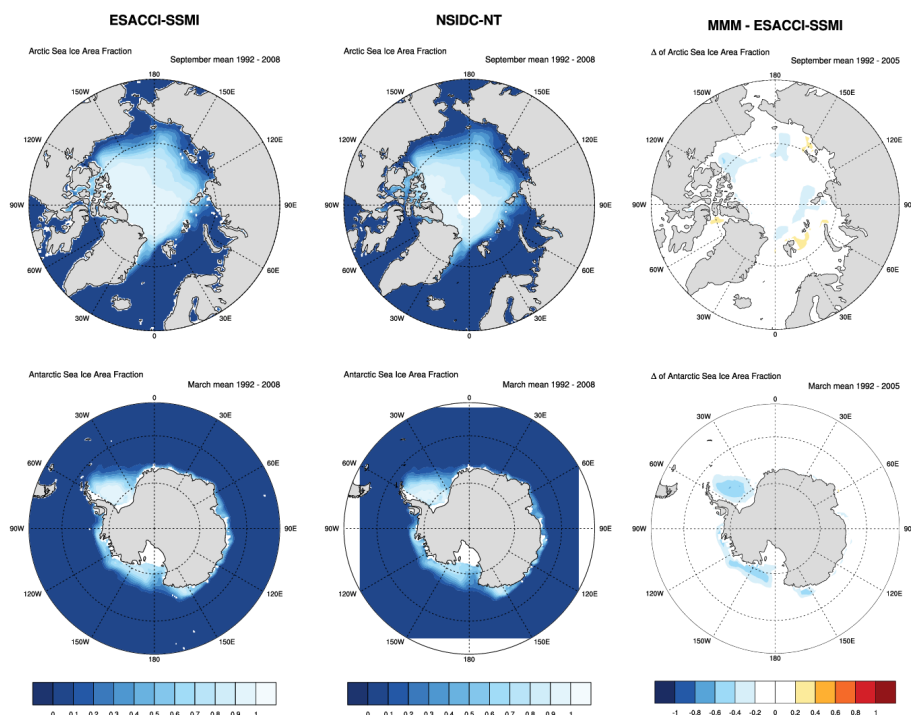


Figure 6: Polar-stereographic map of Arctic September (upper row) and Antarctic March (lower row) sea ice concentration from ESA CCI SI SSM/I (left column) and NSIDC-NT (middle column) observations averaged over the years 1992-2008. The right column depicts the differences between the CMIP5 multi-model mean and the ESA CCI SI SSM/I observations averaged over the years 1992-2005 (Lauer et al. 2017, Fig. 6).

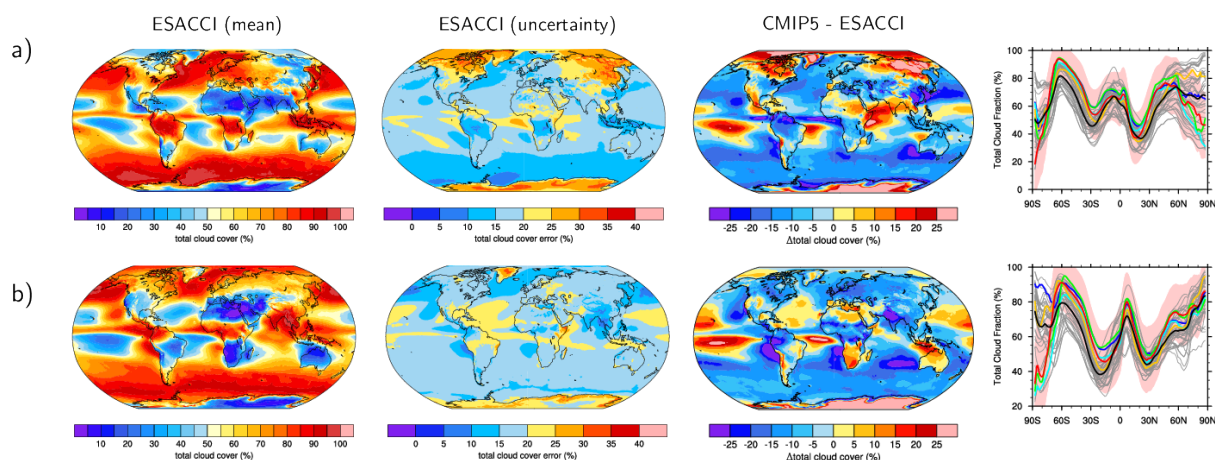


Figure 7: Maps of the multi-years seasonal mean of total cloud cover, 1-sigma uncertainty from ESA CCI cloud, the differences between the ESA CCI data and the CMIP5 multi-model mean, and zonal means (Lauer et al. 2017, Fig. 7).

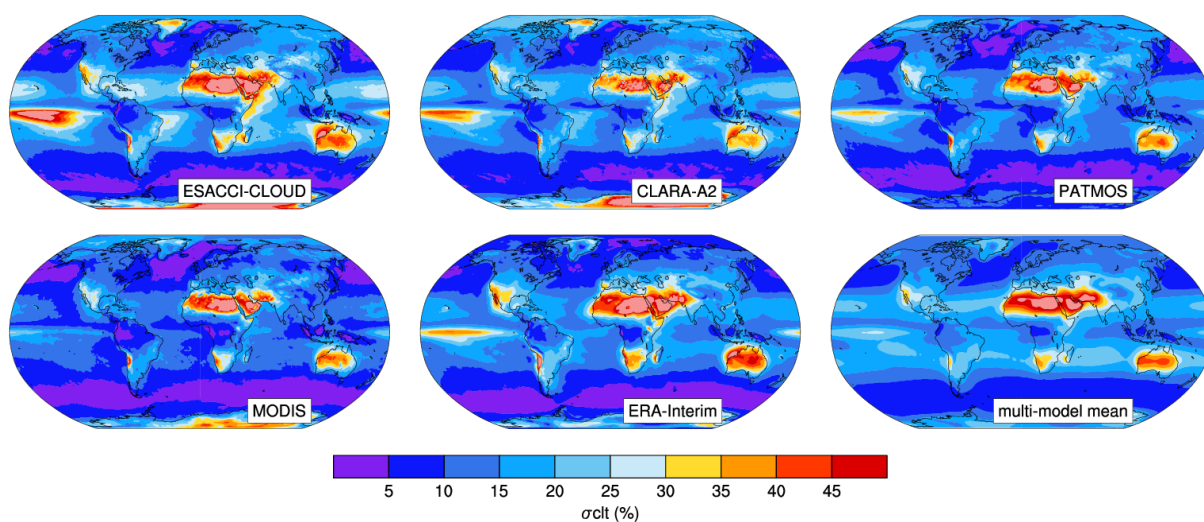


Figure 8: Interannual variability in total cloud cover estimate from relative temporal standard deviation of the deseasonalized monthly means time series from 1982 to 2014 (Lauer et al. 2017, Fig. 8).

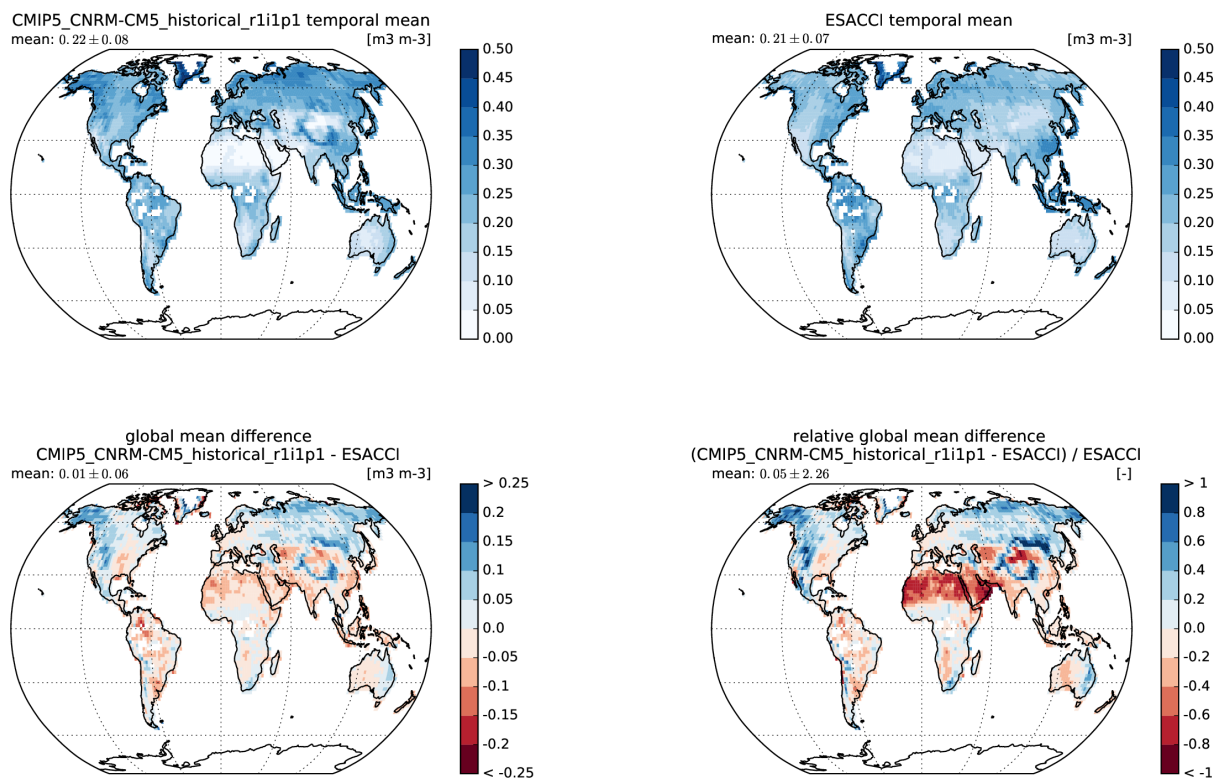


Figure 9: Temporal mean fields of volumetric soil moisture from the CNRM-CM5 model (top left), the ESA CCI soil moisture data set (top right) as well as their absolute (bottom left) and relative differences (bottom right) (Lauer et al. 2017, Fig. 9).

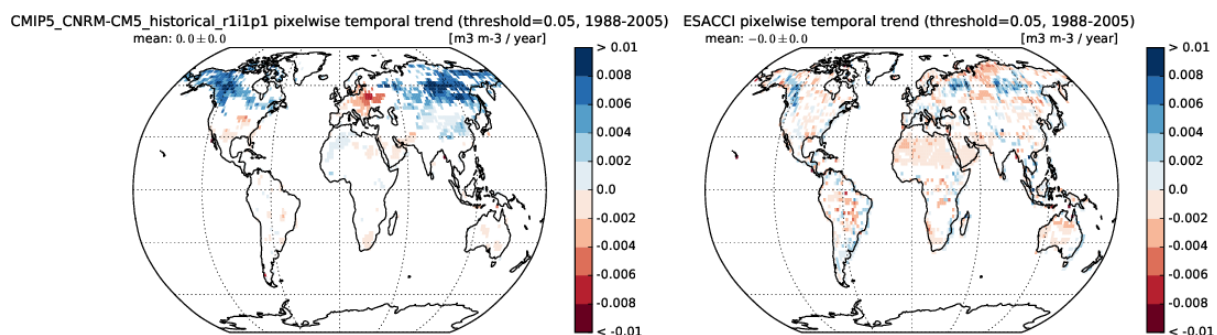


Figure 10: Temporal trend in soil moisture over the period 1988-2008 as derived from the CNRM-CM5 model (left) and the ESA CCI soil moisture data sets (right) (Lauer et al. 2017, Fig. 10).

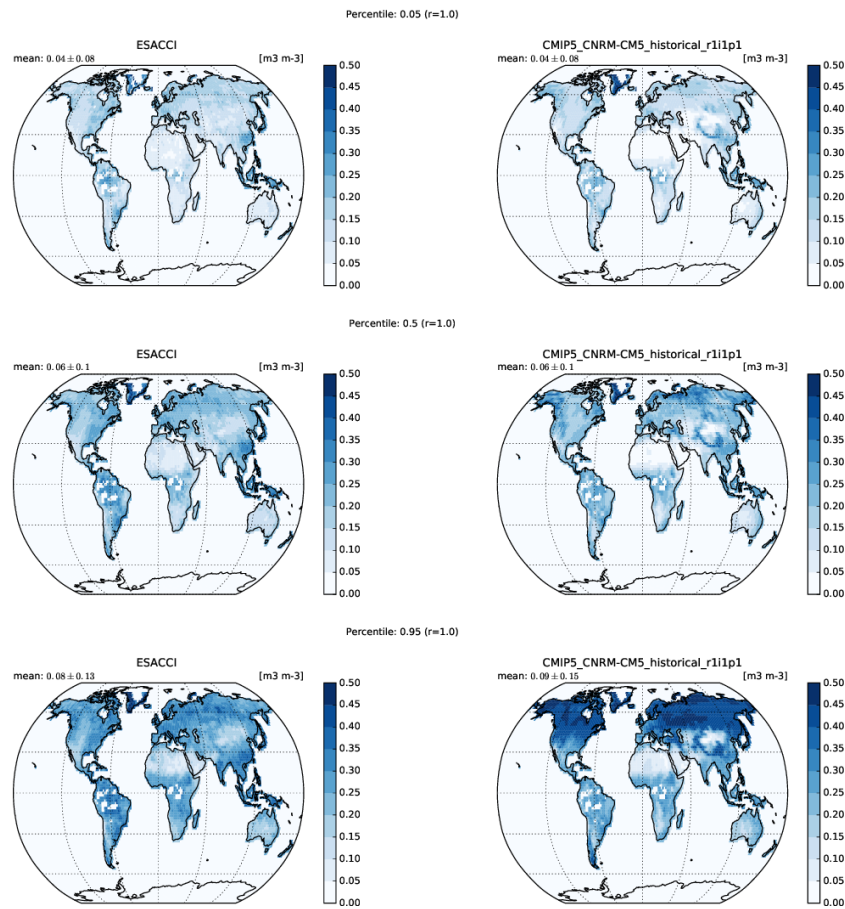


Figure 11: Percentile maps for ESA CCI soil moisture (left column) and soil moisture from CNRM-CM5 (right column) (Lauer et al. 2017, Fig. 11).

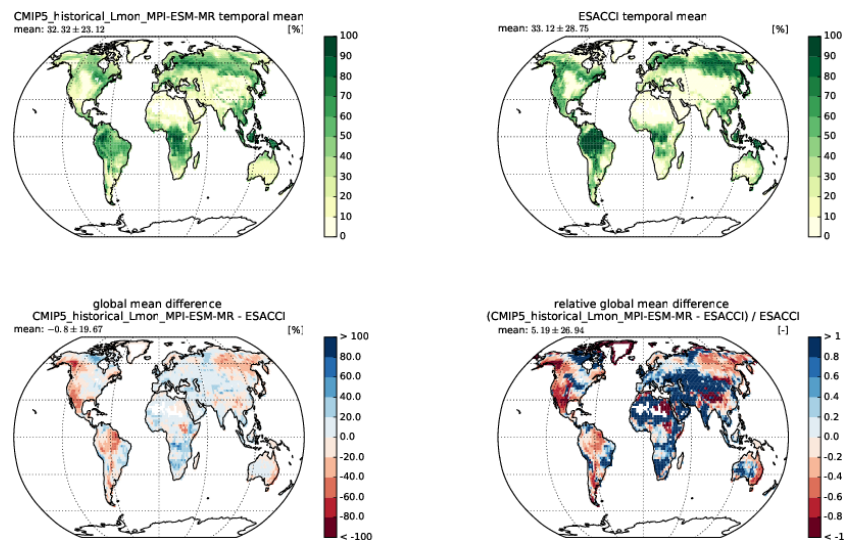


Figure 12: Area fraction (%) of forest and shrub cover in the MPI-ESM-MR model (top left) and the ESA CCI land cover data set (top right) and absolute (bottom left) and relative differences (bottom right) (Lauer et al. 2017, Fig. 12).

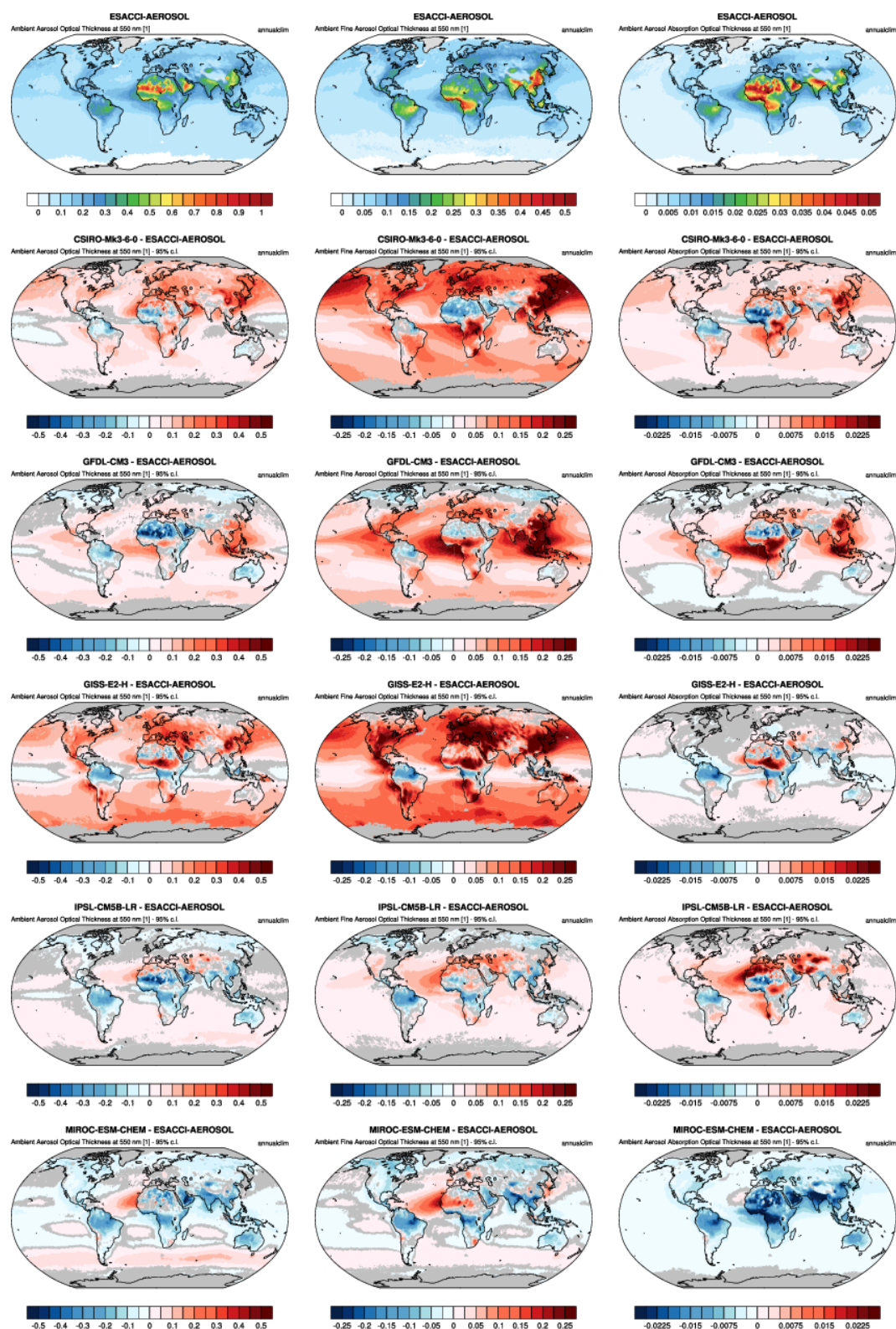


Figure 13: Climatological mean AOD (left column), fine mode optical depth (middle) and absorption optical depth (right column) at 550 nm averaged over the period 1997-2011. The first row shows the the observations (ESA CCI ATSR SU v4.21), the other rows the differences between selected CMIP5 models with interactive aerosols and the ESA CCI data (Lauer et al. 2017, Fig. 14).

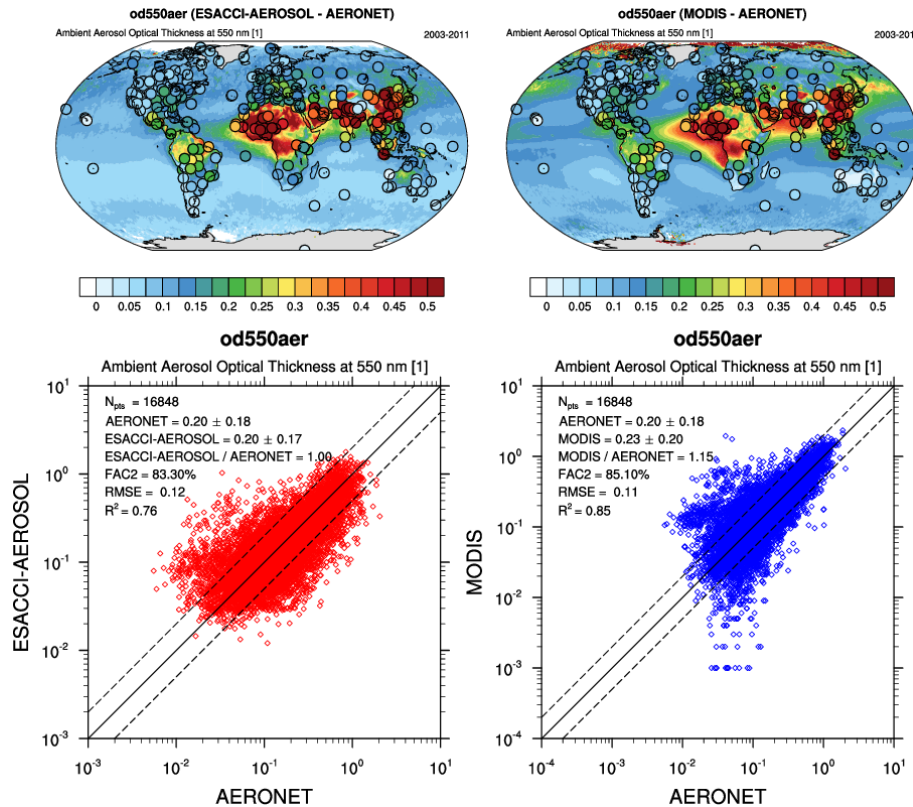


Figure 14: Comparison of AOD at 550 nm from the ESA CCI ATSR SU v4.21 and the MODIS Terra C6 satellite products against the AERONET ground-based measurements for the period 2003-2011. The top row shows the AERONET values as open circles plotted on top of the satellite data averaged over the same time period. The bottom row shows scatter plots of spatially and temporally collocated measurements on a monthly-mean basis (Lauer et al. 2017, Fig. 15).

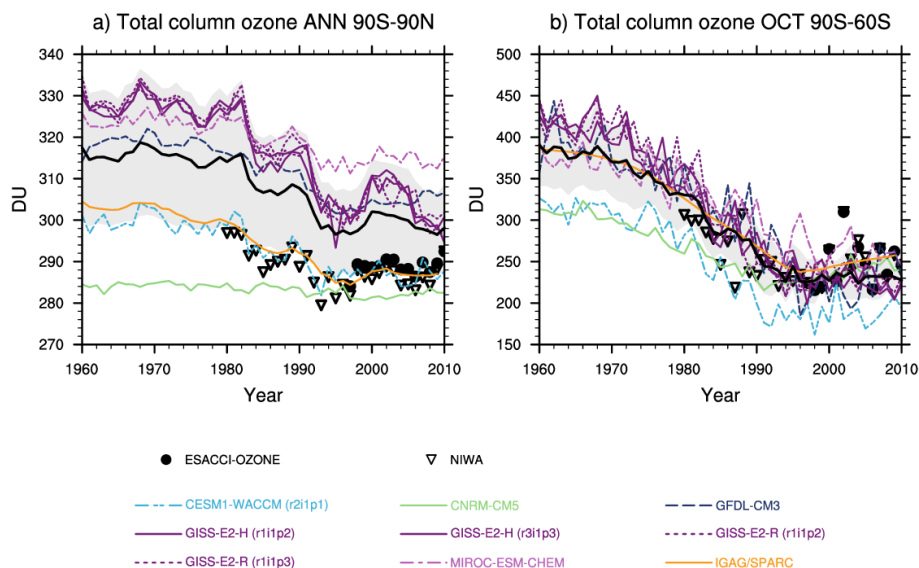


Figure 15: Time series of area-weighted total column ozone from 1960 to 2010 for a) global annual mean (90°S-90°N) and b) Antarctic October mean (60°S-90°S). The figure shows the multi-model mean (black line) and standard deviation (gray shading) as well as individual CMIP5 models with interactive chemistry (colored lines) compared with ESA CCI (filled circles) and NIWA (open triangles) data (Lauer et al. 2017, Fig. 16).

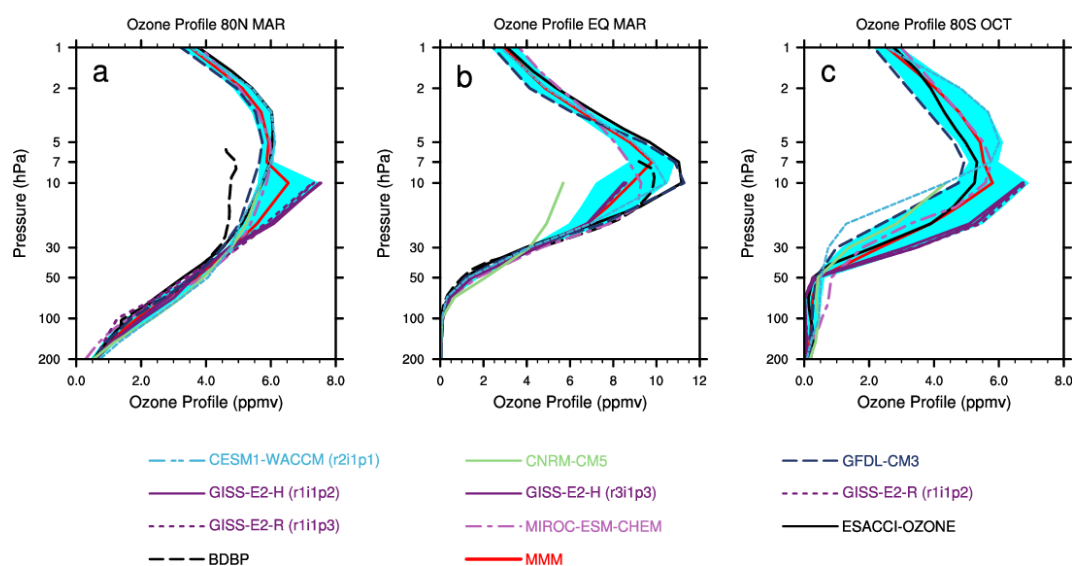


Figure 16: Vertical ozone profile climatologies (2007-2008) at a) 80°N in March, b) the equator in March, and c) at 80°S in October from individual CMIP5 models with interactive chemistry (colored lines) and the ESA CCI ozone data set (solid black line). The multi-model mean (MMM) is shown as a red solid line with one standard deviation of the inter-model spread shown as the light-blue shaded area (Lauer et al. 2017, Fig. 17).

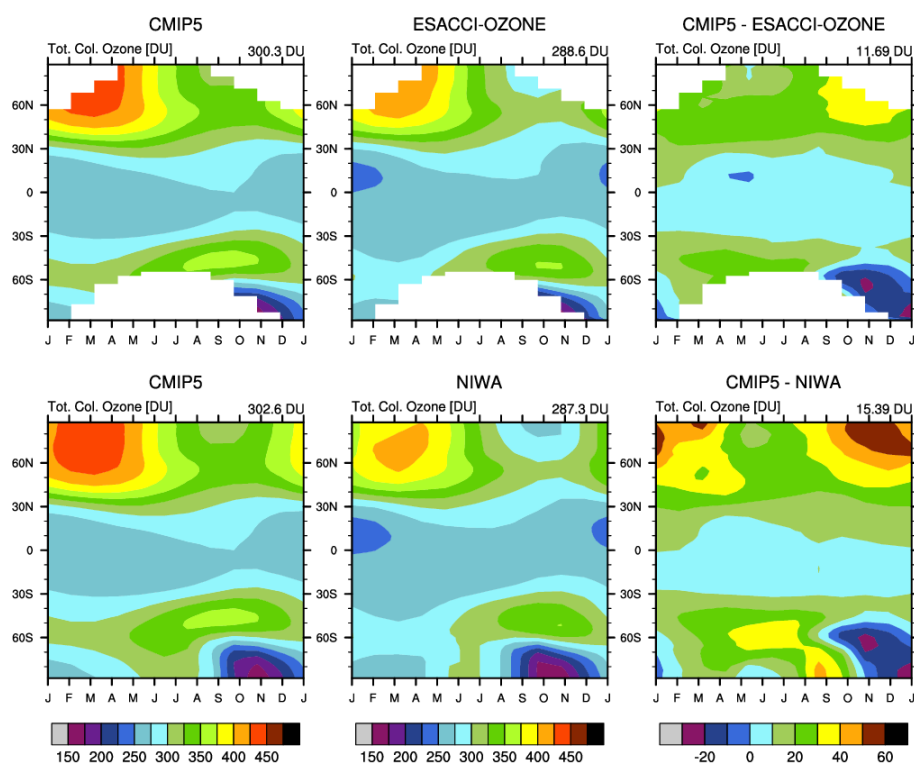


Figure 17: Total column ozone climatologies (1997-2010) for (upper row, from left to right) the multi-model mean of CMIP5 models with interactive chemistry, the ESA CCI ozone data set, and the differences between the CMIP5 multi-model mean and the ESA CCI ozone data. The lower row shows the same plots but for the NIWA combined total column ozone data (Lauer et al. 2017, Fig. 18).

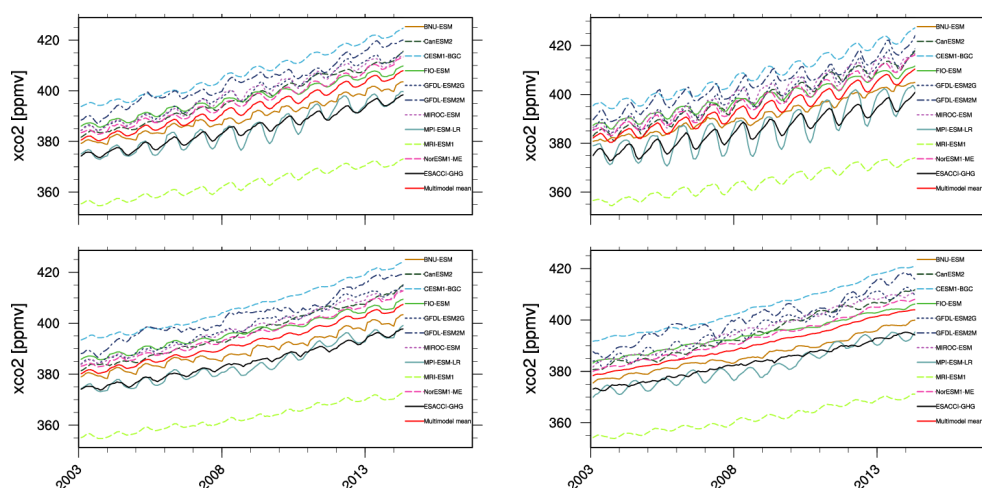


Figure 18: Time series of column averaged carbon dioxide (XCO_2) from 2003 to 2014 from the CMIP5 emission driven simulations for the historical period (2003 to 2005) extended with RCP8.5 simulations (from 2006 to 2014) in comparison with the ESA CCI GHG XCO_2 data (Lauer et al. 2017, Fig. 19).

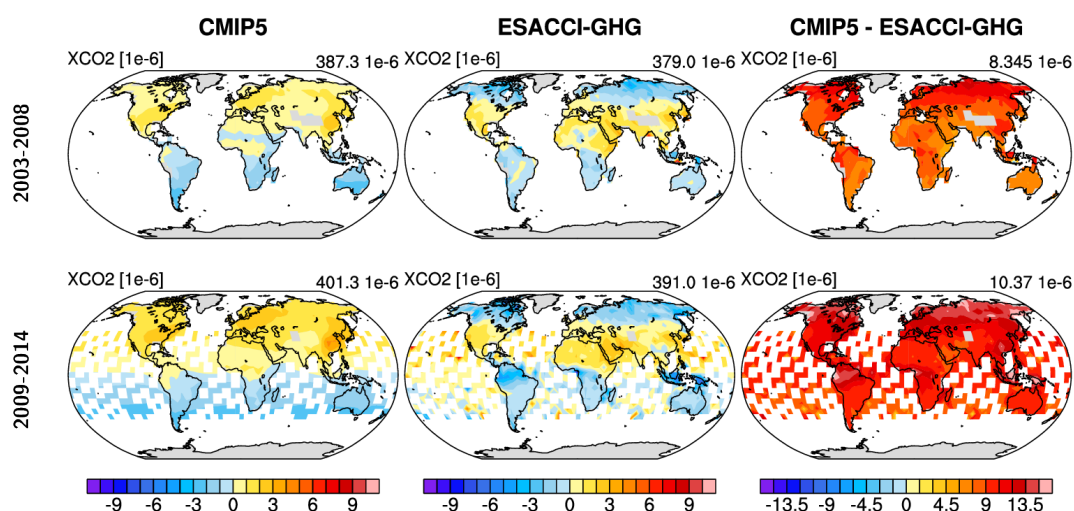


Figure 19: Annual mean XCO_2 climatologies averaged over the years 2003-2008 (top row) and over the years 2009-2014 (bottom row). Shown are deviations from the global annual mean (printed in the right above each panel) for (left) the CMIP5 multi-model mean and (middle) ESA CCI XCO_2 . The right panels show the absolute differences between the CMIP5 multi-model mean and ESA CCI XCO_2 data (Lauer et al. 2017, Fig. 20).

26.1 Overview

This diagnostic produces monthly climatologies of evapotranspiration on a global scale. The model outputs are compared to the LandFlux-EVAL data set (Mueller et al., 2013), which is used as reference product for evapotranspiration. An example of application of this diagnostic on CMIP5 models is presented in Mueller and Seneviratne (2014).

26.2 Available namelists and diagnostics

Namelists are stored in nml/

- namelist_Evapotranspiration.xml

Diagnostics are stored in diag_scripts/

- Evapotranspiration.ncl

26.3 User settings

User setting files (cfg files) are stored in nml/cfg_et/

- Evapotranspiration.ncl

Required diag_script_info attributes

- et_colormap: filename of color table
- cn_levels_mean_basic: contour levels of absolute values (array)
- cn_levels_mean_diff_basic: contour levels of differences
- sftlf_cutoff: threshold for land sea mask (> 50 land grid cell)

Color tables

- diag_scripts/aux/Evapotranspiration/absolute_evapotranspiration.rgb

26.4 Variables

- hfls (atmos, monthly mean, longitude latitude time)

26.5 Observations and reformat scripts

Note: see also header of reformat script for downloading instructions.

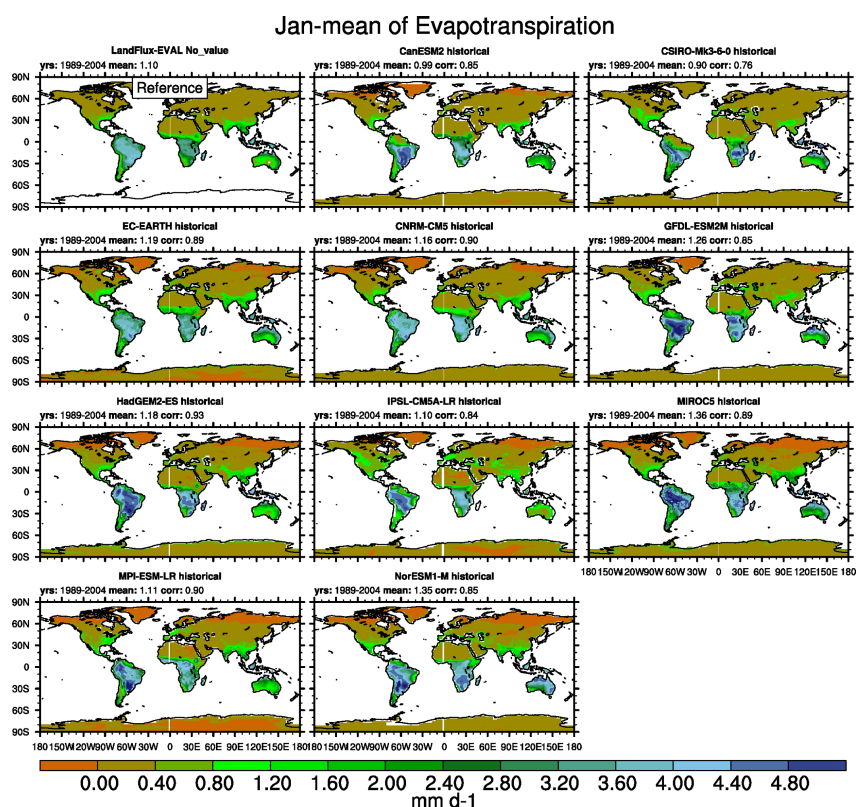
- The LandFlux-EVAL data set (<http://www.iac.ethz.ch/groups/seneviratne/research/LandFlux-EVAL>). The data set is document in Mueller et al (2013). Download data at the above link, then follow the instructions in the reformat script available in the repository. The data set is free to use after registration but may not be redistributed.

Reformat script: reformat_scripts/obs/reformat_obs_LandFlux-EVAL.ncl

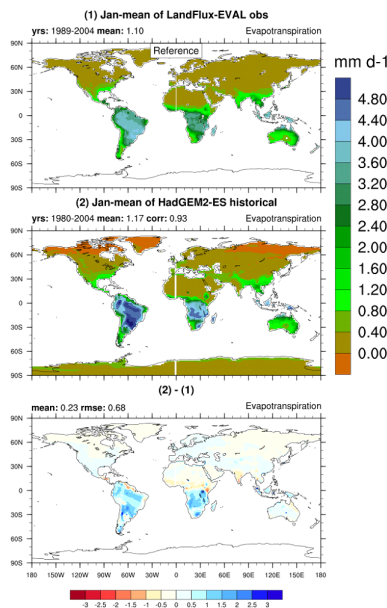
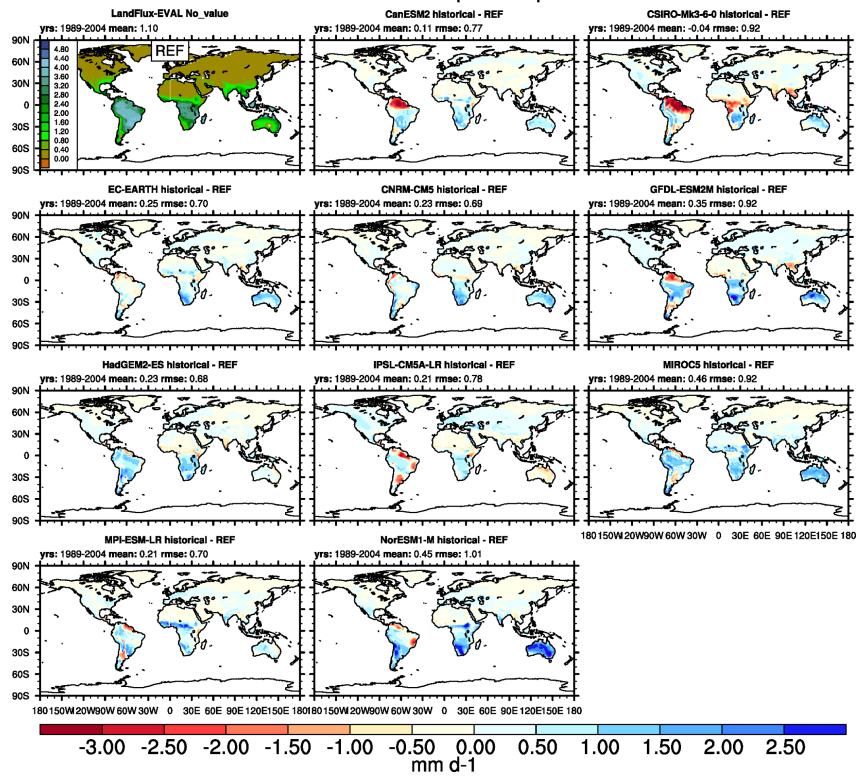
26.6 References

- Mueller, B., and S. I. Seneviratne (2014), Systematic land climate and evapotranspiration biases in CMIP5 simulations, *Geophys. Res. Lett.*, 41, 128-134, doi: 10.1002/2013GL058055.
- Mueller, B., Hirschi, M., Jimenez, C., Ciais, P., Dirmeyer, P. A., Dolman, A. J., Fisher, J. B., Jung, M., Ludwig, F., Maignan, F., Miralles, D. G., McCabe, M. F., Reichstein, M., Sheffield, J., Wang, K., Wood, E. F., Zhang, Y., and Seneviratne, S. I. (2013), Benchmark products for land evapotranspiration: LandFlux-EVAL multi-data set synthesis, *Hydrol. Earth Syst. Sci.*, 17, 3707-3720, doi: 10.5194/hess-17-3707-2013.

26.7 Example plots



Jan-mean of Evapotranspiration



27.1 Overview

The goal of this effort is to code up routines to reproduce Chapter 9 of AR5, so that the plots can be readily reproduced and compared to previous CMIP versions. In this way we can next time start with what was available in the previous round and can focus on developing more innovative methods of analysis rather than constantly having to “re-invent the wheel”.

The plots will be done based on a collection of individual namelists. The following figures from Flato et al. (2013) can currently be reproduced:

- Figure 9.2 a,b,c: Annual-mean surface air temperature for the period 1980-2005. a) multi-model mean, b) bias as the difference between the CMIP5 multi-model mean and the climatology from ERA-Interim (Dee et al., 2011), c) mean absolute model error with respect to the climatology from ERA-Interim.
- Figure 9.4: Annual-mean precipitation rate (mm day^{-1}) for the period 1980-2005. a) multi-model mean, b) bias as the difference between the CMIP5 multi-model mean and the climatology from the Global Precipitation Climatology Project (Adler et al., 2003), c) difference between the multi-model mean and the ECMWF reanalysis of the seasonality, and d) difference between the multi-model mean and the ERA-Interim absolute seasonality.
- Figure 9.5: Climatological (1985-2005) annual-mean cloud radiative effects in Wm^{-2} for the CMIP5 models against CERES EBAF (2001-2011) in Wm^{-2} . Top row shows the shortwave effect; middle row the longwave effect, and bottom row the net effect. Multi-model-mean biases against CERES EBAF 2.6 are shown on the left, whereas the right panels show zonal averages from CERES EBAF 2.6 (black), the individual CMIP5 models (thin gray lines), and the multi-model mean (thick red line).
- Figure 9.7: Relative space-time root-mean square error (RMSE) calculated from the 1980-2005 climatological seasonal cycle of the CMIP5 historical simulations. A relative performance is displayed, with blue shading indicating performance being better and red shading worse, than the median of all model results. A diagonal split of a grid square shows the relative error with respect to the reference data set (lower right triangle) and the alternate data set (upper left triangle). White boxes are used when data is not available for the given model and variable or no alternate data set has been used. The figure shows that performance varies across CMIP5 models and variables, with some models comparing better with observations for one variable and another model performing better for a different variable.
- Figure 9.8: Observed and simulated time series of the anomalies in annual and global mean surface temperature. All anomalies are differences from the 1961-1990 time-mean of each individual time series. The reference period 1961-1990 is indicated by yellow shading; vertical dashed gray lines represent times of

major volcanic eruptions. Single simulations for CMIP5 models (thin lines); multi-model mean (thick red line); different observations (thick black lines).

- Figure 9.10: Total column ozone time series for (a) annual global and (b) Antarctic October mean. CMIP5 models are shown in colored lines and the multi-model mean in thick black, their standard deviation as gray shaded area, and observations (NIWA) (black symbols).
- Figure 9.24 a,b: Time series (1960-2005) of (a) September mean Arctic and (b) February mean Antarctic sea ice extent from the CMIP5 historical simulations. The CMIP5 ensemble mean is highlighted in dark red and the individual ensemble members of each model (colored lines) are shown in different line styles. The model results are compared to observations from the NSIDC (1978-2011, black solid line) and the Hadley Centre Sea ice and Sea Surface Temperature (HadISST, 1978-2011, black dashed line).
- Figure 9.28: Multi-year average aerosol optical depth (AOD) in comparison with satellite data.
- Figure 9.29: Time series of global oceanic mean aerosol optical depth (AOD) from individual CMIP5 models historical (1850-2005) and RCP 4.5 (2006-2010) simulations compared with MODIS and ESACCI-AEROSOL satellite data.
- Figure 9.30: Composite diurnal cycle of precipitation averaged over land / ocean for different latitude bands / seasons (June-July-August (JJA), December-January-February (DJF), or their sum) at local time.
- Figure 9.31: (a, b) The two leading Empirical Orthogonal Functions (EOFs) of outgoing longwave radiation (OLR). The 20- to 100-day filtered OLR from observations and each of the CMIP5 historical simulations is projected on these two leading EOFs to obtain MJO Principal Component time series. The scatterplot (c) shows the maximum positive correlation between the resulting MJO Principal Components and the time lag at which it occurred for all winters (November to March). The maximum positive correlation is an indication of the coherence with which the MJO convection propagates from the Indian Ocean to the Maritime Continent/western Pacific, and the time lag is approximately one fourth of the period of the MJO.
- Figure 9.32: Monsoon precipitation intensity (upper panels) and monsoon precipitation domain (lower panels) for TRMM and an example of deviations from observations from three CMIP5 models (EC-Earth, HadGEM2-ES, and GFDL-ESM2M).
- Figure 9.35: Maximum entropy power spectra of surface air temperature averaged over the NINO3 region for the CMIP5 models and. The vertical lines correspond to periods of two and seven years.
- Figure 9.36: ENSO metrics for pre-industrial control simulations in CMIP3 and CMIP5. (a) and (b): SST anomaly standard deviation (deg C) in Nino 3 and Nino 4, respectively, (c) precipitation response (standard deviation, mm/day) in Nino4.
- Figure 9.45: The carbon cycle-climate feedback (γ_{LT}) versus the short-term sensitivity of atmospheric CO_2 to interannual temperature variability (γ_{IAV}) in the tropics for CMIP5 models. The red line shows the best fit line across the CMIP5 simulations and the vertical dashed lines show the observed range of γ_{IAV} . (b) Probability distribution function (PDF) for γ_{LT} . The solid line is derived after applying the interannual variability (IAV) constraint to the models while the dashed line is the prior PDF derived purely from the models before applying the IAV constraint.

27.2 Available namelists and diagnostics

Namelists are stored in nml/

- namelist_flato13ipcc.xml

Diagnostics are stored in diag_scripts/

- aerosol_satellite.ncl (Fig. 9.28: Multi-year average AOD in comparison with satellite data)
- carbon_constraint.ncl (fig. 9.45b: Emergent constraints on carbon cycle feedbacks)
- carbon_corr_2var.ncl (fig. 9.45b: Emergent constraints on carbon cycle feedbacks)
- carbon_dummy.ncl (fig. 9.45b: Emergent constraints on carbon cycle feedbacks)
- clouds_bias.ncl (figs. 9.2 a,b,c + 9.4 a,b: Clouds)

- clouds_ipcc.ncl (fig. 9.5: Clouds)
- create_co2flux.ncl (fig. 9.45b: Emergent constraints on carbon cycle feedbacks)
- enso_mem.ncl (fig. 9.35: Maximum entropy power spectra of surface air temperature averaged over the NINO3 region)
- eyring13jgr_fig02.ncl (fig. 9.10: Ozone and associated climate impacts)
- ipcc-fig-9-30.ncl (fig. 9.30: Composite diurnal cycle of precipitation)
- ipcc-fig-9-31.ncl (fig. 9.31: Two leading Empirical Orthogonal Functions (EOFs) of the observed outgoing longwave radiation (OLR).)
- ipcc-fig-9-36.ncl (fig. 9.36: (Anomaly) standard deviation averaged over Nino 3 and Nino 4 regions.)
- perfmetrics_grading.ncl (fig. 9.7: Performance metrics for essential climate parameters)
- perfmetrics_grading_collect.ncl (fig. 9.7: Performance metrics for essential climate parameters)
- perfmetrics_main.ncl (fig. 9.7: Performance metrics for essential climate parameters)
- SAMonsoon_precip_domain.ncl (fig. 9.32: South Asian Summer Monsoon diagnostics)
- SeaIce_tsline.ncl (fig. 9.24 a,b: Sea ice)
- seasonality_mm.ncl (fig. 9.30: Seasonality)
- tsline.ncl (fig. 9.29: Aerosol; fig. 9.45: Emergent constraints on carbon cycle feedbacks)
- tsline_IPCC_Fig_9_8.ncl (fig. 9.8: Time series of anomalies of annual and global surface temperature)

27.3 User settings

See individual diagnostics (Annex C):

- Aerosol
- Clouds
- Emergent constraints on carbon cycle feedbacks
- Ozone and associated climate impacts
- Performance metrics for essential climate parameters
- Sea ice
- South Asian Summer Monsoon diagnostics

27.4 Variables

- co2 (atmos, monthly mean, longitude, latitude, plev, time)
- fgco2 (ocean, monthly mean, longitude, latitude, time)
- hus (atmos, monthly mean, longitude, latitude, level, time)
- nbp (land, monthly mean, longitude, latitude, time)
- od550aer (aero, monthly mean, longitude, latitude, time)
- pr (atmos, 3-hr/monthly mean, longitude, latitude, time)
- rlut, rlutcs (atmos, monthly mean, longitude, latitude, time)
- rsut, rsutcs (atmos, monthly mean, longitude, latitude, time)
- sic (ocean, monthly mean, longitude, latitude, time)

- ta (atmos, monthly mean, longitude, latitude, level, time)
- tas (atmos, monthly mean, longitude, latitude, time)
- toz (atmos, monthly mean, longitude, latitude, time)
- tro3 (atmos, monthly mean, longitude, latitude, level, time)
- ua (atmos, monthly mean, longitude, latitude, level, time)
- va (atmos, monthly mean, longitude, latitude, level, time)
- zg (atmos, monthly mean, longitude, latitude, level, time)

27.5 Observations and reformat scripts

Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

- AIRS L3 (hus – obs4mips)
- CERES-EBAF (rlut, rlutcs, rsut, rsutcs – obs4mips)
- CMAP (pr – reformat_scripts/obs/reformat_obs_CMAP.ncl)
- ERA-Interim (tas, ta, ua, va, zg, hus – reformat_scripts/obs/reformat_obs_ERA-Interim.ncl)
- ESACCI-AEROSOL (od550aer – reformat_scripts/obs/reformat_obs_ESACCI-AEROSOL.ncl)
- GCP – Global carbon budget including land (nbp) and ocean (fgco2) carbon fluxes (reformat_scripts/obs/reformat_obs_GCP.ncl)
- GPCP-SG (pr – obs4mips)
- HadISST (sic – reformat_scripts/obs/reformat_obs_HadISST.ncl)
- MERRA (pr – obs4mips)
- MODIS-L3 (od550aer – obs4mips)
- NCEP (tas, ta, ua, va, zg – reformat_scripts/obs/reformat_obs_NCEP.ncl)
- NIWA (toz – reformat_scripts/obs/reformat_obs_NIWA.ncl)
- NSIDC-NT (sic – reformat_scripts/obs/reformat_obs_NSIDC.ncl)
- TRMM-3B42 (pr – reformat_scripts/obs/reformat_obs_TRMM-3B42-3hourly.ncl)

27.6 References

- Adler, R. F., Huffman, G. J., Chang, A., Ferraro, R., Xie, P.-P., Janowiak, J., Rudolf, B., Schneider, U., Curtis, S., Bolvin, D., Gruber, A., Susskind, J., Arkin, P., and Nelkin, E.: The Version-2 Global Precipitation Climatology Project (GPCP) Monthly Precipitation Analysis (1979-Present), *J Hydrometeorol*, 4, 1147-1167, 2003.
- Dee, D. P., Uppala, S. M., Simmons, A. J., Berrisford, P., Poli, P., Kobayashi, S., Andrae, U., Balmaseda, M. A., Balsamo, G., Bauer, P., Bechtold, P., Beljaars, A. C. M., van de Berg, L., Bidlot, J., Bormann, N., Delsol, C., Dragani, R., Fuentes, M., Geer, A. J., Haimberger, L., Healy, S. B., Hersbach, H., Holm, E. V., Isaksen, I., Kallberg, P., Kohler, M., Matricardi, M., McNally, A. P., Monge-Sanz, B. M., Morcrette, J. J., Park, B. K., Peubey, C., de Rosnay, P., Tavolato, C., Thepaut, J. N., and Vitart, F.: The ERA-Interim reanalysis: configuration and performance of the data assimilation system, *Q J Roy Meteor Soc*, 137, 553-597, 2011.
- Eyring, V., Righi, M., Lauer, A., Evaldsson, M., Wenzel, S., Jones, C., Anav, A., Andrews, O., Cionni, I., Davin, E. L., Deser, C., Ehbrecht, C., Friedlingstein, P., Gleckler, P., Gottschaldt, K.-D., Hagemann, S., Jukes, M., Kindermann, S., Krasting, J., Kunert, D., Levine, R., Loew, A., Mäkelä, J. Martin, G., Mason, E., Phillips, A. S., Read, S., Rio, C., Roehrig, R., Senfleben, D., Sterl, A., van Ulft, L. H., Walton,

J., Wang, S., and Williams, K. D.: ESMValTool (v1.0) a community diagnostic and performance metrics tool for routine evaluation of Earth System Models in CMIP, *Geosci. Model Dev.*, 9, 1747-1802, doi: 10.5194/gmd-9-1747-2016, 2016.

- Flato, G., J. Marotzke, B. Abiodun, P. Braconnot, S.C. Chou, W. Collins, P. Cox, F. Driouech, S. Emori, V. Eyring, C. Forest, P. Gleckler, E. Guilyardi, C. Jakob, V. Kattsov, C. Reason and M. Rummukainen, 2013: Evaluation of Climate Models. In: *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change* [Stocker, T.F., D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex and P.M. Midgley (eds.)]. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.

27.7 Example plots

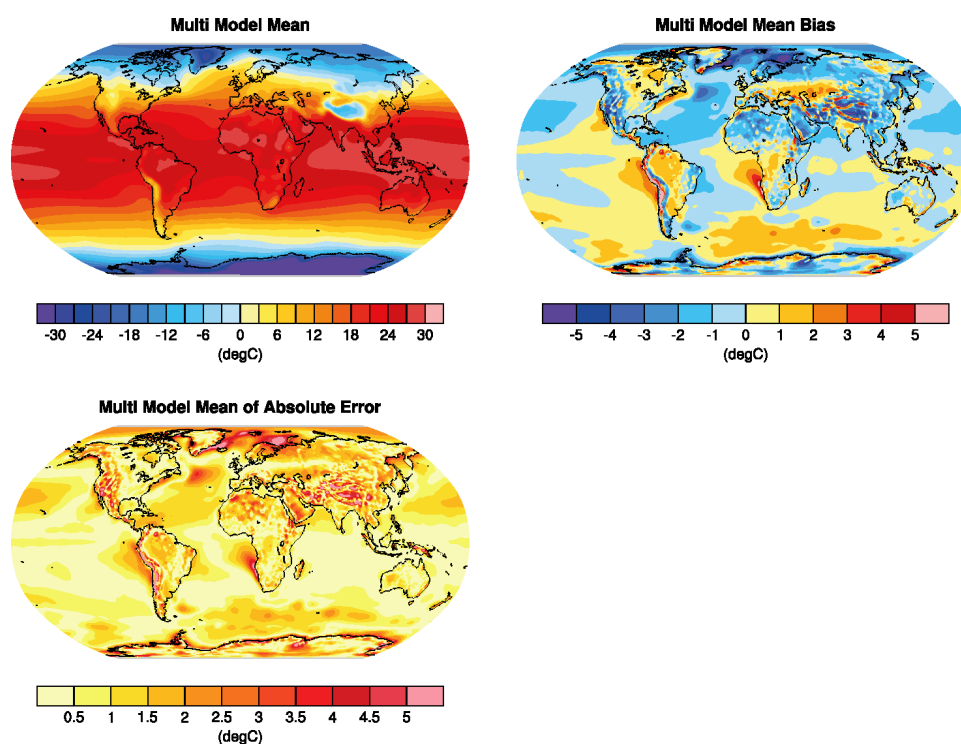


Figure 1: Resembling Flato et al. (2013), Fig. 9.2a,b,c.

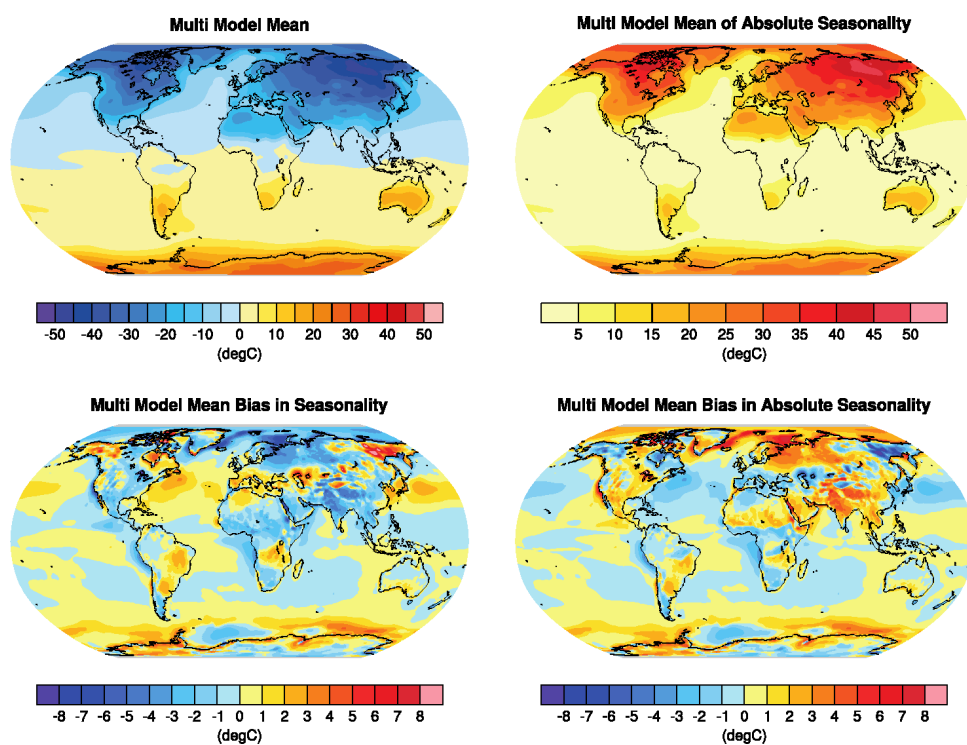


Figure 2: Resembling Flato et al. (2013), Fig. 9.3.

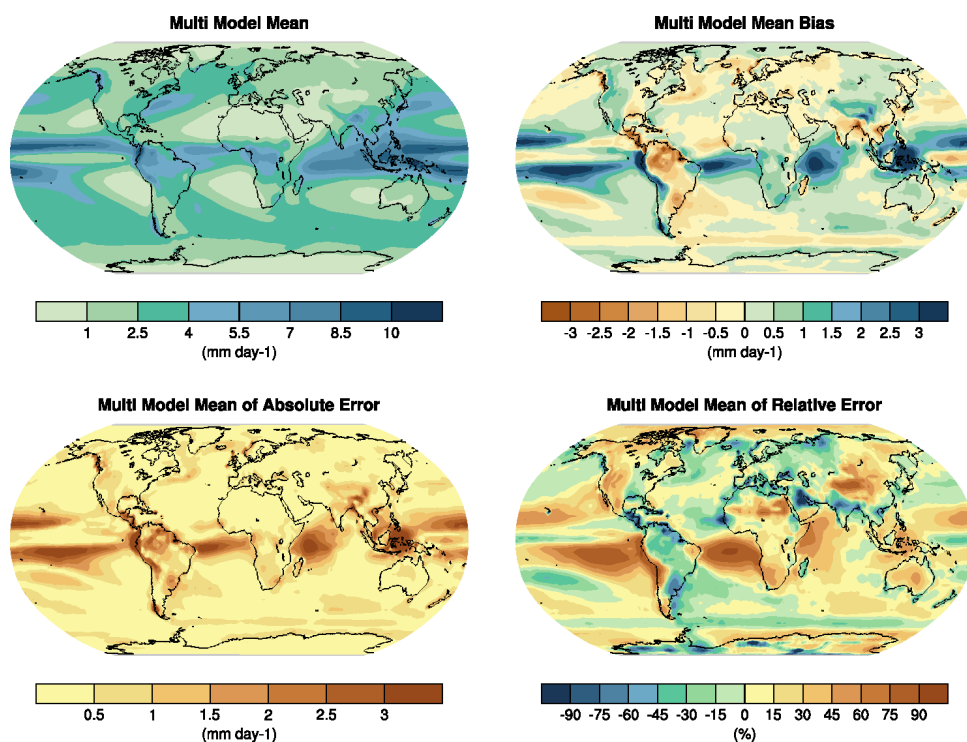
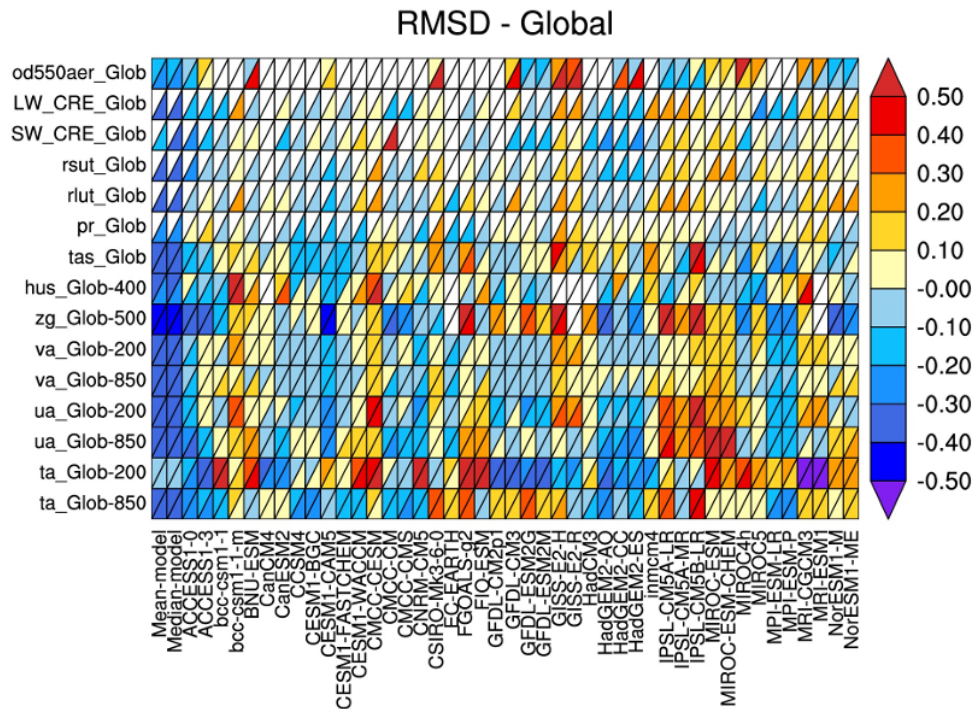


Figure 3: Resembling Flato et al. (2013), Fig. 9.4.



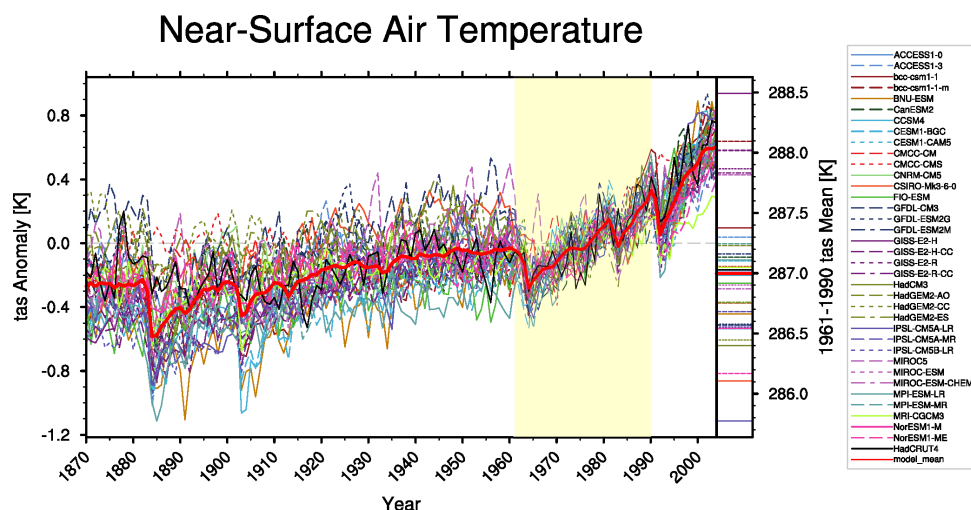


Figure 6: Resembling Flato et al. (2013), Fig. 9.8.

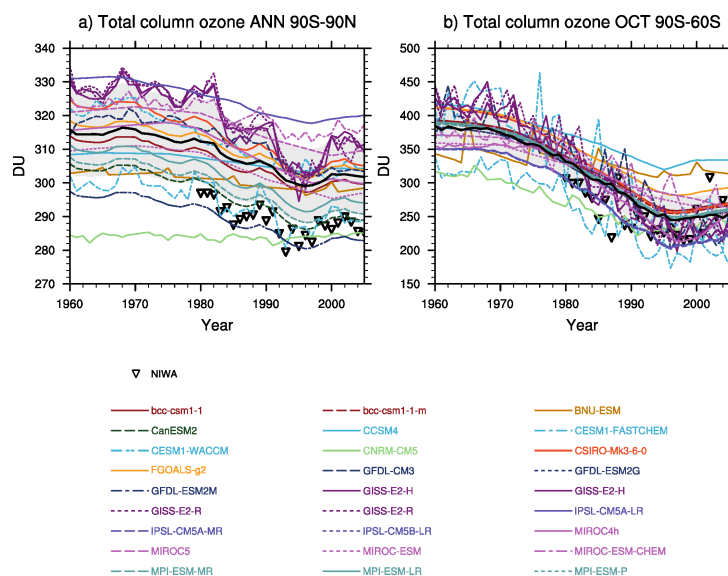


Figure 7: Resembling Flato et al. (2013), Fig. 9.10 (Eyring et al., 2016: Fig. 25).

September Arctic Sea Ice Extent

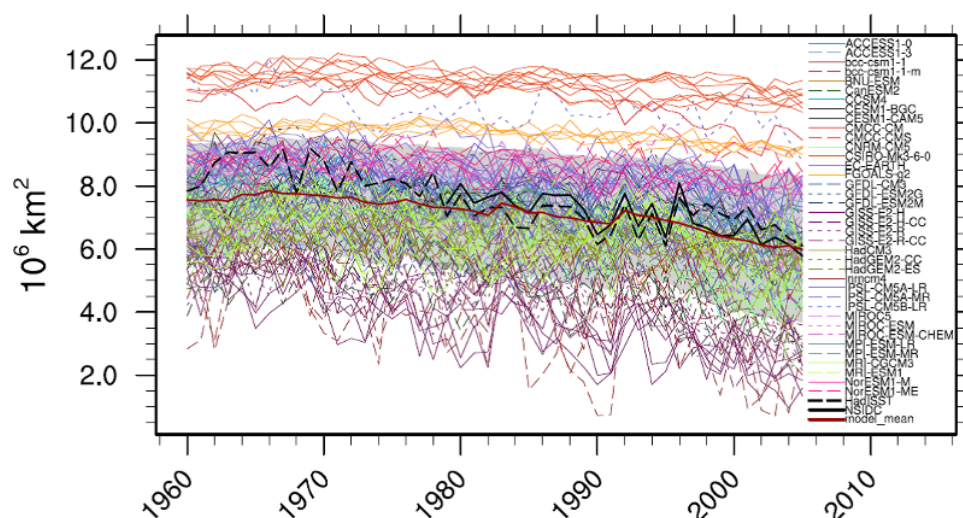


Figure 8: Resembling Flato et al. (2013), Fig. 9.24 (Eyring et al., 2016: Fig. 17).

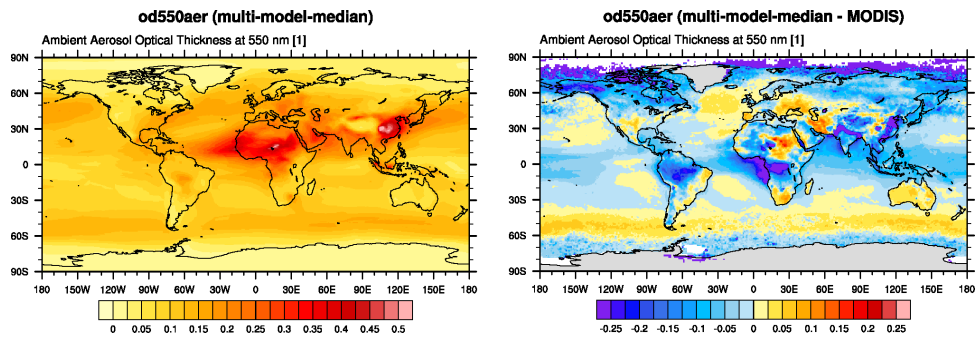


Figure 9: Similar to Flato et al. (2013), Fig. 9.28.

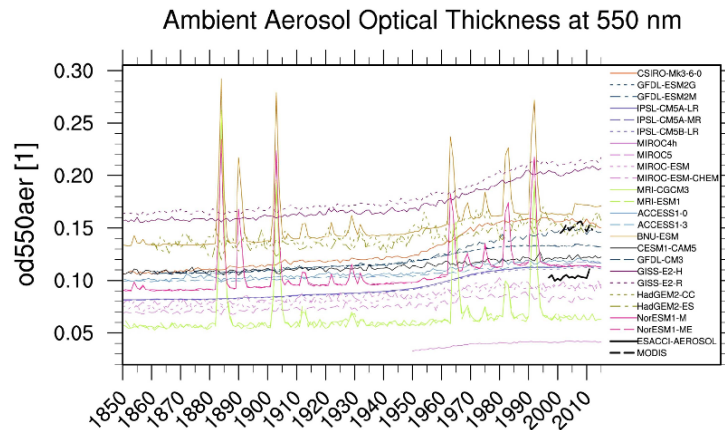


Figure 10: Resembling Flato et al. (2013), Fig. 9.29 (Eyring et al., 2016: Fig. 23).

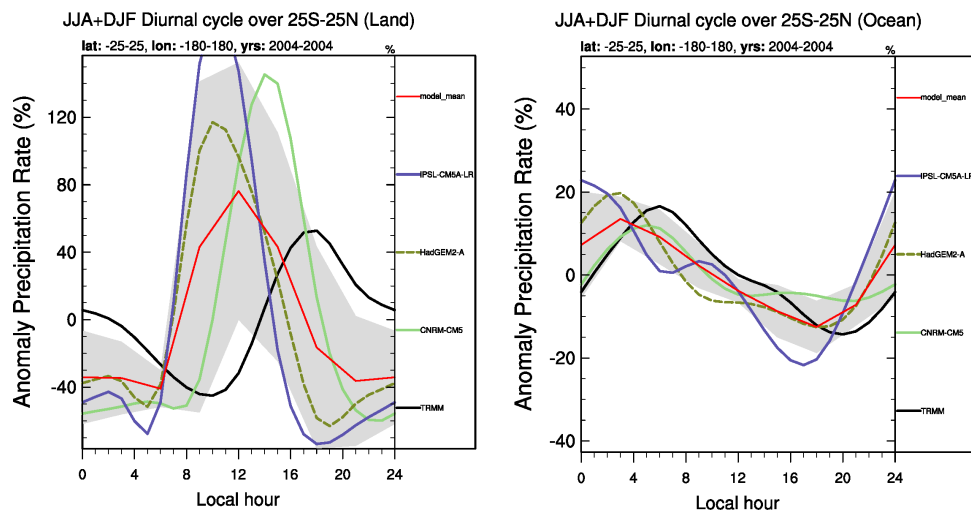


Figure 11: Similar to Flato et al. (2013), Fig. 9.30.

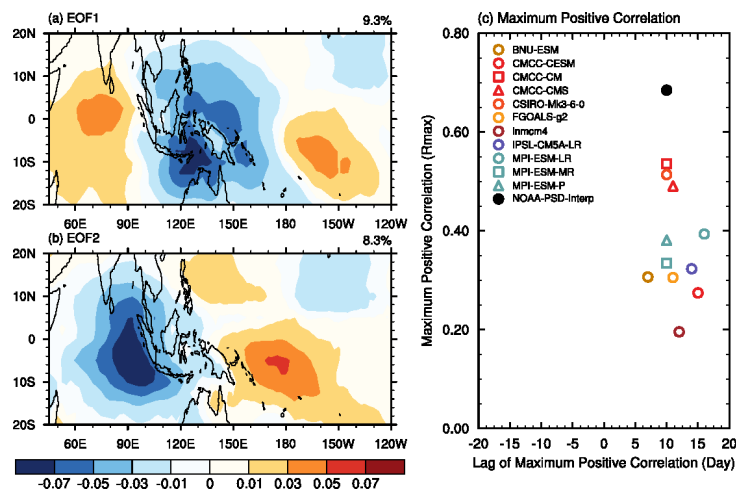


Figure 12: Similar to Flato et al. (2013), Fig. 9.31.

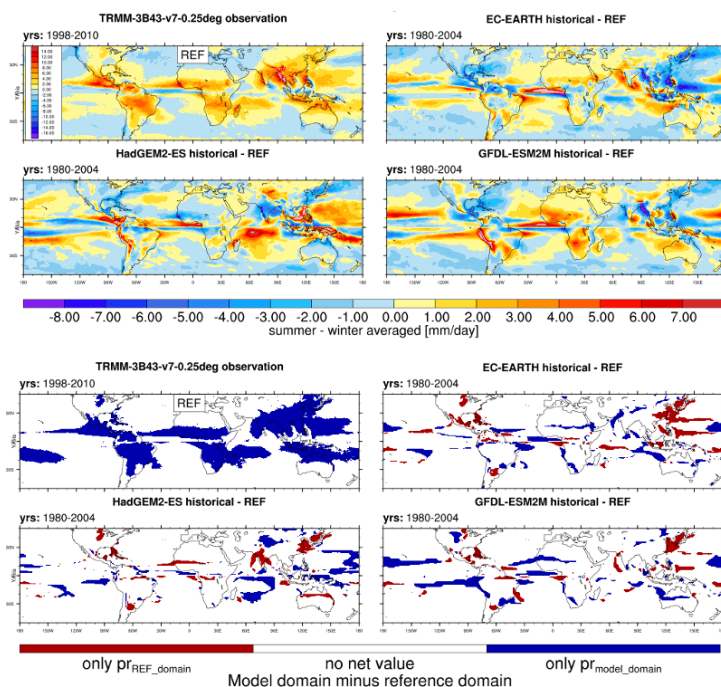


Figure 13: Resembling Flato et al. (2013), Fig. 9.32 (Eyering et al., 2016: Fig. 5).

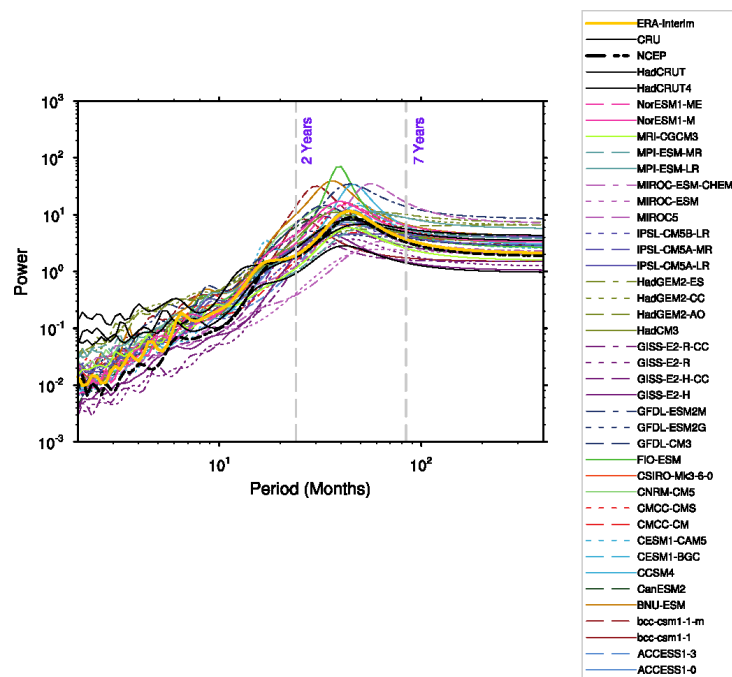


Figure 14: Resembling Flato et al. (2013), Fig. 9.35.

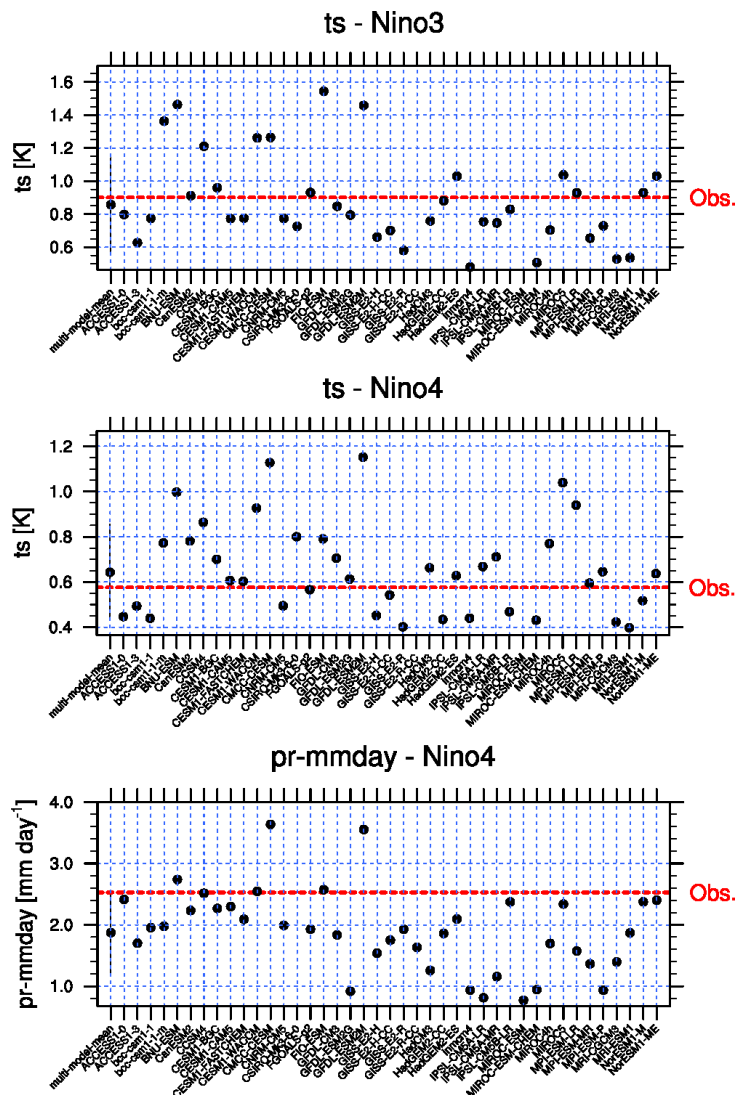


Figure 15: Resembling Flato et al. (2013), Fig. 9.36.

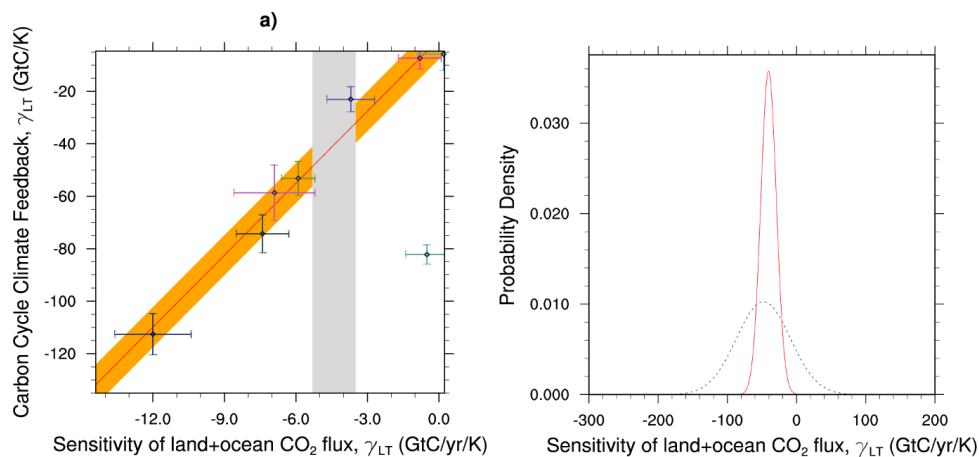


Figure 16: Resembling Flato et al. (2013), Fig. 9.45b (Eyring et al., 2016: Fig. 26).

Land and ocean components of the global carbon cycle

28.1 Overview

This namelist reproduces most of the figures of Anav et al. (2013):

- seasonal cycle plot for different regions
- errorbar plot for different regions showing mean and standard deviation
- scatterplot for different regions showing mean vs. interannual variability
- 3D-scatterplot for different regions showing mean vs. linear trend and the model variability index (MVI) as a third dimension (color coded)
- scatterplot for different regions comparing two variable against each other (cSoil vs. cVeg)

In addition, performance metrics are calculated for all variables using the performance metric diagnostics (see details in section *Performance metrics for essential climate parameters*). This however applies only to variables on a regular grid (i.e., not to fgco2), as irregular grids are not yet supported by the performance “mwtr”.

28.2 Available namelists and diagnostics

Namelists are stored in nml/

- namelist_anav13jclim.xml

Diagnostics are stored in diag_scripts/

- CarbonCycle_main.ncl: calculate temporal and spatial averages and plots the variable as error-bar and seasonal cycle plots.
- CarbonCycle_MVI.ncl: calculate the model variability index (MVI), interannual variability (IAV) and mean, and draw them in a 3D scatter plot.
- CarbonCycle_2vars.ncl: draw a scatter plot with two variables.
- (perfmetrics_main.ncl): see section *Performance metrics for essential climate parameters*
- (perfmetrics_grading.ncl): see section *Performance metrics for essential climate parameters*

28.3 User settings

User setting files (cfg files) are stored in `nml/cfg_anav13jclim/`

1. CarbonCycle_MVI.ncl

Diag_script_info attributes

- `grid`: target grid for regridding (“0.5deg”, “1deg”, “2deg”)
- `region`: region for spatial average (“Global”, “Northern Hemisphere”, “Southern Hemisphere”, “Tropics”)
- `ref_model`: reference model, e.g. “CRU”

2. CarbonCycle_main.ncl

Diag_script_info attributes

- `region`: region for spatial average (“Global”, “Northern Hemisphere”, “Southern Hemisphere”, “Tropics”)
- `legend_outside`: plot legend in a separate file (True, False)
- `styleset`: plot style set (“CMIP5”, “EMAC”, “DEFAULT”)
- `sort`: sort models alphabetically (True, False)
- `seasonal_cycle_plot`: create seasonal cycle plot (True, False)
- `errorbar_plot`: create error bar plot (True, False)
- `mean_IAV_plot`: create mean (x-axis), IAV (y-axis) plot (True, False)

3. CarbonCycle_2vars.ncl

Diag_script_info attributes

- `region`: region for spatial average (“Global”, “Northern Hemisphere”, “Southern Hemisphere”, “Tropics”)
- `legend_outside`: plot legend in a separate file (True, False)
- `styleset`: plot style set (“CMIP5”, “EMAC”, “DEFAULT”)

4. perfmetrics_main.ncl, perfmetric_grading.ncl

Diag_script_info attributes

- see [Section 34](#)

28.4 Variables

- `tas` (atmos, monthly mean, longitude latitude time)
- `pr` (atmos, monthly mean, longitude latitude time)
- `nbp` (lmon, monthly mean, longitude latitude time)
- `gpp` (lmon, monthly mean, longitude latitude time)
- `lai` (lmon, monthly mean, longitude latitude time)
- `cVeg` (lmon, monthly mean, longitude latitude time)
- `cSoil` (lmon, monthly mean, longitude latitude time)
- `fgco2` (omon, monthly mean, longitude latitude time)

28.5 Observations and reformat scripts

Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

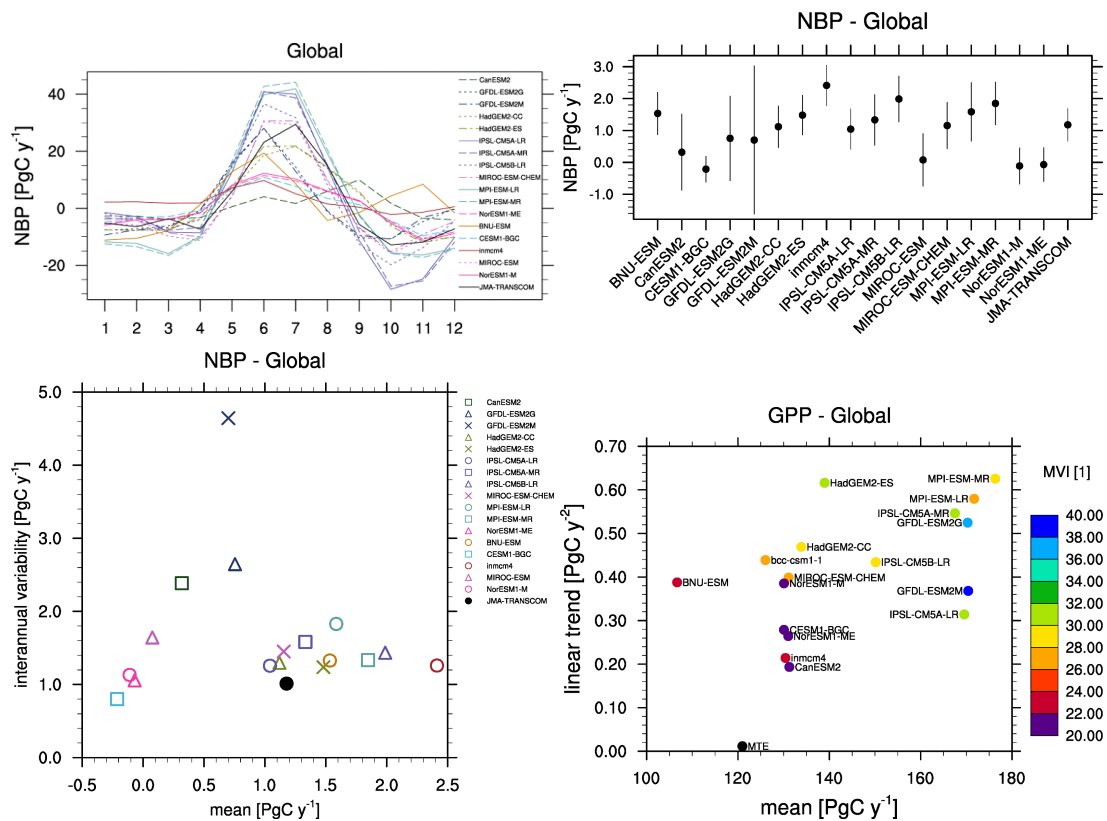
- CRU (near-surface air temperature, precipitation)
Reformat script: reformat_scripts/obs/reformat_obs_CRU.ncl
- MTE (gross primary productivity of carbon)
- LAI3g (leaf area index)
- JMA-TRANSCOM (CO₂ exchange)
- HSWD (soil carbon content)
- NDP (vegetation carbon content)

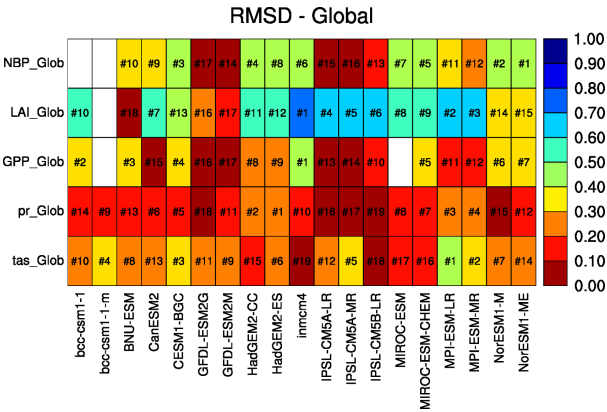
Reformat scripts: The reformat scripts for the following data sets **could not be included in the ESMValTool v1.1**: MTE, LAI3g, JMA-TRANSCOM, HSWD, and NDP. Please contact the author of this diagnostic package (A. Anav) for obtaining the reformat scripts.

28.6 References

- Anav, A. et al.: Evaluating the land and ocean components of the global carbon cycle in the CMIP5 Earth System Models, *J. Climate*, 26, 6901-6843, doi: 10.1175/JCLI-D-12-00417.1, 2013.

28.7 Example plots





Marine biogeochemistry

29.1 Overview

A series of routines are provided to support the evaluation of ocean biogeochemical cycles at global scales as simulated by both ocean-only and coupled climate-carbon cycle models. Diagnostics have been implemented to generate time series line and contour plots for climatological distributions and inter-annual or inter-seasonal (e.g., JJAS) variability of selected ocean biogeochemical variables along with companion plots for differences (“diff”) relative to a chosen reference (observational) data set. Basic benchmarking statistics are also provided (bias, standard deviation, root mean squared error, correlation coefficient).

29.2 Available namelists and diagnostics

Namelists are stored in `nml/`

- `namelist_GlobalOcean.xml`

Diagnostics are stored in `diag_scripts/`

- `GO_comp_map.ncl`: this diagnostic takes input from up to four models, one of which is defined as the reference, to produce contour maps of the time-mean (climatology) and standard deviation (inter-annual or inter-seasonal [e.g., JJAS] variability) together with the difference between the reference and each model. `GO_comp_map.ncl` is based on `SAM_monsoon/SAMonsoon_wind_basic.ncl` and uses many `SAMonsoon` underlying functions. Extensions to these routines include support for the pre-processing and interpolation of curvilinear ocean grids (e.g., ORCA2, NEMO) along with the option to use 3-D (TO3M; depth-resolving) input model data (ocean) and select a single depth level on which to produce `comp_maps`.
- `GO_tsline.ncl`: this diagnostic (developed from `SeaIce_tsline.ncl`) produces annual and seasonal time series plots from monthly ocean data, and includes reformatting options to convert input irregularly gridded data into a version that NCL can interpret by introducing intermediate referencing co-ordinates in dimensioning the variable, and writing gridcell areas to the same file. Plots can be produced either globally or from a selected latitude-longitude range, with optional extensions including the ability to mask model fields with observational coverage, plot anomaly fields, or overlay derived fields such as linear trend lines, running means or multi-model means. Support for irregular grids (e.g., ORCA2, ORCA1) is provided to, for instance, apply a weighting based on gridcell areas before calculating averages (see `reformat_scripts/GO/`).

29.3 User settings

User setting files (cfg files) are stored in nml/cfg_GO/

1. GO_tsline.ncl

Required diag_script_info attributes

- month: "A" (A = annual mean)
- styleset: "CMIP5" (CMIP5, DEFAULT, GO, EXPT)
- colors = (/ "red", "green", "blue", "yellow", "purple", "black", "gray", "white", "pink", "orange", "brown", "palegreen" /)
- ts_multimean: "True" = plot multi-model mean & stddev; "False" = don't
- ts_latrange: min. and max. latitude for area-averaging of first variable (array)
- ts_lonrange: min. and max. longitude for area-averaging of first variable (array)

Optional diag_info_script attributes

- ts_anom: calculates anomalies with respect to the first 10-year average (anom, noanom)
- ts_ymin: y-axis minimum
- ts_ymax: y-axis maximum
- ts_min_year: start year
- ts_max_year: end year
- ts_smooth: smoothing (True, False)
- ts_trend: overlay trend line (True, False)
- ts_coverage: *not used*

2. GO_comp_map.ncl

Required diag_script_info attributes

- GO_season: season (e.g., "JJASO")
- GO_detrend: detrend data (True, False)
- latrange_basic: min. and max latitude of region (array)
- lonrange_basic: min. and max. longitude of region (array)
- ref_model: name of reference data set
- cn_levels_mean_basic: contour levels for mean (array)
- cn_levels_mean_diff_basic: contour levels for differences (array)
- cn_levels_stddev_basic: contour levels for standard deviation (array)
- cn_levels_stddev_diff_basic: contour levels for difference in standard deviation (array)
- regrid_for_these_projects: list of project names that will be regridded (e.g., "CMIP5_gridfile", "GO_gridfile", "CMIP5_fx") (array)

29.4 Variables

- spco2 (ocnBgchem, monthly mean, longitude latitude time)
- chl (ocnBgchem, monthly mean, longitude latitude time)
- talk (ocnBgchem, monthly mean, longitude latitude time)

- o2 (ocnBgchem, monthly mean, longitude latitude olevel time)

29.5 Observations and reformat scripts

Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

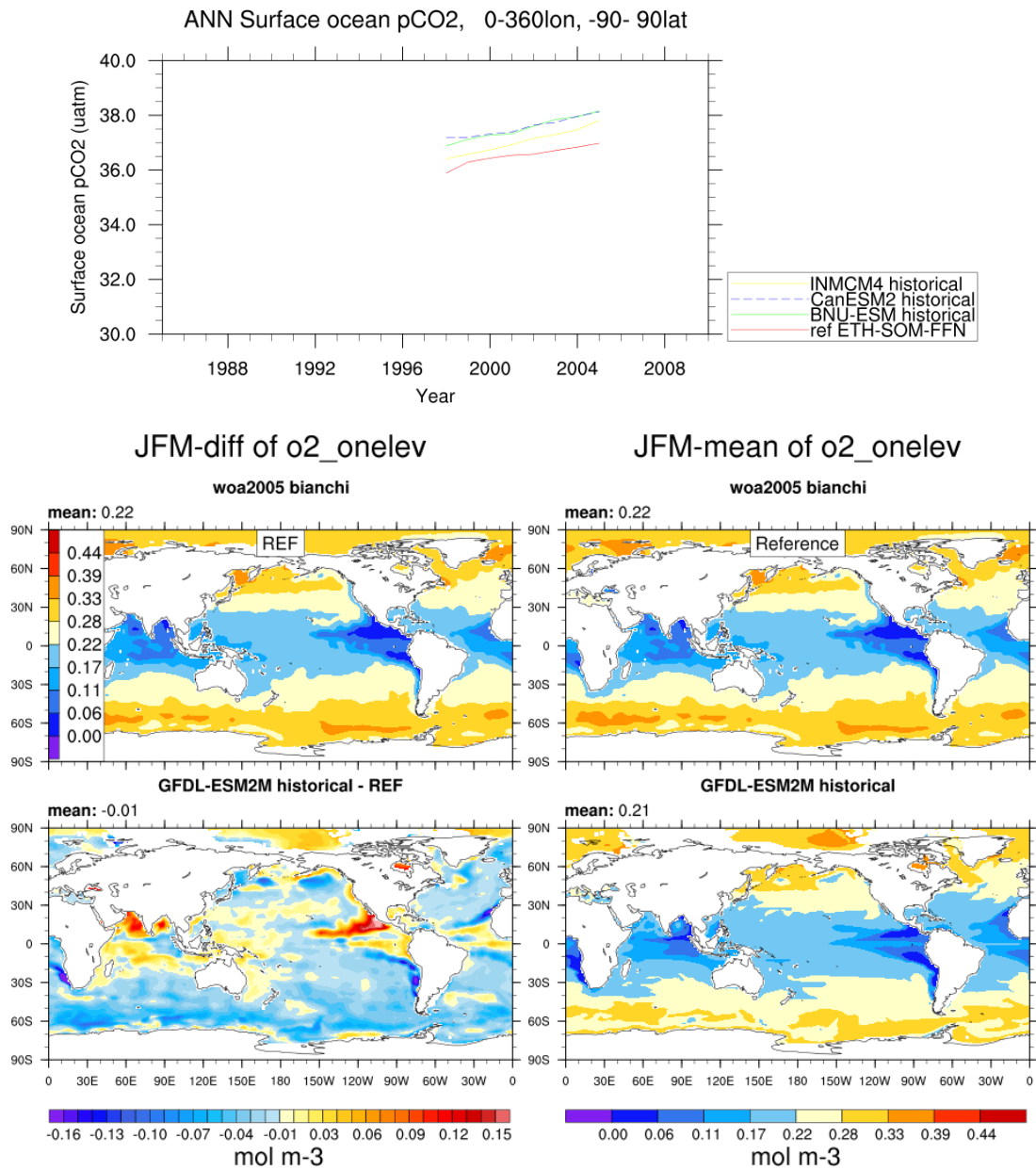
- spco2: Surface Ocean CO₂ Atlas (SOCAT version 2; Bakker et al., 2014)
Reformat script: reformat_scripts/obs/reformat_obs_SOCAT.csh
- spco2: Two Surface Ocean pCO₂ Mapping Intercomparison (SOCOM) products: UEA-SI v1.0 (Jones et al., In Rev.) and ETH-SOM-FFN (Landshüster et al., 2014)
Reformat script: reformat_scripts/obs/reformat_obs_ETH-SOM-FFN.csh
- chl: Chlorophyll concentration data from Sea-viewing Wide Field-of-view Sensor (SeaWiFS; available at: <http://oceancolor.gsfc.nasa.gov/SeaWiFS/>)
Reformat script: reformat_scripts/obs/reformat_obs_SeaWiFS.csh
- o2: World Ocean Atlas (WOA) 2005 dissolved oxygen concentration data with corrections applied as described in Bianchi et al. (2012)
Reformat script: reformat_scripts/obs/reformat_obs_woa2005.csh
- talk: Monthly surface climatological Total Alkalinity (T14; Takahashi et al., 2014)
Reformat script: reformat_scripts/obs/reformat_obs_takahashi14.csh

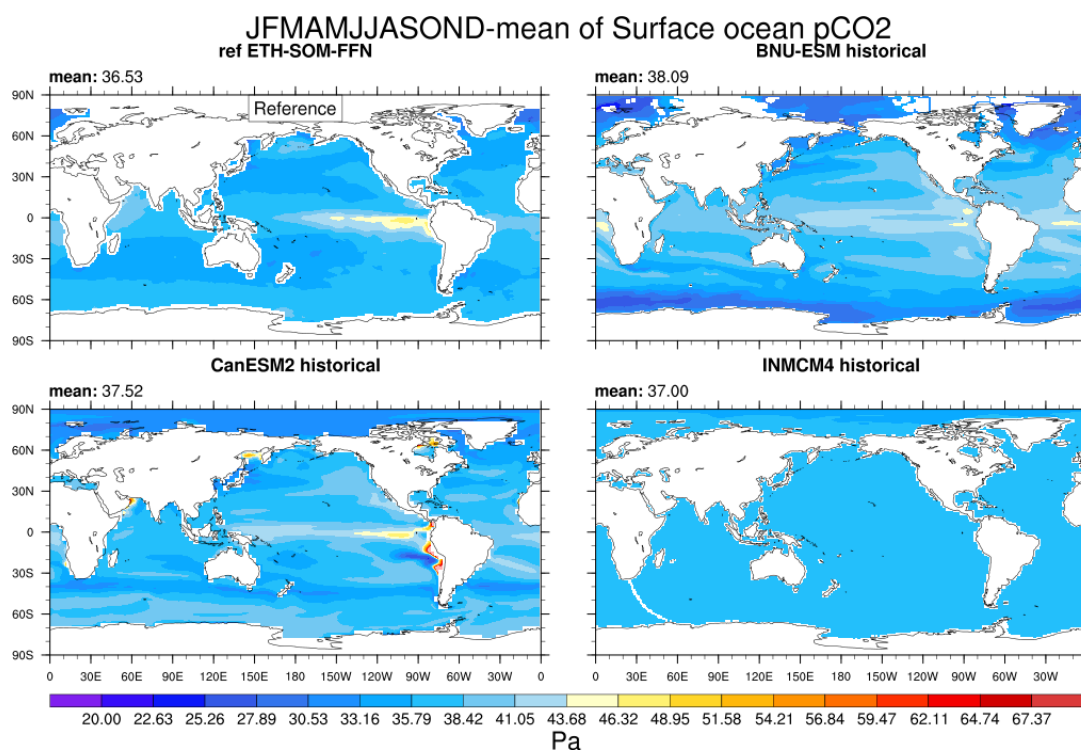
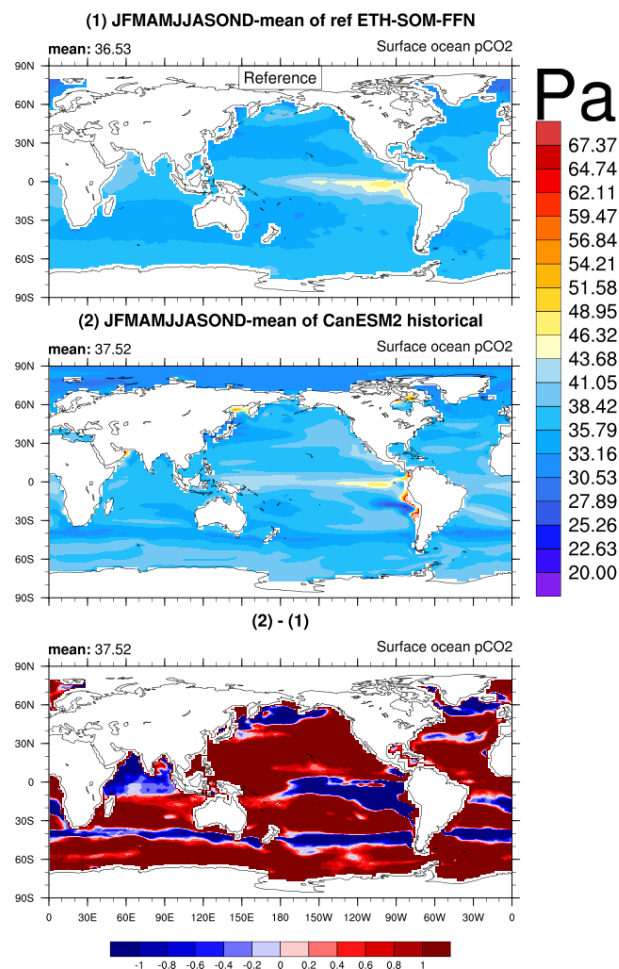
29.6 References

- Bakker, D. C. E., Pfeil, B., Smith, K., Hankin, S., Olsen, A., Alin, S. R., Cosca, C., Harasawa, S., Kozyr, A., Nojiri, Y., O'Brien, K. M., Schuster, U., Telszewski, M., Tilbrook, B., Wada, C., Akl, J., Barbero, L., Bates, N. R., Boutin, J., Bozec, Y., Cai, W.-J., Castle, R. D., Chavez, F. P., Chen, L., Chierici, M., Currie, K., de Baar, H. J. W., Evans, W., Feely, R. A., Fransson, A., Gao, Z., Hales, B., Hardman-Mountford, N. J., Hoppema, M., Huang, W.-J., Hunt, C. W., Huss, B., Ichikawa, T., Johannessen, T., Jones, E. M., Jones, S. D., Jutterström, S., Kitidis, V., Körtzinger, A., Landschützer, P., Lauvset, S. K., Lefèvre, N., Manke, A. B., Mathis, J. T., Merlivat, L., Metzl, N., Murata, A., Newberger, T., Omar, A. M., Ono, T., Park, G.-H., Paterson, K., Pierrot, D., Ríos, A. F., Sabine, C. L., Saito, S., Salisbury, J., Sarma, V. V. S. S., Schlitzer, R., Sieger, R., Skjelvan, I., Steinhoff, T., Sullivan, K. F., Sun, H., Sutton, A. J., Suzuki, T., Sweeney, C., Takahashi, T., Tjiputra, J., Tsurushima, N., van Heuven, S. M. A. C., Vandemark, D., Vlahos, P., Wallace, D. W. R., Wanninkhof, R., and Watson, A. J.: An update to the Surface Ocean CO₂ Atlas (SOCAT version 2), Earth Syst. Sci. Data, 6, 69-90, doi: 10.5194/essd-6-69-2014, 2014.
- Bianchi, D., Dunne, J. P., Sarmiento, J. L., and Galbraith, E. D.: Data-based estimates of suboxia, denitrification, and N₂O production in the ocean and their sensitivities to dissolved O₂, Global Biogeochem. Cy., 26, GB2009, doi: 10.1029/2011GB004209, 2012.
- Jones, S. D., Le Quere, C., Rödenbeck, C., Manning, A. C., and Olsen, A.: A statistical gap-filling method to interpolate global monthly surface ocean carbon dioxide data, J. Adv. Model Earth Syst., in review. Key, R. M., Kozyr, A., Sabine, C. L., Lee, K., Wanninkhof, R., Bullister, J. L., Feely, R. A., Millero, F. J., Mordy, C., and Peng, T.-H.: A global ocean carbon climatology: results from Global Data Analysis Project (GLODAP), Global Biogeochem. Cy., 18, GB4031, doi: 10.1029/2004GB002247, 2004.
- Rödenbeck, C., Keeling, R. F., Bakker, D. C. E., Metzl, N., Olsen, A., Sabine, C., and Heimann, M.: Global surface-ocean pCO₂ and sea-air CO₂ flux variability from an observation-driven ocean mixed-layer scheme, Ocean Sci., 9, 193-216, doi: 10.5194/os-9-193-2013, 2013.
- Takahashi, T., Sutherland, S. C., Chipman, D. W., Goddard, J. G., Ho, C., Newberger, T., Sweeney, C., and Munro, D. R.: Climatological distributions of pH, pCO₂, total CO₂, alkalinity, and CaCO₃ saturation

in the global surface ocean, and temporal changes at selected locations, *Mar. Chem.*, 164, 95-125, doi: 10.1016/j.marchem.2014.06.004, 2014.

29.7 Example plots





Madden-Julian Oscillation (MJO)

30.1 Overview

To assess the main MJO features in ESMs, a namelist with a number of diagnostics developed by the US CLIVAR MJO Working Group (Kim et al., 2009; Waliser et al., 2009) has been implemented in the ESMValTool. These diagnostics are calculated using precipitation (pr), outgoing longwave radiation (OLR) (rlut), eastward (ua) and northward wind (va) at 850 hPa (u850) and 200 hPa (u200) against various observations and reanalysis data sets for boreal summer (May-October) and winter (November-April).

Observation and reanalysis data sets include GPCP-1DD for precipitation, ERA-Interim and NCEP-DOE reanalysis 2 for wind components (Kanamitsu et al., 2001) and NOAA polar-orbiting satellite data for OLR (Liebmann and Smith, 1996). The majority of the scripts are based on example scripts available at <http://ncl.ucar.edu/Applications/mjoclivar.shtml>. Daily data is required for most of the scripts. The basic diagnostics include mean seasonal state and 20-100 day bandpass filtered variance for precipitation and u850 in summer and winter. To better assess and understand model biases in the MJO, a number of more sophisticated diagnostics have also been implemented. These include: univariate empirical orthogonal function (EOF) analysis for 20-100 day bandpass filtered daily anomalies of precipitation, OLR, u850 and u200. To illustrate the northward and eastward propagation of the MJO, lag-longitude and lag-latitude diagrams show either the equatorially (latitude) averaged (10°S-10°N) or zonally (longitude) averaged (80°E-100°E) intraseasonal precipitation anomalies and u850 anomalies correlated against intraseasonal precipitation at an Indian Ocean reference point (75°E-100°E, 10°S-5°N). Similar figures can also be produced for other key variables and regions following the definitions of Waliser et al. (2009). To further explore the MJO intraseasonal variability, the wavenumber-frequency spectra for each season is calculated for individual variables. In addition, we also produce cross-spectral plots to quantify the coherence and phase relationships between precipitation and u850.

30.2 Available namelists and diagnostics

Namelists are stored in nml/

- namelist_mjo_mean_state.xml
- namelist_mjo_daily.xml

Diagnostics are stored in diag_scripts/

- mjo_univariate_eof.ncl
- mjo_wave_freq.ncl

- mjo_precip_u850-200_propagation.ncl
- mjo_precip_uwnd_variance.ncl
- mjo_olr_u850-200_cross_spectra.ncl
- mjo_olr_u850_200_ceof.ncl
- mjo_olr_uv850_ceof_life_cycle.ncl
- mjo_precip_u850_basic_month.ncl

30.3 User settings

User setting files (cfg files) are stored in nml/cfg_mjo/

30.4 Variables

- pr (atmos, daily/monthly mean, longitude latitude time)
- rlut (atmos, daily mean, longitude latitude time)
- ua (atmos, daily/monthly mean, longitude latitude level time)
- va (atmos, daily mean, longitude latitude level time)

30.5 Observations and reformat scripts

Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

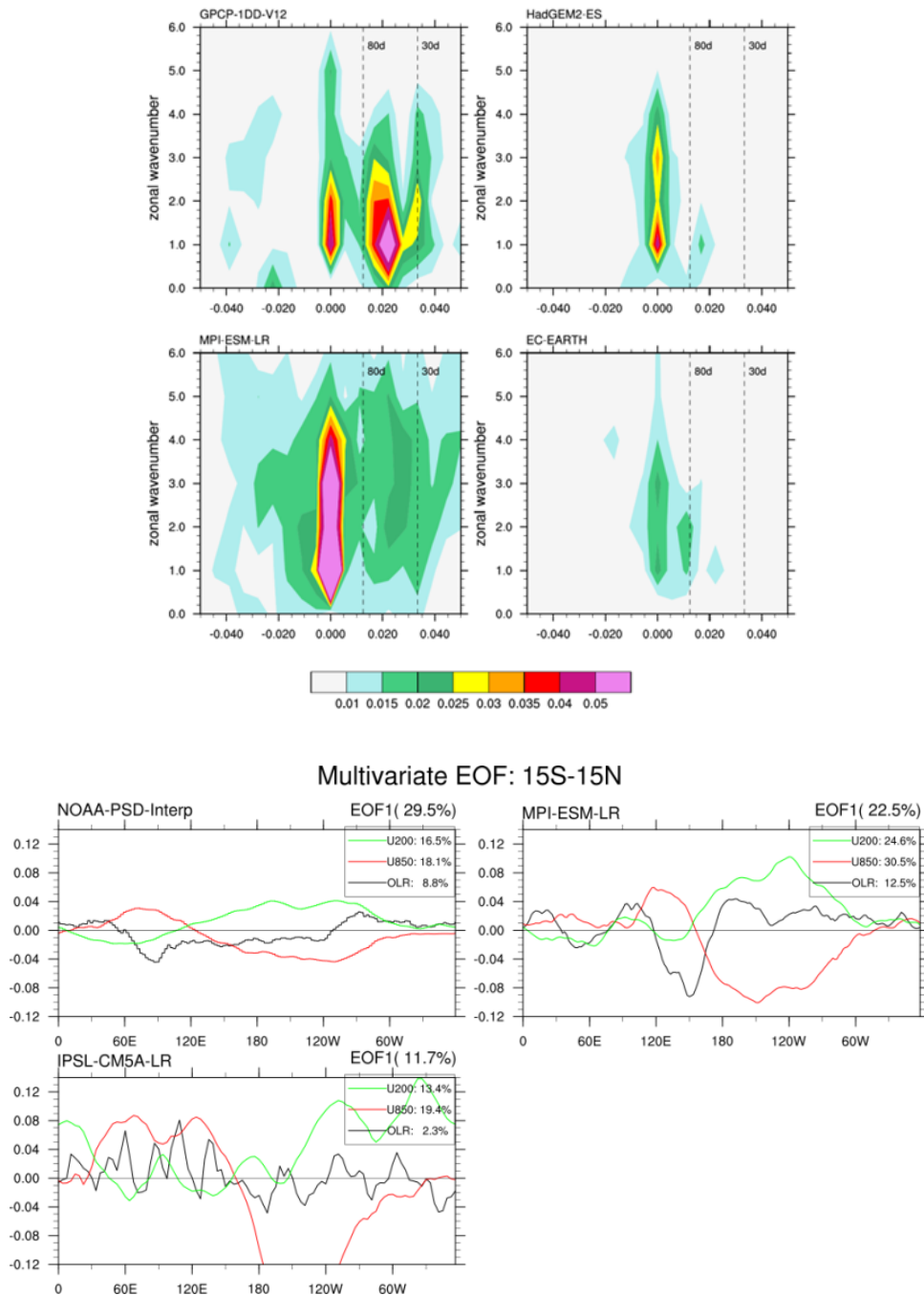
- ERA-Interim (ua, monthly means)
Reformat script: reformat_scripts/obs/ reformat_obs_ERA-Interim.ncl
- GPCP-1DD (pr, daily means obs4mips)
- GPCP-SG (pr, monthly means obs4mips)
- NCEP (ua, va, daily means)
Reformat script: reformat_scripts/obs/reformat_obs_NCEP-daily.ncl
- NOAA-PSD-Interp (rlut, daily means)
Reformat script: reformat_scripts/obs/reformat_obs_NOAA-PSD-Interp.ncl

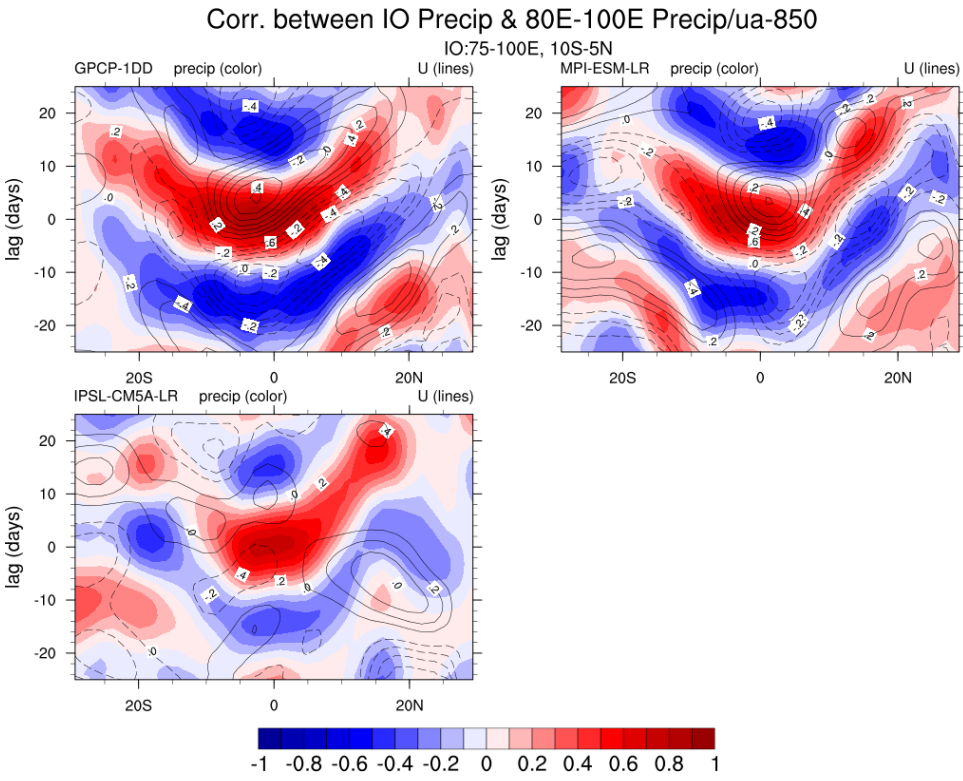
30.6 References

- Kanamitsu, M., Kousky, V., van den Dool, H., Jenne, R., and Fiorino, M.: The NCEP-NCAR 50-Year Reanalysis: Monthly Means CD-ROM and Documentation, B. Am. Meteorol. Soc., 82, 247-267, 2001.
- Kim, D., Sperber, K., Stern, W., Waliser, D., Kang, I. S., Maloney, E., Wang, W., Weickmann, K., Benedict, J., Khairoutdinov, M., Lee, M. I., Neale, R., Suarez, M., Thayer-Calder, K., and Zhang, G.: Application of MJO Simulation Diagnostics to Climate Models, J. Climate, 22, 6413-6436, 2009.
- Liebmann, B. and Smith, C. A.: Description of a complete (interpolated) outgoing longwave radiation data set, B. Am. Meteorol. Soc., 77, 1275-1277, 1996.

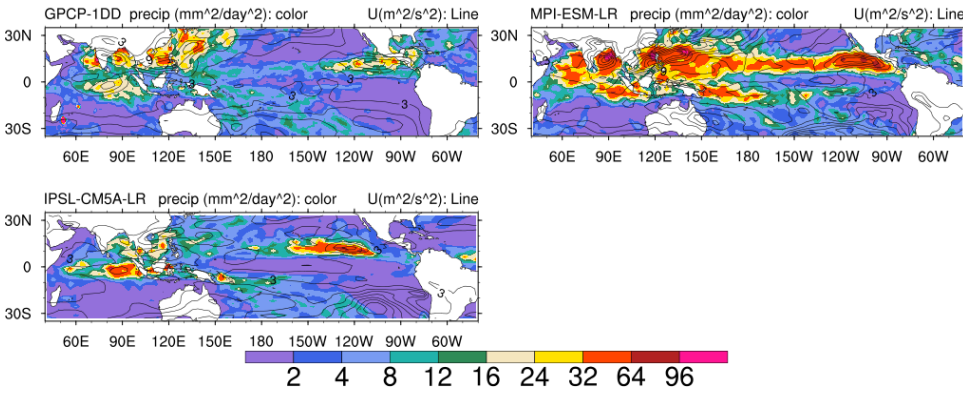
- Waliser, D., Sperber, K., Hendon, H., Kim, D., Wheeler, M., Weickmann, K., Zhang, C., Donner, L., Gottschalck, J., Higgins, W., Kang, I. S., Legler, D., Moncrieff, M., Vitart, F., Wang, B., Wang, W., Woolnough, S., Maloney, E., Schubert, S., Stern, W., and Oscillation, C. M.-J.: MJO Simulation Diagnostics, J. Climate, 22, 3006-3030, 2009.

30.7 Example plots





summer



NCAR's Climate Variability Diagnostics Package (CVDP)

31.1 Overview

The NCAR CVDP version currently implemented into the ESMValTool is v4.1.

The Climate Variability Diagnostics Package (CVDP) developed by NCAR's Climate Analysis Section (Phillips et al., 2014) has been implemented into the ESMValTool in order to be able to run it within this framework and alongside the ESGF on CMIP output. CVDP can be used to evaluate the major modes of climate variability including ENSO, PDO, AMO, Northern and Southern Annular Modes (NAM and SAM), North Atlantic Oscillation (NAO), Pacific North and South American teleconnection patterns (PNA and PSA). In addition it calculates global trend maps and index time series for the above modes and the North Pacific Index, the Tropical North Atlantic SST, Tropical South Atlantic SST, Tropical Indian Ocean SST, Niño1+2, Niño3, and Niño4 times series, and the Indian Ocean Dipole (IOD).

CVDP is developed as a standalone tool outside the ESMValTool. Once a new version of CVDP is released, the ESMValTool will be updated accordingly. Therefore, the structure of CVDP was kept as is and a wrapper has been written to be able to run CVDP directly within the ESMValTool.

31.2 Available namelists and diagnostics

Namelists are stored in nml/

- namelist_CVDP.xml

Diagnostics are stored in diag_scripts/

Wrapper scripts to run CVDP within the framework of the ESMValTool

- cvdp_obs.ncl: run for each variable separately with observational data available; renames the ESMValTool output (observations) following the filename conventions of the CVDP and creates the CVDP namelist "namelist_obs".
- cvdp_ocean.ncl: renames the ESMValTool output (ocean variables) following the filename conventions of the CVDP.
- cvdp_atmos.ncl: renames the ESMValTool output (atmosphere variables) following the filename conventions of the CVDP and creates the CVDP namelist "namelist" containing the models. The script then runs the CVDP via a call to the wrapper script cvdp_driver.ncl.

31.3 User settings

User setting files (cfg files) are stored in nml/cfg_CVDP/

1. cvdp_obs.ncl

Required diag_script_info attributes

- obs_ref: list of reference data sets (observations) (array)

2. cvdp_driver.ncl (called by cvdp_atmos.ncl)

The wrapper script cvdp_driver.ncl sets the user options for the CVDP.

31.4 Variables

- ts (atmos, monthly mean, longitude latitude time)
- psl (atmos, monthly mean, longitude latitude time)
- tas (atmos, monthly mean, longitude latitude time)
- pr (atmos, monthly mean, longitude latitude time)
- msftmyz (ocean, monthly mean, latitude level basin time)

31.5 Observations and Reformat Scripts

Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

- Sea Surface Temperature (SST): HadISST (1870 - 2005)

Reformat script: reformat_scripts/obs/reformat_obs_HadISST.ncl

- Sea Level Pressure (PSL): ERA-Interim/IFS-Cy31r2 (1979 - 2013)

Reformat script: reformat_scripts/obs/reformat_obs_ERA-Interim.ncl

- 2m Air Temperature (TAS): NCEP (1979 - 2005)

Reformat script: reformat_scripts/obs/reformat_obs_NCEP.ncl

- Precipitation (PR): GPCP-SG (1979 - 2005) (obs4mips)

31.6 References

- Danabasoglu, G., S. G. Yeager, Y. -O. Kwon, J. J. Tribbia, A. S. Phillips, and J. W. Hurrell, 2012: Variability of the Atlantic Meridional Overturning Circulation in CCSM4. *J. Climate*, 25, 5153-5172, doi: 10.1175/JCLI-D-11-00463.1 (AMOC).
- Deser, C., M. A. Alexander, S. -P. Xie, and A. S. Phillips, 2010: Sea surface temperature variability: patterns and mechanisms. *Ann. Rev. Mar. Sci.*, 2010.2, 115-143, doi: 10.1146/annurev-marine-120408-151453 (PDO).
- Deser, C., A. S. Phillips, R. A. Tomas, Y. Okumura, M. A. Alexander, A. Capotondi, J. D. Scott, Y. -O. Kwon, and M. Ohba, 2012: ENSO and Pacific Decadal Variability in Community Climate System Model Version 4. *J. Climate*, 25, 2622-2651, doi: 10.1175/JCLI-D-11-00301.1 (ENSO Spatial Composites).
- Hurrell, J. W., and C. Deser, 2009: North Atlantic climate variability: The role of the North Atlantic Oscillation. *J. Mar. Syst.*, 78, 28-41, doi 10.1016/j.jmarsys.2008.11.026 (NAO).

- Mantua, N. J., S. R. Hare, Y. Zhang, J. M. Wallace, and R. Francis, 1997: A Pacific interdecadal climate oscillation with impacts on salmon production. *Bull. Amer. Met. Soc.*, 1069-1079 (PDO).
- Phillips, A. S., Deser, C., and Fasullo, J.: Evaluating Modes of Variability in Climate Models, *Eos Trans. AGU*, 95(49), 453-455, 2014 (Overview Paper of Climate Variability Diagnostics Package).
- Thompson, D. W. J. and J.M. Wallace, 2000: Annular modes in the extratropical circulation. Part I: Month-to-month variability. *J. Climate*, 13, 1000-1016 (NAM).
- Trenberth, K. E., and D. J. Shea, 2006: Atlantic hurricanes and natural variability in 2005, *Geophys. Res. Lett.*, 33, L12704, doi: 10.1029/2006GL026894 (AMO).

31.7 Example plots

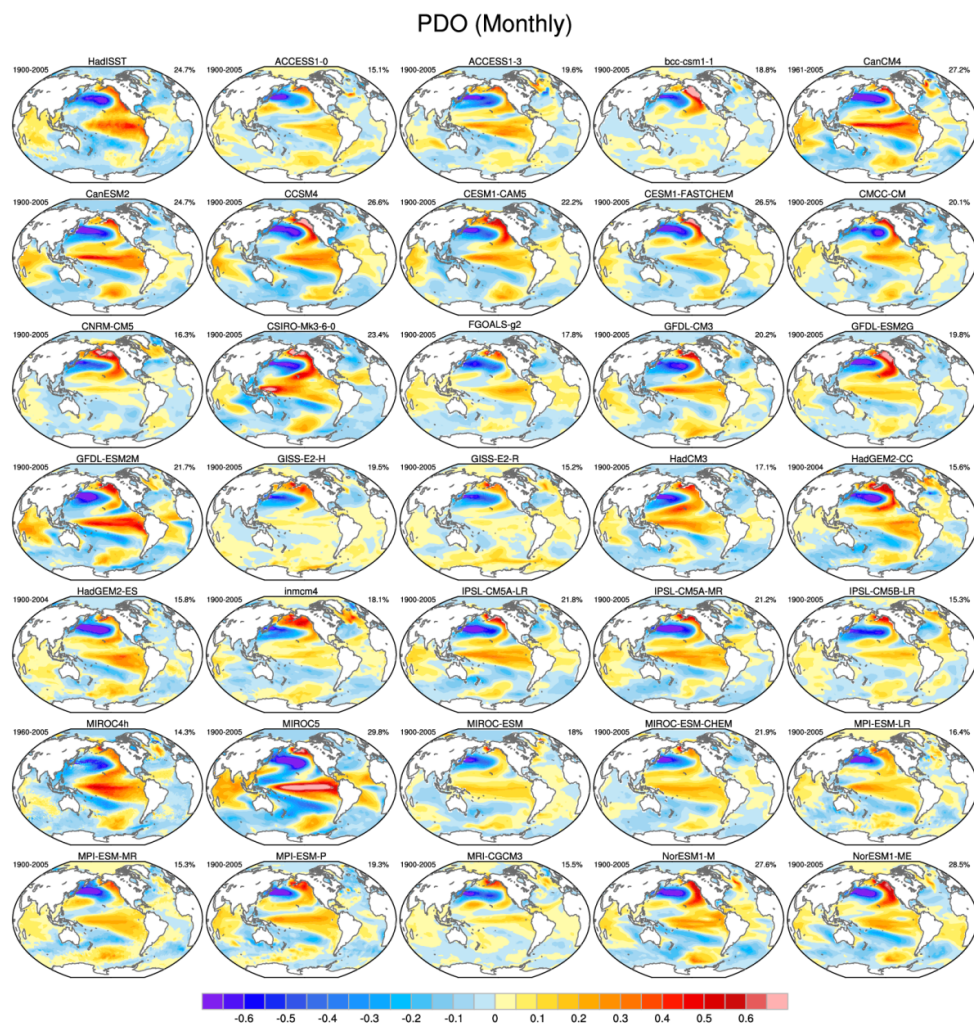


Figure 1: The leading empirical orthogonal function (EOF) of monthly sea surface temperature (SST) anomalies over the North Pacific (after removing the global mean SST anomaly) based on the HadISST observational data set (top left) during 1900-2005, and (remaining panels) several model simulations of the Coupled Model Intercomparison Project Phase 5 (CMIP5), for the models MIROC4h and CanCM4 during 1960-2005.

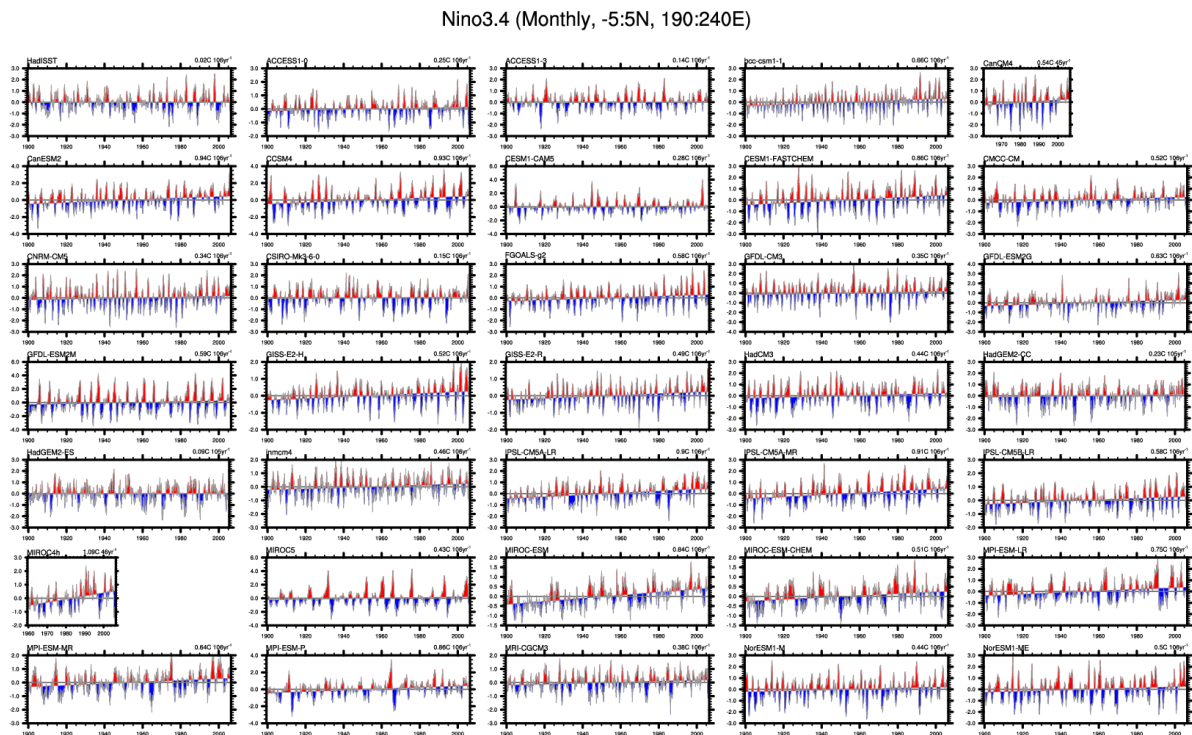


Figure 2: The weighted area average of monthly SST anomalies in the region 5°S - 5°N , 190° - 240°E also known as the “Niño-3.4 Index” (Trenberth et al., 2002) based on the HadISST observational data set (top left) for the period 1900-2005, and (remaining panels) several model simulations of the Coupled Model Intercomparison Project Phase 5 (CMIP5), for the models MIROC4h and CanCM4 during 1960-2005. The red/blue shading on the Niño3.4 time series denotes positive/negative deviations from the best-fit linear trend line.

La Niña Composite (3°S:3°N)

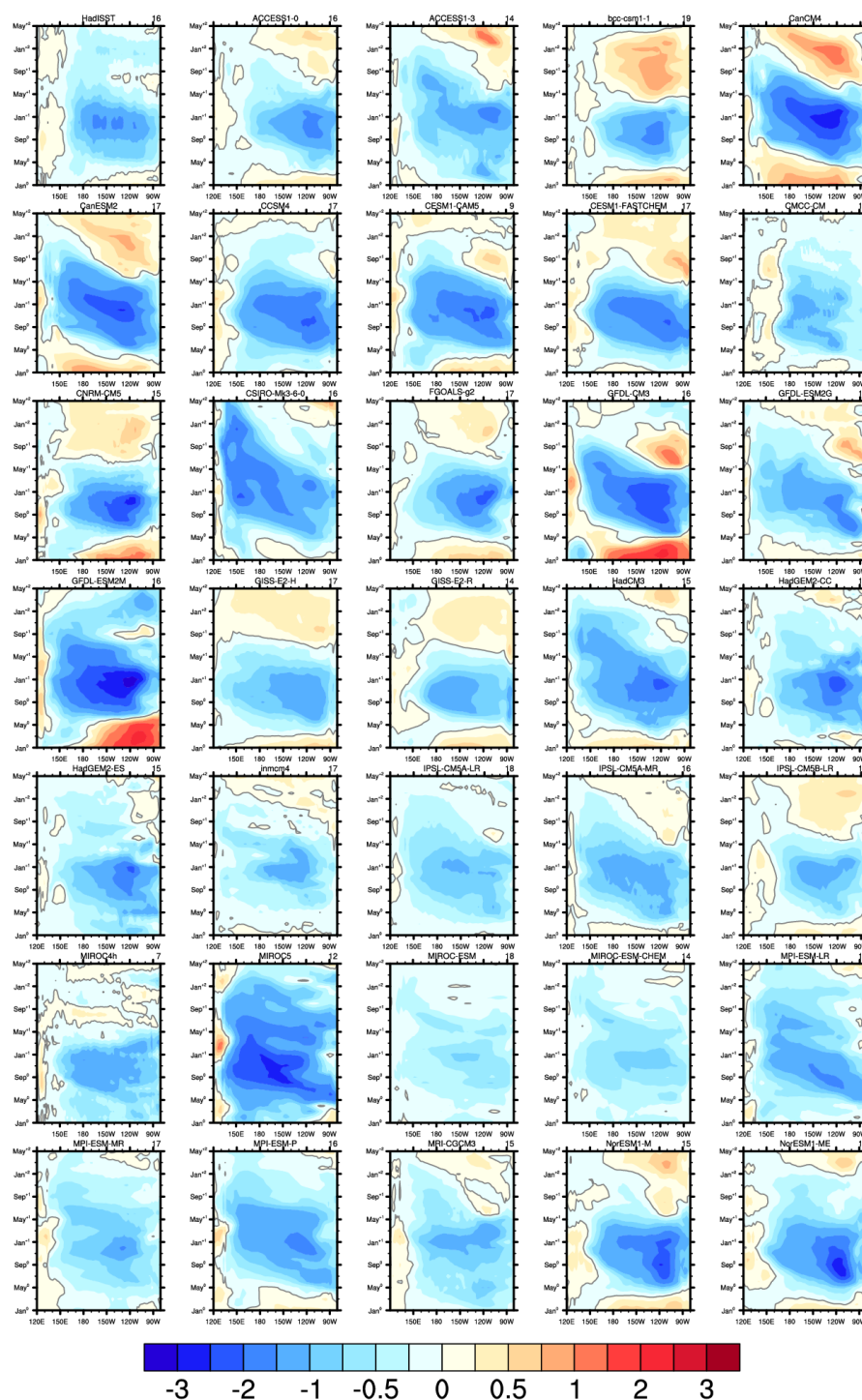


Figure 3: Longitude-latitude sections of composite SST anomalies along the equator (3°N-3°S) for La Niña based on the HadISST observational data set (top left) for the period 1900-2005, and (remaining panels) several model simulations of the Coupled Model Intercomparison Project Phase 5 (CMIP5), for the models MIROC4h and CanCM4 during 1960-2005. The number at the top right of each panel indicates the number of events used for the composite.

AMO (Monthly)

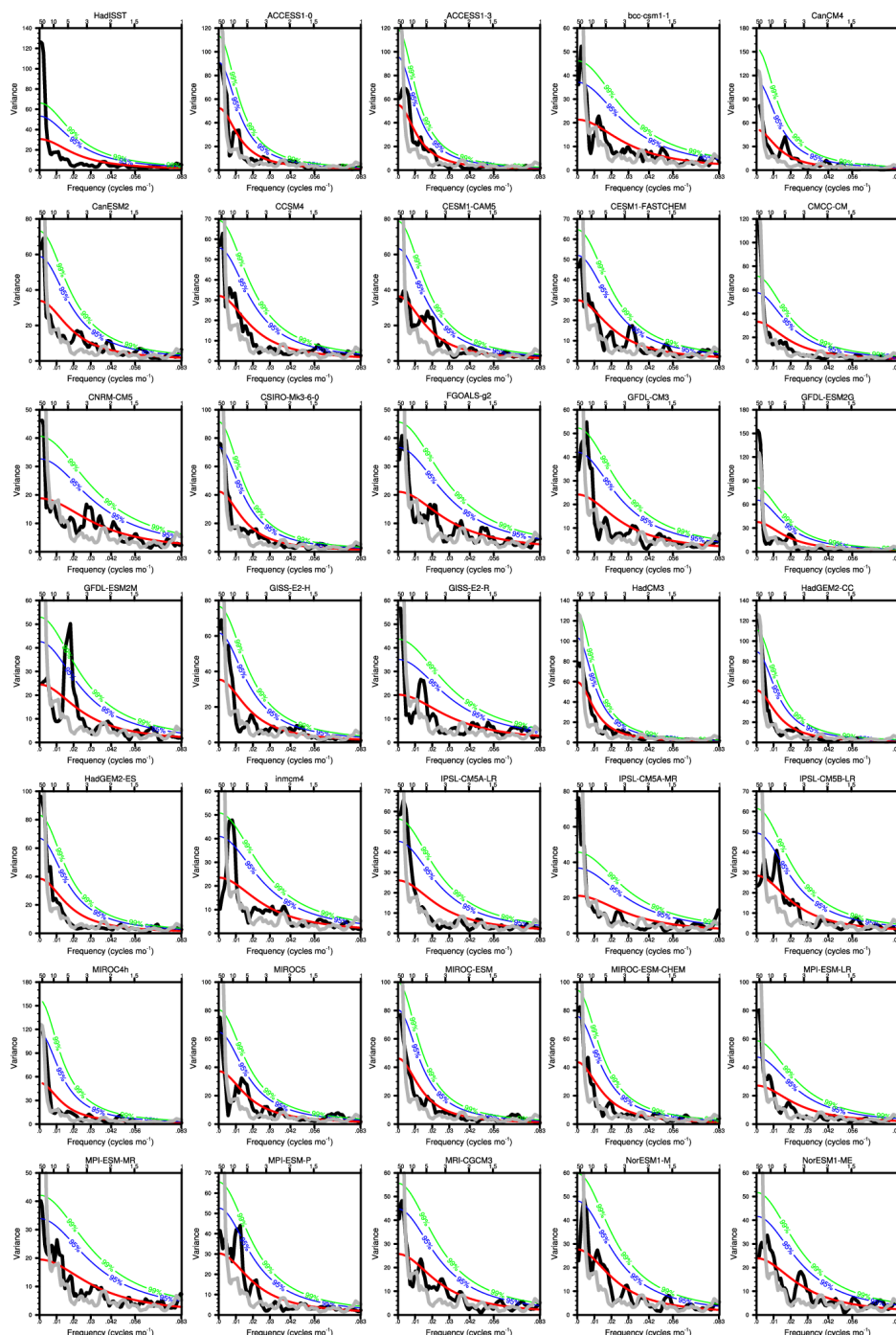


Figure 4: The power spectrum as a function of frequency in cycle per month of the Niño-3.4 SST index for the HadISST observational data set (top left) for the period 1900-2005, and (remaining panels) several model simulations of the Coupled Model Intercomparison Project Phase 5 (CMIP5), for the models MIROC4h and CanCM4 during 1960-2005. The black line denotes the spectrum. The red line indicates the best-fit first-order Markov red noise spectrum, the blue line its 95% and the green line its 99% confidence bounds. The observational spectrum is overlaid in gray on each model spectrum if available.

Ozone and associated climate impacts

32.1 Overview

This namelist, implemented into the ESMValTool to evaluate atmospheric chemistry and the climate impact of stratospheric ozone changes, reproduces selected plots from Eyring et al. (2013), i.e. their figs. 1, 2, 4, 6, 7, 10, and 11. These include calculation of the zonally averaged seasonal cycle of total ozone columns (Figure 1), time series of the total ozone averaged over given regions (Figure 2), climatological mean tropospheric ozone columns (Figure 3), stratospheric ozone time series (Figure 4), differences in vertical ozone profiles between the 2090s and 2000s (Figure 5), trends in annual mean ozone, temperature, and jet position (Figure 6), and trend relationships between ozone and temperature and jet position (Figure 7).

32.2 Available namelists and diagnostics

Namelists are stored in nml/

- namelist_eyring13jgr.xml

Diagnostics are stored in diag_scripts/

- eyring13jgr_fig01.ncl: calculates seasonal cycles of zonally averaged total ozone columns.
- eyring13jgr_fig02.ncl: time series of area-weighted total ozone from 1960-2005 for the annual mean averaged over the global domain (90°S-90°N), Tropics (25°S-25°N), northern mid-latitudes (35°N-60°N), southern mid-latitudes (35°S-60°S), and the March and October mean averaged over the Arctic (60°N-90°N) and the Antarctic (60°S-90°S).
- eyring13jgr_fig04.ncl: climatological annual mean tropospheric ozone columns (geographical distribution).
- eyring13jgr_fig06.ncl: 1980 baseline-adjusted stratospheric column ozone time series from 1960-2100.
- eyring13jgr_fig07.ncl: differences in vertically resolved ozone between the 2090s and 2000s for the annual mean averaged over the global domain (90°S-90°N), Tropics (25°S-25°N), northern mid-latitudes (35°N-60°N), southern mid-latitudes (35°S-60°S), and the March and October mean averaged over the Arctic (60°N-90°N) and the Antarctic (60°S-90°S).
- eyring13jgr_fig10.ncl: trends in annual mean near-global (82.5°S-82.5°N) ozone at 50 hPa and temperature at 100 hPa, September-October-November-December (SOND) ozone at 50 hPa over Antarctica (60°S-90°S), October-November-December-January (ONDJ) temperature at 100 hPa over Antarctica (60°S-90°S),

DJF SH jet position at 850 hPa, and DJF upper tropospheric tropical (30°S-30°N) temperatures at 250 hPa. The trends are calculated over 1979-2005 for the past and over 2006-2050 for the future.

- `eyring13jgr_fig11.ncl`: trend relationship between annual mean near-global (82.5°S-82.5°N) ozone at 50 hPa and temperature at 100 hPa, SOND ozone at 50 hPa and ONDJ temperature at 100 hPa over Antarctica (60°S-90°S), SOND ozone at 50 hPa and DJF jet position at 850 hPa; and DJF 250 hPa tropical (30°S-30°N) temperatures and DJF jet position at 850 hPa. The trends are calculated over 1979-2005 for the past and over 2006-2050 for the future.

32.3 User settings

User setting files (cfg files) are stored in `nml/cfg_eyring13jgr`

1. `eyring13jgr_fig01.ncl`

diag_script_info attributes

- `rgb_file`: path + filename of color table (e.g., `"diag_scripts/lib/ncl/rgb/eyring_toz.rgb"`)
- `styleset`: style set ("DEFAULT", "CMIP5")
- `font`: overrides default font (e.g., 21 ; see www.ncl.ucar.edu/Document/Graphics/Resources/tx.shtml#txFont)
- `range_option`: 0 = as in nml, 1 = overlapping time period
- `lbLabelBarOn`: plot a label bar (True, False)
- `e13fig01_` = "True"
- `e13fig01_list_chem_mod`: list of models in the group "chem" (array of strings, default = `(/"All"/)`)
- `e13fig01_list_chem_mod_string`: plotting label for group "chem", e.g., "CMIP5"
- `e13fig01_list_nochem_mod`: list of models in the group "nochem" (array of strings, default = `(/"")`)
- `e13fig01_list_nochem_mod_string`: plotting label for group "nochem", e.g., "NoChem"
- `e13fig01_diff_ref`: name of reference model for difference plots, e.g., "NIWA"

2. `eyring13jgr_fig02.ncl`

diag_script_info attributes

- `e13fig02_latrange`: min. and max. latitude of the regions (n-element array of 2-element pairs, e.g., `(/(-90,90/), (/90,-60/)/)`; one pair of latitudes is required for each season (see below)
- `styleset`: style set ("DEFAULT", "CMIP5")
- `e13fig02_season`: seasons (n-element array of strings, "ANN", "JAN", "FEB", "MAR", "DJF", "SON", etc.)
- `e13fig02_XMin`: min. x-values (start years) for plotting (n-element array, e.g., `(/1960., 1960./)`; array is required to have the same number of elements as "seasons" (see above)
- `e13fig02_XMax`: max. x-values (end years) for plotting (n-element array, e.g., `(/2005., 2005./)`; array is required to have the same number of elements as "seasons" (see above)
- `e13fig02_YMin`: min. y-values for plotting (n-element array, e.g., `(/260., 150./)`; array is required to have the same number of elements as "seasons" (see above)
- `e13fig02_YMax`: max. y-values for plotting (n-element array, e.g., `(/340., 500./)`; array is required to have the same number of elements as "seasons" (see above)
- `e13fig02_legend`: plot legend (string, e.g., "True")
- `e13fig02_legend_MMM`: include multi model mean in legend (string, e.g., "False")
- `list_chem_mod`: list of models in the group "chem" (array of strings, default = `(/"All"/)`)
- `list_nochem_mod`: list of models in the group "nochem" (array of strings, default = `(/"None"/)`)

3. eyring13jgr_fig04.ncl

diag_script_info attributes

- **styleset:** style set ("DEFAULT", "CMIP5")
- **font:** overrides default font (e.g., 21 ; see www.ncl.ucar.edu/Document/Graphics/Resources/tx.shtml#txFont)
- **range_option:** 0 = as in nml, 1 = overlapping time period
- **lbLabelBarOn:** plot a label bar (True, False)
- **e13fig04_ = "True"**
- **e13fig04_list_chem_mod:** list of models in the group "chem" (array of strings, default = (/ "All" /))
- **e13fig04_list_chem_mod_string:** plotting label for group "chem", e.g., "CMIP5"
- **e13fig04_list_nochem_mod:** list of models in the group "nochem" (array of strings, default = (/ "" /))
- **e13fig04_list_nochem_mod_string:** plotting label for group "nochem", e.g., "NoChem"
- **e13fig04_diff_ref:** name of reference model for difference plots, e.g., "AURA-MLS-OMI"
- **mpProjection:** map projection, optional (e.g., "CylindricalEquidistant") (see <http://www.ncl.ucar.edu/Document/Graphics/Resources/mp.shtml#mpProjection> for available projections)

4. eyring13jgr_fig06.ncl

diag_script_info attributes

- **e13fig06_latrange:** min. and max. latitude of the regions (n-element array of 2-element pairs, e.g., (/(-90,90/), (/ -90,-60/)/)); one pair of latitudes is required for each season (see below)
- **styleset:** style set ("DEFAULT", "CMIP5")
- **e13fig06_season:** seasons (n-element array of strings, "ANN", "JAN", "FEB", "MAR", "DJF", "SON", etc.)
- **e13fig06_baseline_adj:** do baseline adjustment (string: "True", "False")
- **e13fig06_baseline:** year for baseline adjustment (e.g., 1980)
- **e13fig06_mod_plot:** "MMT" = plot of the MultiModel mean of each scenario and selection "list_chem_mod" and "list_nochem_mod"; "IMT" = plot of each single model trend; "RAW" = plot of each model as raw data
- **e13fig06_mod_plot_CI:** plot confidence interval (string: "True", "False"); for e13fig06_mod_plot = "MMT" only!
- **e13fig06_mod_plot_PI:** plot prediction interval (string: "True", "False"); for e13fig06_mod_plot = "MMT" only!
- **e13fig06_XMin:** min. x-values (start years) for plotting (n-element array, e.g., (/1960., 1960./)); array is required to have the same number of elements as "seasons" (see above)
- **e13fig06_XMax:** max. x-values (end years) for plotting (n-element array, e.g., (/2010., 2010./)); array is required to have the same number of elements as "seasons" (see above)
- **e13fig06_YMin:** min. y-values for plotting (n-element array, e.g., (/260., 150./)); array is required to have the same number of elements as "seasons" (see above)
- **e13fig06_YMax:** max. y-values for plotting (n-element array, e.g., (/330., 500./)); array is required to have the same number of elements as "seasons" (see above)
- **list_chem_mod:** list of models in the group "chem" (array of strings, default = (/ "All" /))
- **list_nochem_mod:** list of models in the group "nochem" (array of strings, default = (/ "None" /))
- **e13fig06_labels_exp_esp:** specify experiment name (string: "True", "False"); only if e13fig06_mod_plot = "IMT" or "RAW"!

5. eyring13jgr_fig07.ncl

diag_script_info attributes

- e13fig06_latrange: min. and max. latitude of the regions (n-element array of 2-element pairs, e.g., $((-90, 90), (-90, -60))$); one pair of latitudes is required for each season (see below)
- styleset: style set ("DEFAULT", "CMIP5")
- e13fig07_season: seasons (n-element array of strings, "ANN", "JAN", "FEB", "MAR", "DJF", "SON", etc.)
- e13fig07_period1: start and end year of "period1" (= 2000s), e.g., $(2000., 2009/)$
- e13fig07_period2: start and end year of "period2" (= 2090s), e.g., $(2090., 2099/)$
- e13fig07_XMin: min. x-values for plotting (n-element array, e.g., $(-2., -2./)$); array is required to have the same number of elements as "seasons" (see above)
- e13fig07_XMax: max. x-values for plotting (n-element array, e.g., $(2., 12./)$); array is required to have the same number of elements as "seasons" (see above)
- list_chem_mod: list of models in the group "chem" (array of strings, default = $(/All/)$)
- list_nochem_mod: list of models in the group "nochem" (array of strings, default = $(/None/)$)

6. eyring13jgr_fig10.ncl

diag_script_info attributes

- e13fig10_latrange: min. and max. latitude of the regions (n-element array of 2-element pairs, e.g., $((-30, 30))$); one pair of latitudes is required for each season (see below)
- styleset: style set ("DEFAULT", "CMIP5")
- e13fig10_season: seasons (n-element array of strings, e.g., "ANN", "JAN", "FEB", "MAR", "DJF", "SON", etc.)
- e13fig10_lev: vertical level (in hPa)
- plot_number: string used for plot labeling / sub-figure (e.g., "(a)")
- list_chem_mod: list of models in the group "chem" (array of strings, default = $(/All/)$)
- list_nochem_mod: list of models in the group "nochem" (array of strings, default = $(/None/)$)

7. eyring13jgr_fig11.ncl

diag_script_info attributes

- styleset: style set ("DEFAULT", "CMIP5")
- e13fig11_V0_units: unit label for "variable 0" (x-axis) (string)
- e13fig11_V1_units: unit label for "variable 1" (y-axis) (string)
- e13fig11_V0_latrange: min. and max. latitude of the region for "variable 0"
- e13fig11_V1_latrange: min. and max. latitude of the region for "variable 1"
- e13fig11_V0_season: season for "variable 0" (e.g., "yearly")
- e13fig11_V1_season: season for "variable 1" (e.g., "yearly")
- e13fig11_V0_lev: vertical level (in hPa) for "variable 0"
- e13fig11_V1_lev: vertical level (in hPa) for "variable 1"
- plot_number: string used for plot labeling / sub-figure (e.g., "(a)")
- e13fig11_XMin: min. x-value for plotting
- e13fig11_XMax: max. x-value for plotting
- e13fig11_YMin: min. y-value for plotting

- e13fig11_YMax: max. y-value for plotting
- list_chem_mod: list of models in the group “chem” (array of strings, default = empty)
- list_nochem_mod: list of models in the group “nochem” (array of strings, default = empty)

32.4 Variables

- tro3 (atmos, monthly mean, longitude latitude lev time)
- ta (atmos, monthly mean, longitude latitude lev time)
- ua (atmos, monthly mean, longitude latitude lev time)

32.5 Observations and reformat scripts

- Total column ozone (toz): NIWA (Bodeker et al., 2005)

Reformat script: reformat_scripts/obs/reformat_obs_NIWA.ncl

- Tropospheric column ozone (tropoz): MLS/OMI (Ziemke et al., 2006)

Reformat script: reformat_scripts/obs/reformat_obs_AURA-MLS-OMI.ncl

32.6 References

- Eyring, V., J. M. Arblaster, I. Cionni, J. Sedlacek, J. Perlwitz, P. J. Young, S. Bekki, D. Bergmann, P. Cameron-Smith, W. J. Collins, G. Faluvegi, K.-D. Gottschaldt, L. W. Horowitz, D. E. Kinnison, J.-F. Lamarque, D. R. Marsh, D. Saint-Martin, D. T. Shindell, K. Sudo, S. Szopa, and S. Watanabe, Long-term ozone changes and associated climate impacts in CMIP5 simulations, *J. Geophys. Res. Atmos.*, 118, doi: 10.1002/jgrd.50316, 2013.

32.7 Example plots

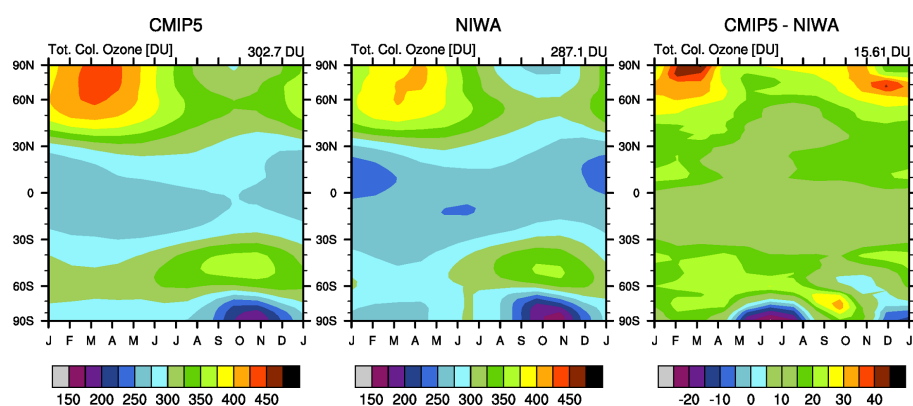


Figure 1: Produced with “eyring13jgr_fig01.ncl”.

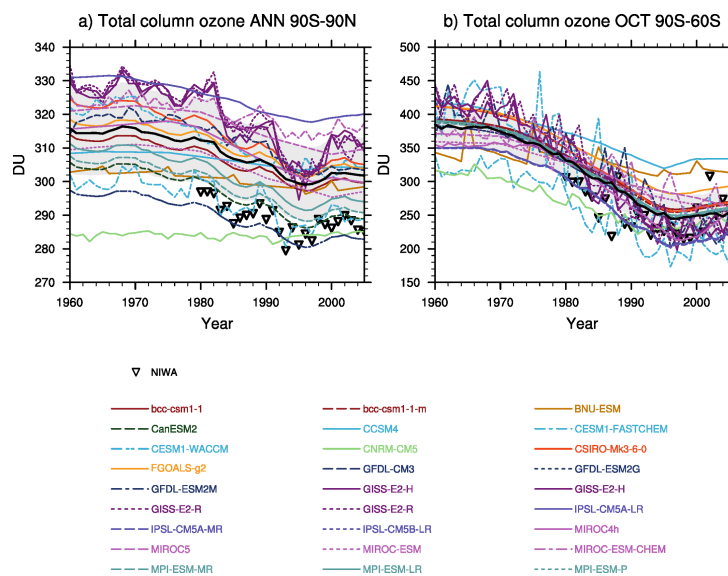


Figure 2: Produced with “eyring13jgr_fig02.ncl”.

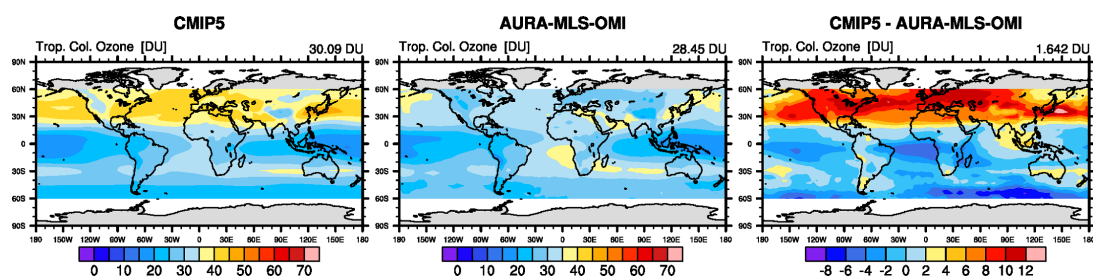


Figure 3: Produced with “eyring13jgr_fig04.ncl”.

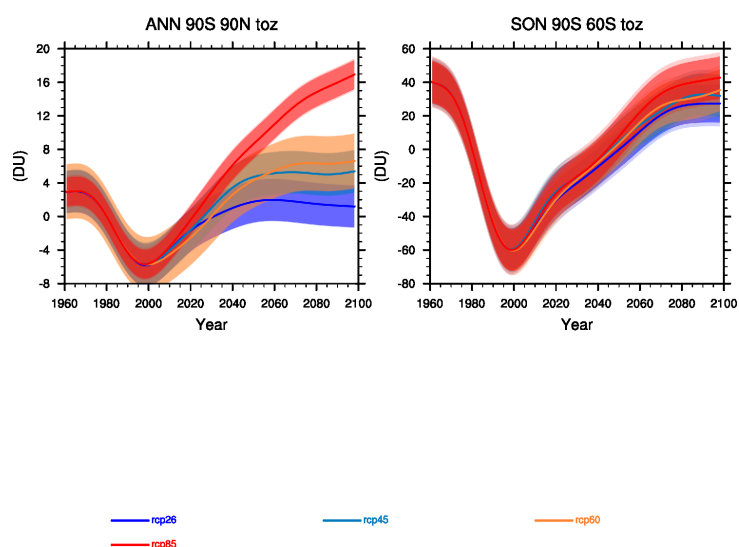


Figure 4: Produced with “eyring13jgr_fig06.ncl”.

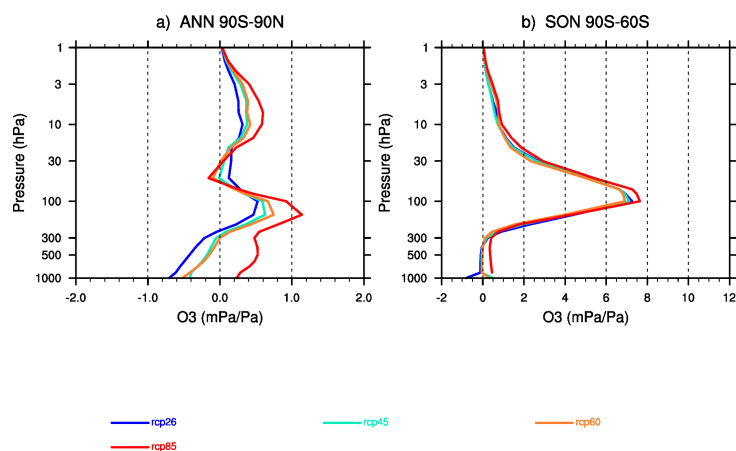


Figure 5: Produced with “eyring13jgr_fig07.ncl”.

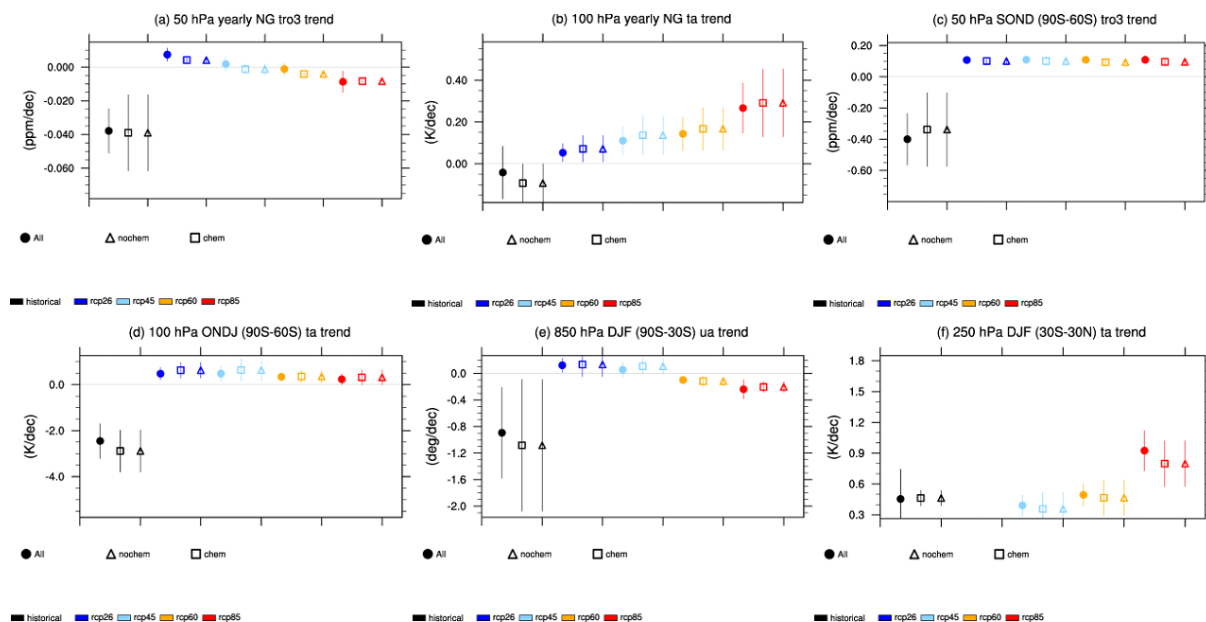


Figure 6: Produced with “eyring13jgr_fig10.ncl”.

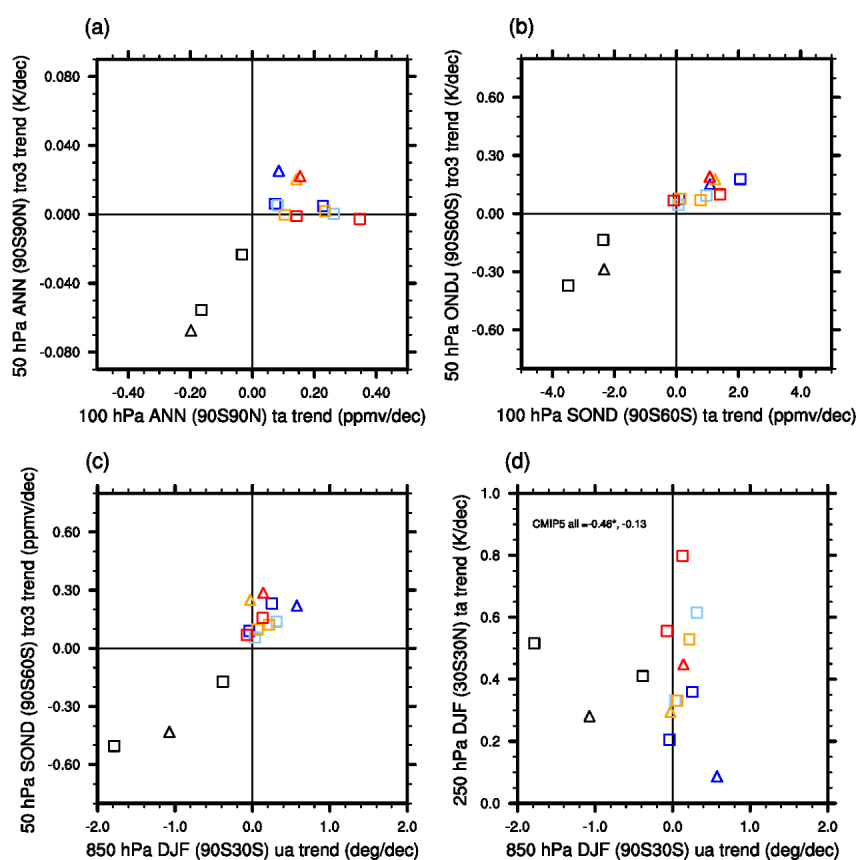


Figure 7: Produced with “eyring13_jgr_fig11.ncl”

Ozone and some precursors

33.1 Overview

This namelist provides diagnostics to evaluate the simulated atmospheric composition in ESMs with a focus on ozone and precursor gases such as NO_x . This includes comparisons of simulated tropospheric ozone columns, seasonal cycle and vertical profiles of trace gases with observations. Model data are sampled for the same regions and the same time of year as the observations, monthly data are weighted according to the number of days in each month contributing to the respective campaign climatology.

33.2 Available namelists and diagnostics

Namelists are stored in `nml/`

- `namelist_righi15gmd_Emmons.xml`
- `namelist_righi15gmd_tropo3.xml`

Diagnostics are stored in `diag_scripts/`

- `anycyc_lat.ncl`: annual cycle contour plots of zonal means.
- `Emmons.ncl`: profiles of atmospheric trace gas mixing ratios, compared to the in-situ data climatologies of Emmons et al. (2000). Simulation data are sampled for the same regions and the same time of year as the observations. Monthly data are weighted according to the number of days in each month contributing to the respective campaign climatology. Note: simulation data are sampled according to the period specified in the namelist. Observational data may be from different years.
- `lat_lon.ncl`: plots the geographical distribution of a variable (latitude-longitude plot).
- `perfmetrics_grading.ncl`: calculates grades according to a given metric with different options for normalization. It requires fields precalculated by `perfmetrics_main.ncl` (see *Performance metrics for essential climate parameters*).
- `perfmetrics_grading_collect.ncl`: collects results from metrics previously calculated by `perfmetrics_grading.ncl` and passes them to the plotting functions (see *Performance metrics for essential climate parameters*).
- `perfmetrics_main.ncl`: calculates and (optionally) plots annual/seasonal cycles, zonal means, lat-lon fields and time-lat-lon fields from input monthly 2-d or 3-d (“T2M”, “T3Ms”) data. The calculated fields can be

also plotted as difference w.r.t. a given reference model. They are also used as input to calculate grading metrics (see `perfmetrics_grading.ncl`) (see *Performance metrics for essential climate parameters*).

- `perfmetrics_taylor.ncl`: calculates grades according to a given metric with different options for normalization. It requires fields precalculated by `perfmetrics_main.ncl` (see *Performance metrics for essential climate parameters*).
- `perfmetrics_taylor_collect.ncl`: collects results from metrics previously calculated by `perfmetrics_taylor.ncl` and passes them to the plotting functions (see *Performance metrics for essential climate parameters*).

33.3 User settings

User setting files (cfg files) are stored in `nml/cfg_righi15gmd/Emmons/` and `nml/cfg_righi15gmd_tropo3/`

1. `anyc_lat.ncl`

Requires `diag_script_info` attributes

- `stylesheet`: style (“DEFAULT”, “righi15gmd”; see `diag_scripts/lib/ncl/style.ncl` for available styles)
- `font`: font type (see www.ncl.ucar.edu/Document/Graphics/Resources/tx.shtml#txFont)

Optional `diag_script_attributes`

- `obsfiles`
- `range_option`: 0 = as in `nml`, 1 = overlapping time period
- `case`
- `y1`
- `y2`
- `obsname`
- `lbLabelBarOn`: plot label bar (True/False)

2. `Emmons.ncl`

- **obsfiles**: list of observational files including full pathnames; all files matching the following will be scanned, all suitable data will be used (mandatory), wildcards are allowed.

OBS parameters

The target variable (e.g., “`vmro3`”) + the following extensions give the variable names in the obs file for specific quantities such as mean, standard deviation, minimum, etc. Deactivate lines corresponding to parameters that are not in the obs file; either `obsvar_mean` or `obsvar_median` is mandatory. “`_N`” is only used for grading: will use equal weights if disabled. Disabling any other quantity will just cause the corresponding whisker not to be drawn (this may be used to switch off whiskers).

- `obsvar_N`: extension to assemble variable name for number of observations, e.g., “`_N`”
- `obsvar_mean`: extension to assemble variable name for mean, e.g., “`_mean`”
- `obsvar_stddev`: extension to assemble variable name for standard deviation, e.g., “`_stddev`”
- `obsvar_min`: extension to assemble variable name for minima, e.g., “`_min`”
- `obsvar_max` : extension to assemble variable name for maxima, e.g., “`_max`”
- `obsvar_5`: extension to assemble variable name for 5% percentiles, e.g., “`_5`”
- `obsvar_25`: extension to assemble variable name for 25% percentiles, e.g., “`_25`”
- `obsvar_median`: extension to assemble variable name for median, e.g., “`_median`”
- `obsvar_75`: extension to assemble variable name for 75% percentiles, e.g., “`_75`”
- `obsvar_95`: extension to assemble variable name for 95% percentiles, e.g., “`_95`”

Campaign parameters, expected to be global attributes of each obs file (all mandatory)

- `obsatt_campaign`: name of the attribute containing the campaign name (plot annotation), e.g., “campaign”
- `obsatt_location`: name of the attribute containing the region (plot annotation), e.g., “location”
- `obsatt_period`: name of the attribute containing the time period covered, e.g., “period”
- `obsatt_latmin`: name of the attribute containing the min. latitude of the region, e.g., “latmin”
- `obsatt_latmax`: name of the attribute containing the max. latitude of the region, e.g., “latmax”
- `obsatt_lonmin`: name of the attribute containing the min. longitude of the region, e.g., “lonmin”
- `obsatt_lonmax`: name of the attribute containing max. longitude of the region, e.g., “lonmax”

Optional FILTER parameters for selection of obs data

- `roi`: region of interest (4-element array given as (/latmin, latmax, lonmin, lonmax/)) (default = (/90., 90., 0., 360./))
- `roi_match`: “contained”, “center”, “overlap”, “encloses”, “outside” (default: “center”); e.g., “center” = center of observational region (given by global attributes of observational file) must be inside the “region of interest (roi)” for the data to be considered
- `poi`: period of interest for observations (2-element array), (/first, last/) years to be considered (default = (/1900, 2100/))
- `poi_match`: “contained” / “center” / “overlap” / “encloses” / “outside” (default = “overlap”), e.g., “overlap” = period of observations (given by the corresponding global attribute of the observational data file) must overlap with the “period of interest (poi)” for the data to be considered

Optional GENERAL parameters

- `quantity`: determines quantities to be evaluated and plotted for grading (“mean”, “median” (not fully implemented yet))
- `ncdf`: enable to output to netCDF: either use “default” or give a full file name (default = no netCDF output)
- `interpolation`: flag determining regridding from simulation to observations’ pressure levels: 1 = linear, 2 = log; sign (-/+) = with/without extrapolation (default: -1 or +1 will be used, depending on levels)

Optional PANELING parameters (none of them mandatory, because there are defaults)

- `max_vert`: max. plots per column (default = 1)
- `max_hori`: max. plots per row (default = 1)
- `aux_info`: string to include in outfile name (default = period of interest)

Optional STYLE parameters (used by function `profile_plev`)

- `styleset`: style, e.g., “rghi15gmd” (see `diag_scripts/lib/ncl/style.ncl` for available styles) (default = “DEFAULT”)
- `colors`: override line colors (list), e.g., (/“black”/)
- `dashes`: overrides line styles (list), e.g., (/“0”/)
- `thicks`: override line thicknesses (list), e.g., (/2, 1, 1, 1, 1, 1/) * 6.
- `annots`: e.g., (/“data”/)
- `FuncCode`: overrides default function code for strings
- `font`: overrides default font (see www.ncl.ucar.edu/Document/Graphics/Resources/tx.shtml#txFont)

Optional PLOTTING parameters (used by function `profile_plev`)

- `ptop`: top pressure in hPa; “auto” = observational range from input file (default = minimum of pressure coordinate from observational data file)

- pbot: surface pressure in hPa; “auto” = observational range from input file (default = maximum of pressure coordinate from observational data file)
- zoom: zoom x-axis range; “yes” = consider only mean/median when determining x-axis range; “no” = use min/max/5%/95% values to determine x-axis range; alternatively, an explicit range can be specified, e.g., (/0, 300/) (default = “yes”)
- inline_legend: legend in the plot (True, False) (default = False)

3. lat_lon.ncl

Required diag_script_info attributes

- styleset: style (“DEFAULT”, “righi15gmd”; see diag_scripts/lib/ncl/style.ncl for available styles)
- font: font type (see www.ncl.ucar.edu/Document/Graphics/Resources/tx.shtml#txFont)
- mpProjection: map projection, e.g., “CylindricalEquidistant”; see <http://www.ncl.ucar.edu/Document/Graphics/Resources/mp.shtml#mpProjection> for available projections)

Optional diag_script_attributes

- range_option: 0 = as in nml, 1 = overlapping time period
- gsnZonalMean: plot zonal mean next to map (True, False)

4. perfmetrics_grading.ncl, perfmetrics_grading_collect.ncl, perfmetrics_main.ncl, perfmetrics_taylor.ncl, perfmetrics_taylor_collect.ncl

See “user settings” for *Performance metrics for essential climate parameters*.

33.4 Variables

- vmro3 (monthly mean, time level latitude longitude)
- vmrnox (monthly mean, time level latitude longitude)
- vmrco (monthly mean, time level latitude longitude)
- vmrc2h4 (monthly mean, time level latitude longitude)
- vmrc2h6 (monthly mean, time level latitude longitude)
- vmrc3h6 (monthly mean, time level latitude longitude)
- vmrc3h8 (monthly mean, time level latitude longitude)
- vmrch3coch3 (monthly mean, time level latitude longitude)
- tropoz (monthly mean, time latitude longitude)
- toz (monthly mean, time latitude longitude)
- vmrco
- vmro3

33.5 Observations and reformat scripts

Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

- Total column ozone (toz): NIWA (Bodeker et al., 2005) and GTO-ECV (Loyola and Coldewey-Egbers, 2012; Loyola et al., 2009)

Reformat scripts:

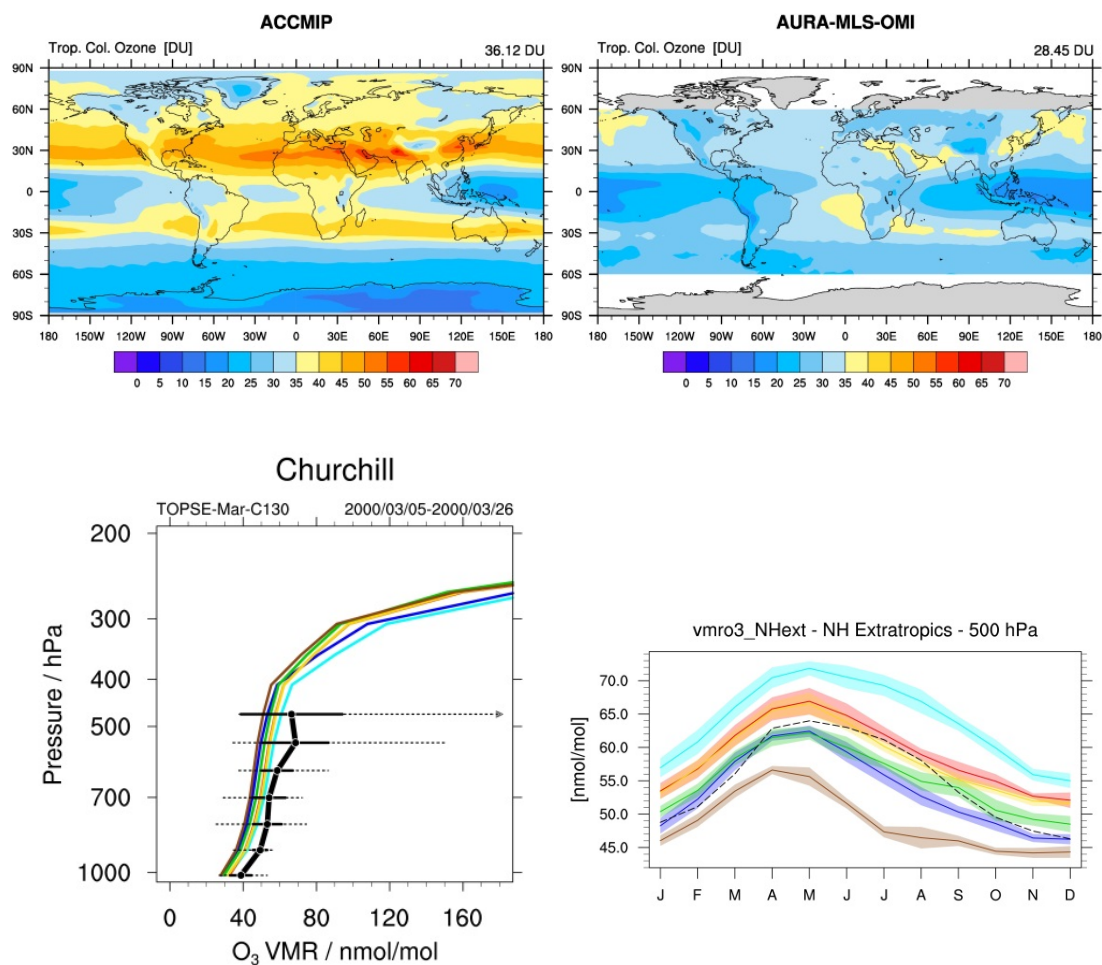
- reformat_scripts/obs/reformat_obs_NIWA.ncl

- reformat_scripts/obs/reformat_obs_GTO-ECV.ncl
- Tropospheric column ozone (tropoz): MLS/OMI (Ziemke et al., 2006)
Reformat script: reformat_scripts/obs/reformat_obs_AURA-MLS-OMI.ncl
- Ozonesonde profiles: Tilmes et al. (2012)
Reformat script: reformat_scripts/obs/reformat_obs_Tilmes.ncl
- CO surface mixing ratios: GLOBALVIEW (2010)
Reformat script: reformat_scripts/obs/reformat_obs_GLOBALVIEW.ncl
- Trace gas in-situ campaign composite data: Emmons et al. (2000)
Reformat script: reformat_scripts/obs/reformat_obs_Emmons.ncl

33.6 References

- Bodeker, G. E., Shiona, H., and Eskes, H.: Indicators of Antarctic ozone depletion, *Atmos. Chem. Phys.*, 5, 2603-2615, doi: 10.5194/acp-5-2603-2005, 2005.
- Emmons, L. K., Hauglustaine, D. A., Müller, J.-F., Carroll, M. A., Brasseur, G. P., Brunner, D., Staehelin, J., Thouret, V., and Marenco, A.: Data composites of airborne observation of tropospheric ozone and its precursors, *J. Geophys. Res.*, 105, 20497-20538, 2000.
- GLOBALVIEW-CO2: Cooperative Atmospheric Data Integration Project – Carbon Dioxide, CD-ROM, NOAA ESRL, Boulder, Colorado, available at: <ftp://ftp.cmdl.noaa.gov> (last access: 2 October 2014), 2010.
- Klinger, C., Quantitative evaluation of ozone and selected climate parameters in the chemistry-climate model EMAC, Master Thesis, Ludwig-Maximilians-Universität München, 2011.
- Loyola, D. and Coldewey-Egbers, M.: Multi-sensor data merging with stacked neural networks for the creation of satellite long-term climate data records, *EURASIP J. Adv. Sig. Pr.*, 2012, 1-10, doi: 10.1186/1687-6180-2012-91, 2012.
- Loyola, D. G., Coldewey-Egbers, R. M., Dameris, M., Garny, H., Stenke, A., Van Roozendael, M., Lerot, C., Balis, D., and Koukouli, M.: Global long-term monitoring of the ozone layer – a prerequisite for predictions, *Int. J. Remote Sens.*, 30, 4295-4318, doi: 10.1080/01431160902825016, 2009.
- Righi, M., V. Eyring, K.-D Gottschaldt, C. Klinger, F. Frank, P. Jöckel, and I. Cionni, Quantitative evaluation of ozone and selected climate parameters in a set of EMAC simulations, *Geosci. Model Dev.*, 8, 733-768, doi: 10.5194/gmd-8-733-2015, 2015.
- Tilmes, S., Lamarque, J.-F., Emmons, L. K., Conley, A., Schultz, M. G., Saunio, M., Thouret, V., Thompson, A. M., Oltmans, S. J., Johnson, B., and Tarasick, D.: Technical Note: Ozonesonde climatology between 1995 and 2011: description, evaluation and applications, *Atmos. Chem. Phys.*, 12, 7475-7497, doi: 10.5194/acp-12-7475-2012, 2012.
- Ziemke, J. R., Chandra, S., Duncan, B. N., Froidevaux, L., Bhartia, P. K., Levelt, P. F., and Waters, J. W.: Tropospheric ozone determined from Aura OMI and MLS: Evaluation of measurements and comparison with the Global Modeling Initiative's Chemical Transport Model, *J. Geophys. Res.*, 111, D19303, doi: 10.1029/2006JD007089, 2006.

33.7 Example plots



Performance metrics for essential climate parameters

34.1 Overview

The goal is to create a standard namelist for the calculation of performance metrics to quantify the ability of the models to reproduce the climatological mean annual cycle for selected “Essential Climate Variables” (ECVs) plus some additional corresponding diagnostics and plots to better understand and interpret the results. The namelist can be used to calculate performance metrics at different vertical levels (e.g., 5, 30, 200, 850 hPa as in Gleckler et al., 2008) and in four regions (global, tropics 20°N-20°S, northern extratropics 20°-90°N, southern extratropics 20°-90°S). As an additional reference, we consider the Righi et al. (2015) paper.

34.2 Available namelists and diagnostics

Namelists are stored in `nml/`

- `namelist_perfmetrics_CMIP5.xml`

Diagnostics are stored in `diag_scripts/`

- `perfmetrics_grading.ncl`: calculates grades according to a given metric, with different options for normalization. It requires fields precalculated by `perfmetrics_main.ncl`.
- `perfmetrics_grading_collect.ncl`: collects results from metrics previously calculated by `perfmetrics_grading.ncl` and passes them to the plotting functions.
- `perfmetrics_main.ncl`: calculates and (optionally) plots annual/seasonal cycles, zonal means, lat-lon fields and time-lat-lon fields from input monthly 2-d or 3-d (“T2M”, “T3Ms”) data. The calculated fields can be also plotted as difference w.r.t. a given reference model. They are also used as input to calculate grading metrics (see `perfmetrics_grading.ncl`).
- `perfmetrics_taylor.ncl`: calculates grades according to a given metric, with different options for normalization. It requires fields precalculated by `perfmetrics_main.ncl`.
- `perfmetrics_taylor_collect.ncl`: collects results from metrics previously calculated by `perfmetrics_taylor.ncl` and passes them to the plotting functions.

34.3 User settings

User setting files (cfg files) are stored in `nml/cfg_perfmetrics/CMIP5/`

1. `perfmetrics_grading.ncl`

diag_script_info attributes

- `MultiModelMean`: calculate multi-model mean (True, False)
- `MultiModelMedian`: calculate multi-model median (True, False)
- `metric`: applied metric (“RMSD” = root-mean square difference, “BIAS” = mean bias, “stddev_ratio” = ratio of standard deviations of var and ref (for Taylor diagrams only), “correlation” = pattern correlation of var and ref (for Taylor diagrams only)).
- `normalization`: applied normalization (“mean” = normalization with mean, “median” = normalization with media, “centered_median” = subtracting and dividing by the median, “stddev_mean” = normalization with subtracting the mean and dividing by the standard deviation)

2. `perfmetrics_grading_collect.ncl`

Required diag_script_info attributes

- `label_bounds`: min and max of the labelbar
- `label_scale`: bin width of the labelbar
- `disp_values`: switch on/off the grading values on the plot

Optional diag_script_info attributes

- `sort`: sort models in alphabetic order (excluding multi-model mean)
- `title`: plot title
- `scale_font`: scaling factor applied to the default font size

3. `perfmetrics_main.ncl`

diag_script_info attributes

- `plot_type`: plot type (“cycle” (time), “zonal” (plev, lat), “latlon” (lat, lon), “cycle_latlon” (time, lat, lon))
- `time_avg`: time averaging (“monthlyclim”, “seasonalclim”)
- `valid_fraction`: required fraction of valid values
- `level`: vertical level (hPa, “all” for no selection; set to “all” for zonal mean plots)
- `region`: averaging region (“Global”, “Tropics”, “NH extratropics”, “SH extratropics”)
- `grid`: regridding option (“finest”, “coarsest”, “ref”)
- `draw_plots`: create plots (True, False)
- `plot_diff`: create difference plots (only for zonal and lat-lon plots) (True, False)
- `plot_stddev`: plot standard deviation (“all”, “none”, “ref_model” or given model name)
- `legend_outside`: plot legend in a separate file (only for cycle plots) (True, False)
- `styleset`: plot style (only for cycle plots) (“CMIP5”, “DEFAULT”, “EMAC”)
- `t_test`: calculate t-test for difference plots (only for zonal and lat-lon plots) (True, False)
- `conf_level`: confidence level for the t-test (only for zonal and lat-lon plots)

4. `perfmetrics_taylor.ncl`

Required diag_script_info attributes

- `region`: averaging region (“Global”, “Tropics”, “NH extratropics”, “SH extratropics”)

- `time_avg`: time averaging (“monthlyclim”, “seasonalclim”)
- `metric`: selected metric (required but ignored by `permetrics_taylor.ncl`)
- `normalization`: type of metric normalization (required but ignored by `permetrics_taylor.ncl`)

5. `perfmetrics_taylor_collect.ncl`

diag_script_info attributes

- None.

34.4 Variables

- `hus` (atmos, monthly mean, longitude latitude lev time)
- `od550aer` (aero, monthly mean, longitude latitude time)
- `pr` (atmos, monthly mean, longitude latitude time)
- `rlut`, `rlutcs`, `rsut`, `rsutcs` (atmos, monthly mean, longitude latitude time)
- `ta` (atmos, monthly mean, longitude latitude lev time)
- `tas` (atmos, monthly mean, longitude latitude time)
- `ua` (atmos, monthly mean, longitude latitude lev time)
- `va` (atmos, monthly mean, longitude latitude lev time)
- `zg` (atmos, monthly mean, longitude latitude lev time)

34.5 Observations and reformat scripts

Note: (1) `obs4mips` data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-`obs4mips` data for download instructions.

- AIRS L3 (`hus` – `obs4mips`)
- CERES-EBAF (`rlut`, `rlutcs`, `rsut`, `rsutcs` – `obs4mips`)
- ERA-Interim (`tas`, `ta`, `ua`, `va`, `zg`, `hus` – `reformat_scripts/obs/reformat_obs_ERA-Interim.ncl`)
- ESACCI-AEROSOL (`od550aer` – `reformat_scripts/obs/reformat_obs_ESACCI-AEROSOL.ncl`)
- GPCP-SG (`pr` – `obs4mips`)
- MODIS-L3 (`od550aer` – `obs4mips`)
- NCEP (`tas`, `ta`, `ua`, `va`, `zg` – `reformat_scripts/obs/reformat_obs_NCEP.ncl`)

34.6 References

- Gleckler, P. J., K. E. Taylor, and C. Doutriaux, Performance metrics for climate models, *J. Geophys. Res.*, 113, D06104, doi: 10.1029/2007JD008972 (2008).
- Righi, M., Eyring, V., Klinger, C., Frank, F., Gottschaldt, K.-D., Jöckel, P., and Cionni, I.: Quantitative evaluation of oone and selected climate parameters in a set of EMAC simulations, *Geosci. Model Dev.*, 8, 733, doi: 10.5194/gmd-8-733-2015 (2015).

Precipitation dependence on soil moisture

35.1 Overview

The “precipitation dependence on soil moisture diagnostic” is used for the analysis of the coupling between soil moisture and precipitation in Taylor et al. 2012. The script `sm_pr_diag_nml.py` provides the diagnostic on a regular $5^\circ \times 5^\circ$ grid for the land surface between 60°S and 60°N . In the output plots (see example [Figure 1](#)) shading blue (red) indicates convective precipitation more likely over wetter (drier) soils.

35.2 Available namelists and diagnostics

Namelists are stored in `nml/`

- `namelist_sm_pr.xml`

Diagnostics are stored in `diag_scripts/`

- `sm_pr_diag_nml.py`: this script computes and plot the diagnostic “preference for afternoon precipitation over soil moisture anomalies” as in Fig.3 of Taylor et al. (2012).

35.3 Specific Routines

- `main(project_info)`: call all callable fuctions to read CMIP5 data, compute and plot the diagnostic
- `read_pr_sm_topo(project_info)`: read and extract input for computing the diagnostic from CMIP5 input data and topography from aux file `topo_var_5x5.gra`
- `get_smclim(sm, lon, time)`: compute monthly soil moisture climatology at 0600 LT (note that more strictly this is between 0600 and 0859 LT)
- `get_monthly_input(project_info, mn, time, lon, lat, time_bnds_1, pr, sm, fileout, samplefileout, verbosity)`: prepare and agregate by month the input data required by the F90 routines `global_rain_sm` and `sample_events`
- F90 routine `global_rain_sm`: identifies rain events (time and location) from 3-hourly rainfall

- F90 routine `sample_events`: creates two datasets of difference in soil moisture between the locations of rainfall maximum and minimum, dataset 1 is for the events themselves, dataset 2 for the same locations but in different years (i.e. non-events)
- `get_p_val(in_dir)`: compares the statistics of the two datasets above by resampling, to compute percentiles (`p_values`) of “preference for afternoon precipitation over soil moisture anomalies” as in Fig.3 of Taylor et al. 2012
- `write_nc(fileout, xs, ys, p_vals, project_info)`: save netCDF file with the diagnostic in a regular 5°x5° grid
- `plot_diagnostic(fileout, plot_dir, project_info)`: plot diagnostic and save to .png file

Installation procedure

The diagnostics uses two Fortran programs that have to be compiled and interfaced to Python. This is done via the F2PY - Fortran to Python interface generator. General instructions for this is:

```
cd diag_scripts
f2py --fcompiler=gfortran -c -m global_rain_sm global_rain_sm.f90
f2py --fcompiler=gfortran -c -m sample_events sample_events.f90
cd ../
<Run diagnostics as usual>
```

35.4 User settings

User setting files (cfg files) are stored in `nml/ cfg_sm_pr/`

35.5 Variables

- `pr` – precipitation (atmos, 3hr, time latitude longitude)
- `mrsos` – moisture in upper portion of soil column (land, 3hr, time latitude longitude)

35.6 Observations and reformat scripts

None.

35.7 References

- Taylor, C. M., R. A. M. de Jeu, F. Guichard, P. P. Harris, and W. A. Dorigo (2012), Afternoon rain more likely over drier soils, *Nature*, 489(7416), 423-426, doi: 10.1038/nature11377.

35.8 Example plots

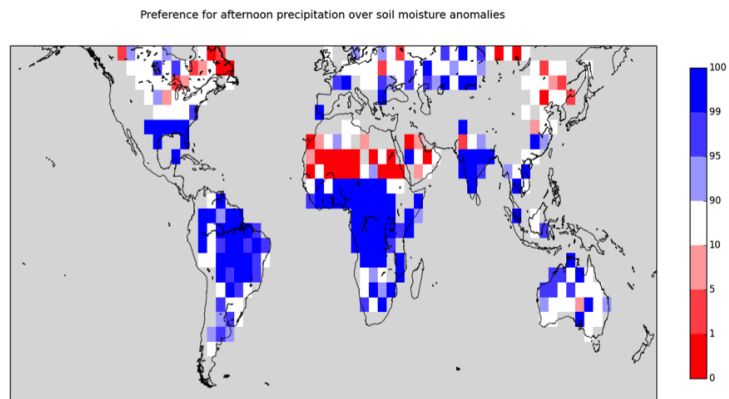


Figure 1: CNRM-CM5_sm_pr_diag_plot (CNRM-CM5, 3hr, amip, r1i1p1, 1999-2008).

36.1 Overview

The sea ice diagnostics cover sea ice extent, concentration, and thickness. Work is underway to include other variables and processes in future releases of the ESMValTool. Current diagnostics include time series of Arctic and Antarctic sea ice area and extent (calculated as the total area (km²) of grid cells with sea ice concentrations (sic) of at least 15%). Also included are the seasonal cycle of sea ice extent, and polar stereographic contour and polar contour difference plots of Arctic and Antarctic sea ice concentration and sea ice thickness.

36.2 Available namelists and diagnostics

Namelists are stored in nml/

- namelist_Sealce.xml

Diagnostics are stored in diag_scripts/

- SeaIce_tsline.ncl: time series line plots of total sea ice area and extent (accumulated) for northern and southern hemispheres with optional multi-model mean and standard deviation. One value is used per model per year, either annual mean or the mean value of a selected month.
- SeaIce_ancyc.ncl: as SeaIce_tsline.ncl, but for the annual cycle (multi-year monthly mean values).
- SeaIce_polcon.ncl: polar stereographic plots of sea ice concentration (= sea ice area fraction) and sea ice thickness for individual models or observational data sets, for Arctic and Antarctic regions with flexible paneling of the individual plots. The edges of sea ice extent can be highlighted via an optional red line.
- SeaIce_polcon_diff.ncl: polar stereographic plots of sea ice area concentration and thickness difference between individual models and reference data (e.g., an observational data set) for both Arctic and Antarctic with flexible paneling of the individual plots. All data are regridded to a common grid (1°x1°) before comparison.

36.3 User settings

User setting files (cfg files) are stored in nml/cfg_Sealce/

- region: label of region to be plotted (“Arctic”, “Antarctic”); make sure to specify correct observational data for the selected region in the sea ice namelist.
- month: “A” = annual mean, “3” = March (Antarctic), “9” = September (Arctic)
- styleset: “CMIP5”, “DEFAULT”
- fill_pole_hole: fill observational hole at North Pole, default = False
- legend_outside: True: draw legend in an extra plot

Settings specific to SeaIce_polcon, SeaIce_polcon_diff, SeaIce_ancyc

- range_option: 0 = use each model's whole time range as specified in namelist, 1 = use only intersection of all time ranges

Setting specific to SeaIce_tsline.ncl and SeaIce_ancyc.ncl

- multi_model_mean: plots multi-model mean and standard deviation (“y”, “n”)
- EMs_in_lg: create legend label for each individual ensemble member (True, False)

Settings specific to SeaIce_polcon.ncl and SeaIce_polcon_diff.ncl

- contour_extent: draw a red contour line for sic extent in polar stereographic plots (“y”, “n”)
- max_vert: max. number of rows on a panel page (vertical)
- max_hori: max. number of columns on a panel page (horizontal)
- max_lat: Antarctic plotted from 90°S up to this latitude
- min_lat: Arctic plotted from 90°N up to this latitude
- PanelTop: tune to get full title of uppermost row (1 = no top margin, default = 0.99)

Settings specific to SeaIce_polcon_diff.ncl

- ref_model: reference model, as specified in annotations; if this string is not found, the routine will print a list of valid strings before stopping
- dst_grid: path to destination grid file for Climate Date Operators (CDO), required by cdo_remapdis; e.g.: “./diag_scripts/aux/CDO/cdo_dst_grid_g010”
- grid_min: min. contour value (default = -1.0)
- grid_max: max. contour value (default = 1.0)
- grid_step: step between contours (default = 0.2)
- grid_center: value to center the color bar (default = 0.0)

36.4 Variables

- sic (sea ice, monthly mean, longitude latitude time)
- sit (sea ice, monthly mean, longitude latitude time)

36.5 Observations and reformat scripts

Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

- National Snow and Ice Data Center (NSIDC)

Reformat script: reformat_scripts/obs/reformat_obs_NSIDC.ncl

- Hadley Centre Sea Ice and Sea Surface Temperature data set (HadISST)

Reformat script: reformat_scripts/obs/reformat_obs_HadISST.ncl

- Pan-Arctic Ice Ocean Modelling and Assimilation System (PIOMAS)

Reformat script: reformat_scripts/obs/reformat_obs_PIOMAS.f90

36.6 References

- Bräu, M.: Sea-ice in decadal and long-term simulations with the Max Planck Institute Earth System Model, Bachelor thesis, LMU, 2013.
- Hübner, M.: Evaluation of Sea-ice in the Max Planck Institute Earth System Model, Bachelor thesis, LMU, 2013.

36.7 Example plots

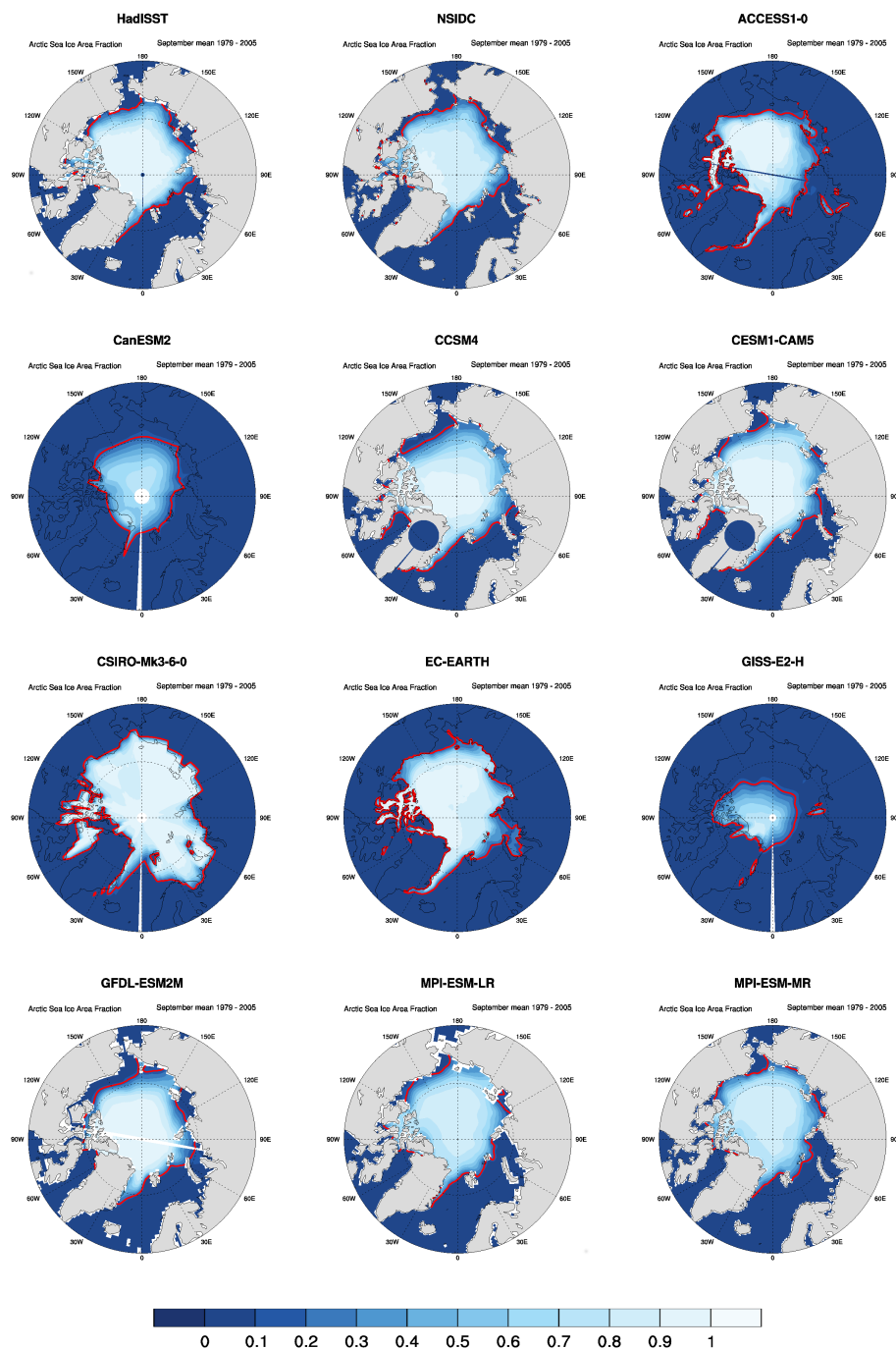


Figure 1: Polar-stereographic contour maps (Arctic) of sea ice concentration averaged over the period 1979-2005 from HadISST and NSIDC observations, as well as historical CMIP5 simulations from different Earth system models. The red line indicates the sea ice extent (i.e., sea ice concentration of 15%).

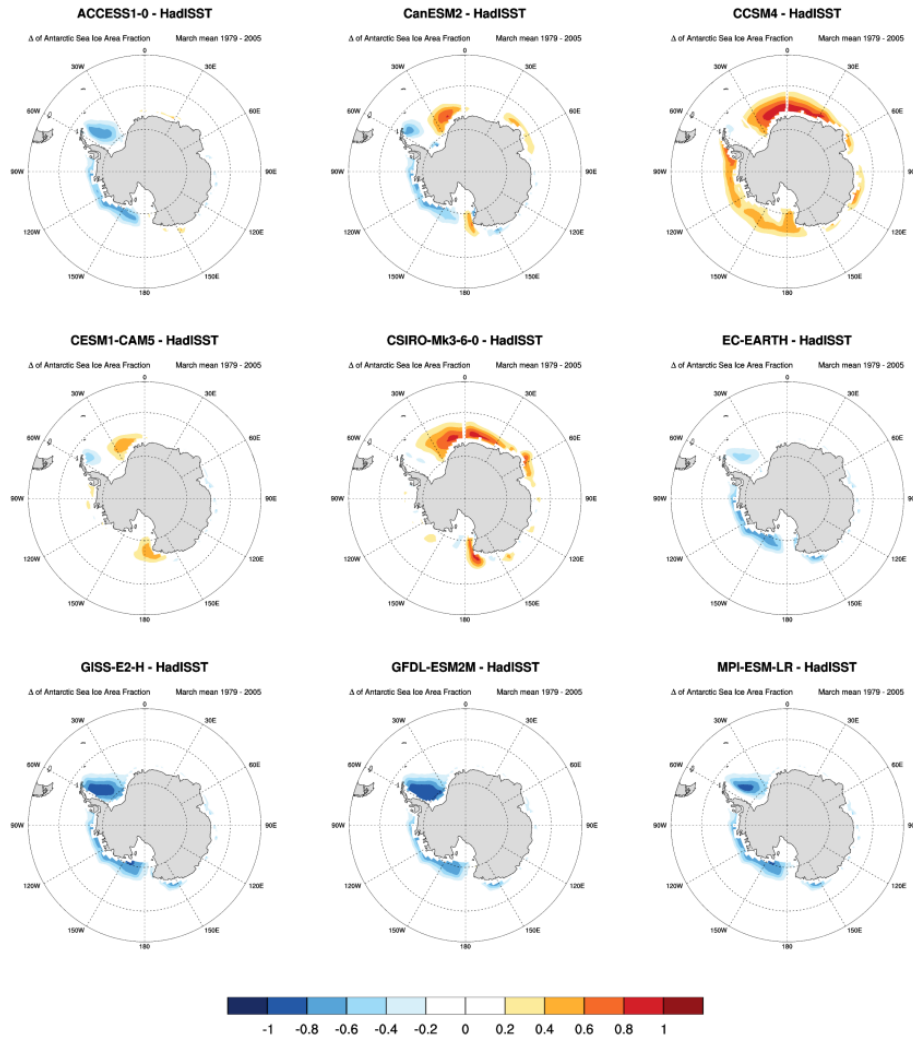


Figure 2: Polar-stereographic projections (Antarctic) of the difference in sea ice concentration between historical CMIP5 simulations from different Earth system models and HadISST observations (1979-2005). Red (blue) colors indicate a positive (negative) bias of the respective model towards observations.

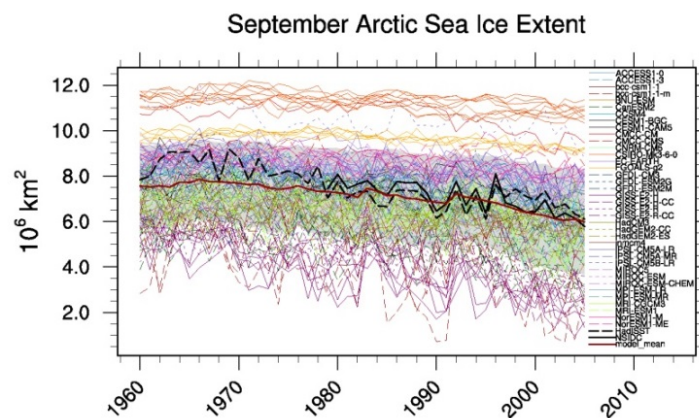


Figure 3: Timeseries (1960-2005) of September Arctic sea ice extent from different historical CMIP5 Earth system model simulations, and HadISST (black, dashed) and NSIDC (black, solid) observations. The thick red line represents the multi-model mean. Sea ice extent is the total area of all grid cells with a sea ice concentration of at least 15%.

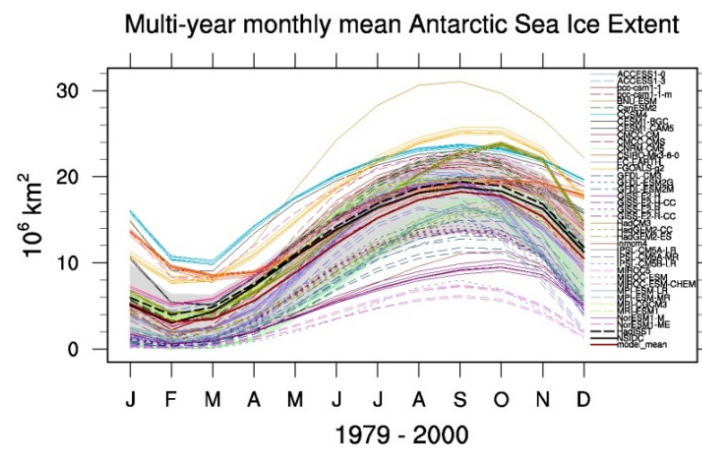


Figure 4: Same as Figure 3, but for the annual cycle of Antarctic sea ice extent.

37.1 Overview

Three South Asian Summer Monsoon (SASM) namelists for the basic climatology, seasonal cycle, intra-seasonal and inter-annual variability and key teleconnections have been implemented into the ESMValTool focusing on SASM rainfall and horizontal winds in June-September (JJAS). The goal is to provide a comprehensive overview of the basic features of the South Asian Monsoon for a certain model compared to other models, observational and reanalysis data sets for a range of processes and variables relevant to the monsoon.

37.2 Available Namelists and Diagnostics

Namelists are stored in nml/

- namelist_SAMonsoon_AMIP.xml
- namelist_SAMonsoon_daily.xml
- namelist_SAMonsoon.xml

Diagnostics are stored in diag_scripts/

- SAMonsoon_precip_basic.ncl
- SAMonsoon_precip_daily.ncl
- SAMonsoon_precip_domain.ncl
- SAMonsoon_precip_IAV.ncl
- SAMonsoon_precip_propagation.ncl
- SAMonsoon_precip_seasonal.ncl
- SAMonsoon_teleconnections.ncl
- SAMonsoon_wind_basic.ncl
- SAMonsoon_wind_IAV.ncl
- SAMonsoon_wind_seasonal.ncl

37.3 User settings

User setting files (cfg files) are stored in `nml/cfg_SAMonsoon/`

1. SAMonsoon_wind_basic.ncl

Required diag_script_info attributes

- `styleset`: “CMIP5”, “DEFAULT”
- `season`: season in letters, e.g., “JJAS”
- `latrange_basic`: min. and max. latitude of crop area (array)
- `lonrange_basic`: min. and max. longitude of crop area (array)
- `cn_levels_mean_basic`: contour levels for mean plots (array)
- `cn_levels_mean_diff_basic`: contour levels for mean difference plots (array)
- `cn_levels_stddev_basic`: contour levels for standard deviation plots (array)
- `cn_levels_stddev_diff_basic`: contour levels for standard deviation difference plots (array)
- `my_region`: monsoon region (must be set to “SA”)

2. SAMonsoon_wind_seasonal.ncl, SAMonsoon_wind_IAV.ncl

Required diag_script_info attributes

- `styleset`: “CMIP5”, “DEFAULT”
- `season`: season in letters, e.g., “JJAS”
- `multi_model_mean`: calculate multi-model mean (“y”, “n”)

3. SAMonsoon_precip_basic.ncl, SAMonsoon_precip_seasonal.ncl, SAMonsoon_precip.ncl, SAMonsoon_precip_domain.ncl, SAMonsoon_precip_basic_daily.ncl, SAMonsoon_precip_IAV.ncl, SAMonsoon_precip_propagation.ncl

Required diag_script_info attributes

- `season`: season in letters, e.g., “JJAS”

Settings specific for the precip basic `plot_type`

- `latrange_basic`: min. and max. latitude (2-element array)
- `lonrange_basic`: min. and max longitude (2-element array)
- `cn_levels_mean_basic`: contour levels for mean plots (n-element array)
- `cn_levels_mean_diff_basic`: contour levels for mean difference plots (n-element array)
- `cn_levels_stddev_basic`: contour levels for standard deviation plots (n-element array)
- `cn_levels_stddev_diff_basic`: contour levels for standard deviation difference plots (n-element array)
- `cn_levels_stddev_norm_basic` (only `SAMonsoon_precip_basic_daily.ncl`): contour levels for standard deviation plots (n-element array)
- `cn_levels_stddev_norm_diff_basic` (only `SAMonsoon_precip_basic_daily.ncl`): contour levels for standard deviation difference plots (n-element array)

Settings specific for the precip seasonal `plot_type` (`SAMonsoon_precip_seasonal.ncl`)

- `latrange_seasonal`: min. and max. latitude of crop area (2-element array)
- `lonrange_seasonal`: min. and max. longitude of crop area (2-element array)
- `styleset`: “CMIP5”, “DEFAULT”
- `multi_model_mean`: calculate multi-model mean (“y”, “n”)

Settings shared by the precip global domain/intensity plot_type

- summer_season: “MJJAS”
- winter_season: “NDJFM”

Settings specific for the precip global domain plot_type

- latrange_global: min. and max. latitude (2-element array)
- cn_levels_global_domain_mean: contour levels for global domain plots (array)
- cn_levels_global_domain_diff: contour levels for global domain difference plots (array)
- high_intensity_cutoff_mmday: cut-off for monsoon domain definition in mm day-1

Settings specific for the precip intensity plots

- cn_levels_intensity_mean: contour levels for intensity mean plots (n-element array)
- cn_levels_intensity_diff: contour levels for intensity difference plots (n-element array)
- abs_cmap: path and filename of color map for absolute values
- diff_cmap: path and filename of color map for difference values
- my_region: monsoon region (must be set to “SA”)

Settings specific for daily precipitation plots (SAMonsoon_precip_daily.ncl)

- isv_latrange: min. and max. latitude of crop area (2-element array)
- isv_lonrange: min. and max. longitude of crop area (2-element array)
- bob_latrange: min. and max. latitude of crop area for Bay of Bengal (2-element array)
- bob_lonrange = min. and max. longitude of crop area for Bay of Bengal (2-element array)
- eio_latrange: min. and max. latitude of crop area for eastern equatorial Indian Ocean (2-element array)
- eio_lonrange: min. and max. longitude of crop area for eastern equatorial Indian Ocean (2-element array)
- filter_min: lower limit used for filtering
- filter_max: upper limit used for filtering
- filter_weights: number of filter weights to use

Settings specific for daily propagation plots (SAMonsoon_precip_propagation.ncl)

- prop_isv_latrange: min. and max. latitude of crop area (2-element array)
- prop_isv_lonrange: min. and max. longitude of crop area (2-element array)
- prop_lag_latrange: min. and max. latitude of crop area for lag computation along longitude (2-element array)
- prop_lag_lonrange: min. and max. longitude of crop area for lag computation along latitude (2-element array)
- prop_ref_latrange: min. and max. latitude of crop area for lag reference area (2-element array)
- prop_ref_lonrange: min. and max. longitude of crop area for lag reference area (2-element array)
- prop_filter_mxlage: size of lag
- ihp: Lanczos switch for low/high band pass filter
- nWgt: total number of weights for Lanczos
- sigma: Lanczos sigma
- min_days: min days in filtering

- max_days: max day in filtering
- fca: start frequency to compute Lanczos weights
- fcb: end frequency to compute Lanczos weights

4. SAMonsoon_teleconnections.ncl

Required diag_script_info attributes

- monsoon_latrange: min. and max. latitude of crop area for Indian Monsoon (2-element array)
- monsoon_lonrange: min. and max. longitude of crop area for Indian Monsoon (2-element array)
- nino34_latrange: min. and max. latitude of Niño3.4 region (2-element array)
- nino34_lonrange: min. and max. longitude of Niño3.4 region (2-element array)
- tele_latrange: min. and max. latitude of crop area for overview area (2-element array)
- tele_lonrange: min. and max. longitude of crop area for overview area (2-element array)
- cn_levels_mean_basic: contour levels for mean plot (n-element array)
- cn_levels_mean_diff_basic: contour levels for mean difference plot (n-element array)
- cn_levels_stddev_basic: contour levels for standard deviation plot (n-element array)
- cn_levels_stddev_diff_basic: contour levels for standard deviation difference plot (n-element array)

37.4 Variables

- pr (atmos, daily/monthly, longitude latitude time)
- ts (atmos, monthly, longitude latitude time)
- ua (atmos, monthly, longitude latitude lev time)
- va (atmos, monthly, longitude latitude lev time)

37.5 Observations and Reformat Scripts

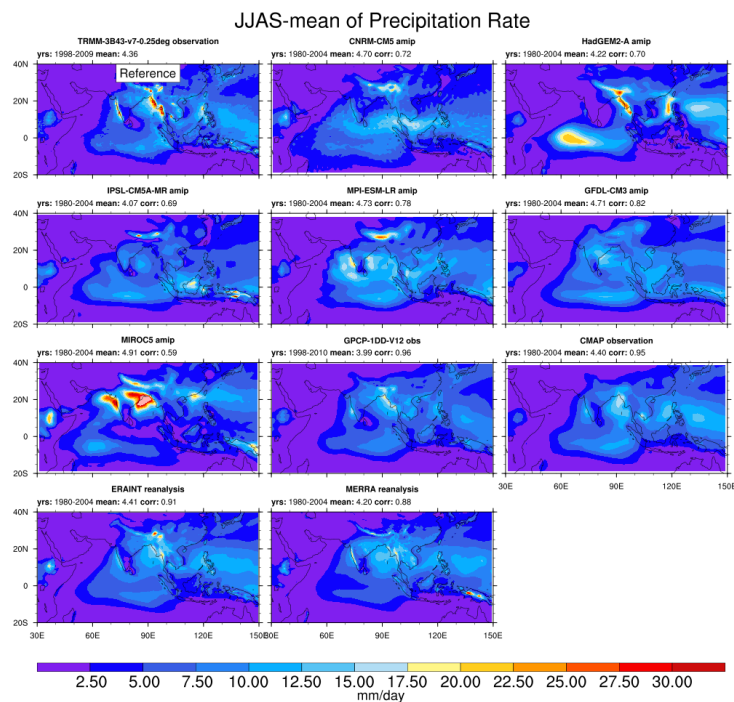
Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

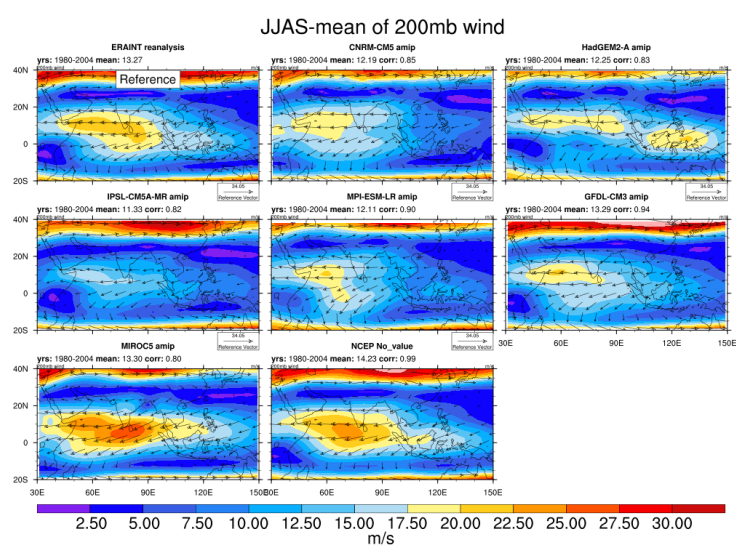
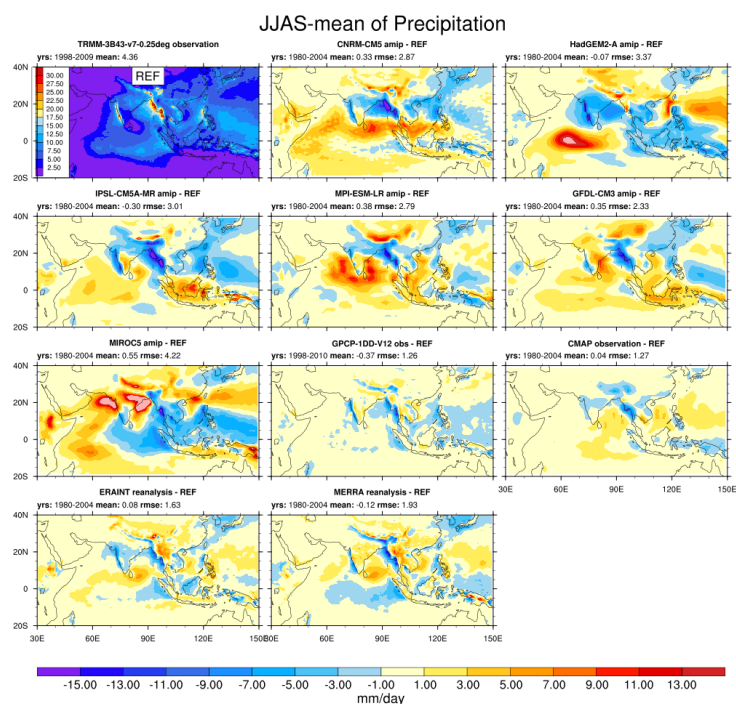
- ERA-Interim (ua, va, pr – reformat_scripts/obs/reformat_obs_ERA-Interim.ncl, reformat_obs_ERA-Interim_surffluxes.ncl)
- NCEP (ua, va – reformat_scripts/obs/reformat_obs_NCEP.ncl)
- TRMM-L3 (pr, monthly means – obs4mips)
- TRMM_3B42 (pr, daily means – reformat_scripts/obs/reformat_obs_TRMM-3B42-daily.ncl)
- HadISST (ts – reformat_scripts/obs/reformat_obs_HadISST.ncl)
- GPCP-1DD (pr, daily means – obs4mips)
- GPCP-SG (pr, monthly means – obs4mips)
- MERRA (pr – obs4mips)
- CMAP (pr – reformat_scripts/obs/reformat_obs_CMAP.ncl)

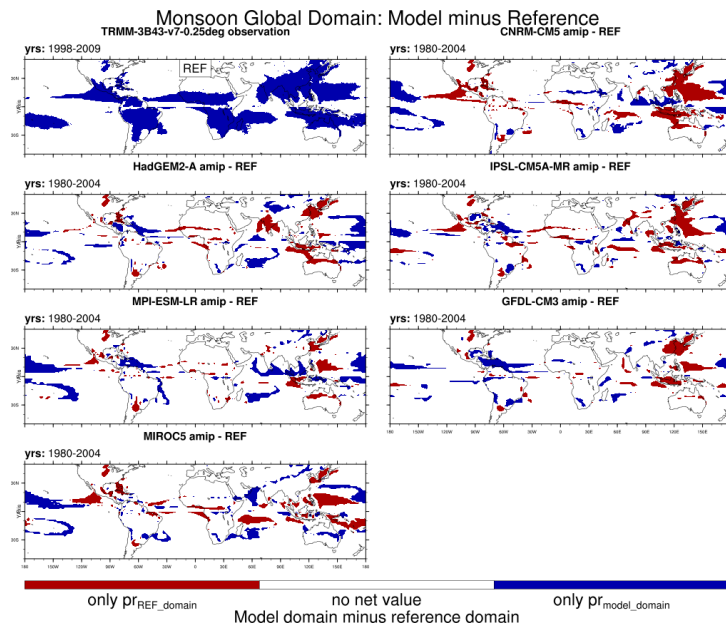
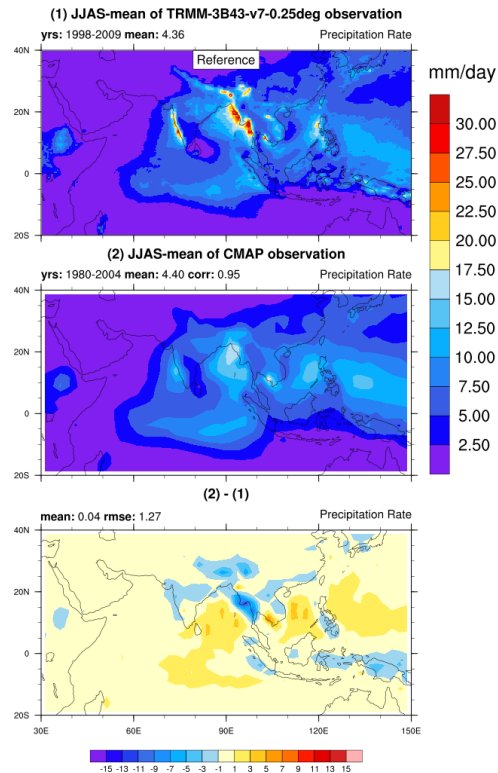
37.6 References

- Sperber, K. R., et al., The Asian summer monsoon: an intercomparison of CMIP5 vs. CMIP3 simulations of the late 20th century, *Clim Dyn* (2013) 41:2711–2744, doi: 10.1007/s00382-012-1607-6, 2012.
- Lin, Jia-Lin, Klaus M. Weickman, George N. Kiladis, Brian E. Mapes, Siegfried D. Schubert, Max J. Suarez, Julio T. Bacmeister, Myong-In Lee, 2008: Subseasonal Variability Associated with Asian Summer Monsoon Simulated by 14 IPCC AR4 Coupled GCMs. *J. Climate*, 21, 4541-4567. doi: <http://dx.doi.org/10.1175/2008JCLI1816.1>.
- Webster, P. J., and S. Yang, 1992: Monsoon and ENSO: Selectively interactive systems. *Quart. J. Roy. Meteor. Soc.*, 118, 877-926. (Webster-Yang dynamical monsoon index)
- Goswami, B. N., B. Krishnamurthy, and H. Annamalai, 1999: A broad-scale circulation index for interannual variability of the Indian summer monsoon. *Quart. J. Roy. Meteor. Soc.*, 125, 611-633. (Goswami dynamical monsoon index)
- Wang, B., and Z. Fan, 1999: Choice of south Asian summer monsoon indices. *Bull. Amer. Meteor. Soc.*, 80, 629-638. (Wang-Fan dynamical monsoon index)
- Wang B., J. Liu, H. J. Kim, P. J. Webster, and S. Y. Yim, Recent change of global monsoon precipitation (1979-2008), *Climate Dynamics*, doi: 10.1007/s00382-011-1266-z, 2011. (Intensity/Monsoon domain reference)

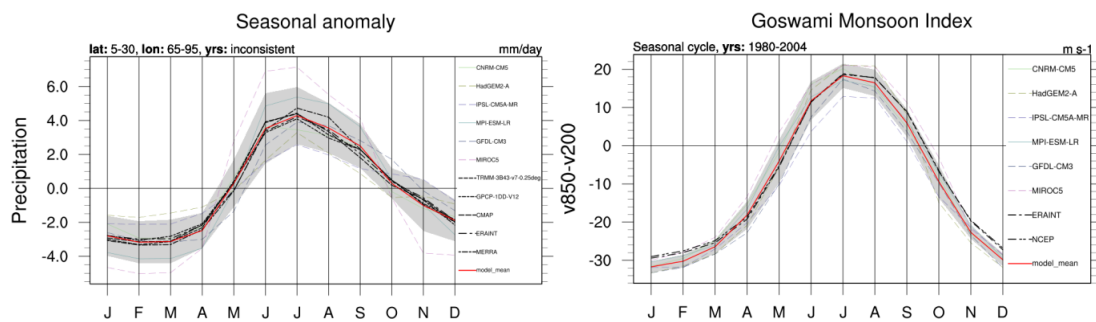
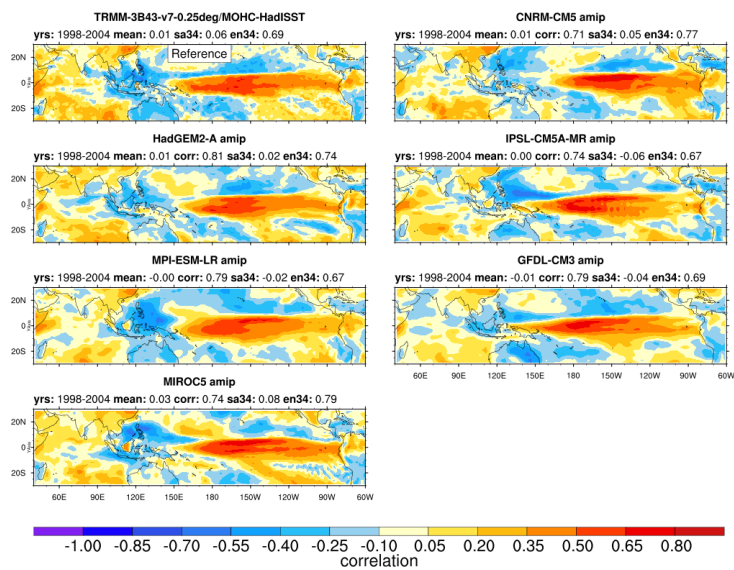
37.7 Example plots



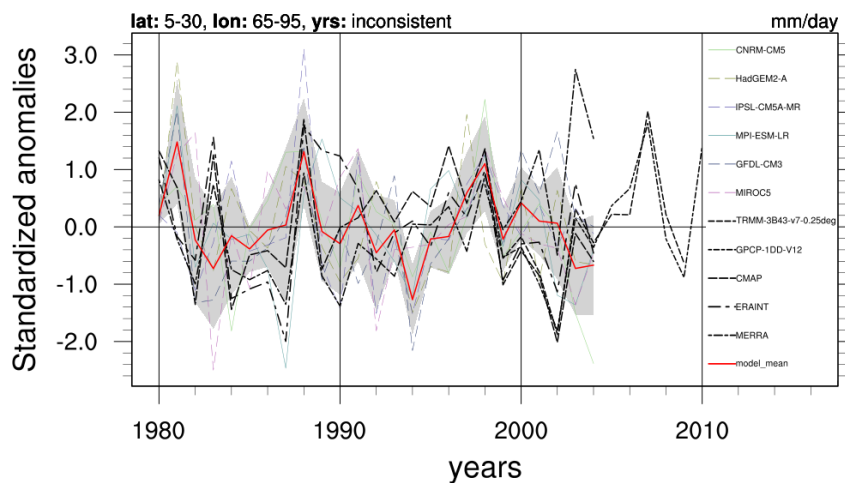


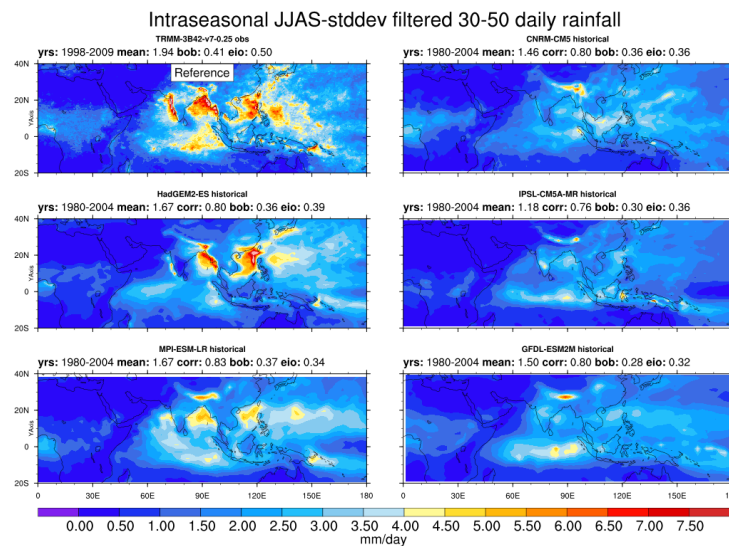
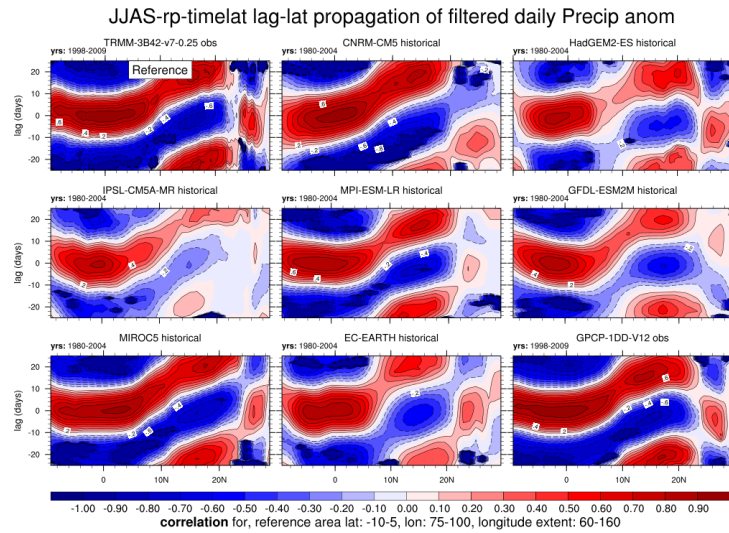


JJAS-mean precip-Nino3.4 SST teleconnection, mean plots



IAV Precipitation (JJAS)





38.1 Overview

The diagnostics compare model estimates of cloud, radiation and surface turbulent flux variables with suitable observational data sets. Seasonal mean maps of TOA, total and clear-sky, outgoing short wave (SW) and long wave (LW) radiation are constructed for the Southern Ocean region (30°S-65°S) comparing available model data with CERES-EBAF satellite observations (Loeb et al., 2005). Absolute values, model to satellite differences and total minus clear sky flux values are plotted through standard namelists. Combined with TOA radiation, seasonal mean maps of total cloud cover, vertically integrated cloud liquid water and cloud ice are also plotted with CloudSat data (Stephens et al., 2002) providing an observational constraint. Finally, seasonal mean maps of surface sensible and latent heat fluxes over the same region compare model estimates with the Woods Hole OAflux observations (Yu et al., 2008). For each of these quantities (TOA radiation, cloud and surface flux variables) climatological annual cycles are constructed using zonal means averaged separately over 3 latitude bands: (i) 30°S-65°S, the entire Southern Ocean, (ii) 30°S-45°S, the subtropical Southern Ocean, and (iii) 45°S-65°S, the mid-latitude Southern Ocean. Finally, annual means of each variable (models and observations) are constructed as zonal means, plotted over 30°S-65°S, and as latitudinal means (over 30°S-65°S) plotted around a longitude circle.

38.2 Available namelists and diagnostics

Namelists are stored in nml/

- namelist_SouthernHemisphere.xml

Diagnostics are stored in diag_scripts/

- SouthernHemisphere.py
- SouthernHemisphere_scatter.py

38.3 User settings

User setting files (cfg files) are stored in nml/cfg_SouthernHemisphere/

1. SouthernHemisphere.py and SouthernHemisphere_scatter.py

Diag_script_info attributes

General options for all diagnostics

- plot_clouds: True, False
- plot_fluxes: True, False
- plot_radiation: True, False
- plot_scatter: True, False
- plot_background_grid: True, False
- plot_total_cover: True, False
- plot_liquid_path: True, False
- plot_ice_path: True, False
- plot_optical_depth: True, False
- plot_flux_maps: True, False
- plot_radiation_maps: True, False

Common sub keys for all diagnostics (plot_): X_maps, X_averages, sub_areas

- plot_lat_averages: True, False
- plot_lon_averages: True, False
- plot_monthly_averages: True, False
- plot_sub_areas: True, False
- mask_unwanted_values: mask values (e.g., missing values) exceeding “mask_limit_low” and “mask_limit_high” (True, False)
- mask_limit_low: lower threshold used for creating a mask (if “mask_unwanted_values” = True)
- mask_limit_high: upper threshold used for creating a mask (if “mask_unwanted_values” = True)

Configuration for Southern Hemisphere maps and plots. All plots will be generated for all areas but only monthly averages for sub_areas.

- areas: name of area(s), defined separately for each area (see below), e.g., default
- sub_areas: name of sub-area(s), defined separately for each sub-area (see below), e.g., northern southern
- scatter_areas: cloud vs radiation scatter plot areas; areas must be defined separately (see below), e.g., default
- seasons: which months to plot for each contour and lat/lon mean plots (e.g., DJF MAM JJA SON); each season has to be defined separately (see below)

Definition of each area, e.g., [SouthernHemisphere_default]

- lat_min: min. latitude
- lat_max: max. latitude
- lon_min: min. longitude
- lon_max: max. longitude
- stride: color difference interval (how many units per color/shade); if set to 0, the stride is calculated automatically using “maxshades” (max. number of colors/shades)
- maxshades: max. number of colors/shades

The contour_limits_* are contour map limits and are given by 3 or 5 integers (min, max, diff, [dev_min, dev_max]). The min and max values define the limits for model maps (and the clear sky (cs) variant). The diff value gives a range [-diff, diff] for difference maps of model - obs (and cs). The last range [dev_min, dev_max] is the range for model/obs - model/obs (cs).

- hfls, hfss (latent, sensible heat); rlut, rsut (long, shortwave radiation).
- contour_limits_clt: contour limits for total cloud cover
- contour_limits_clivi: contour limits for ice water path
- contour_limits_clwvi: contour limits for total condensed water path
- contour_limits_hfls: contour limits for latent heat
- contour_limits_hfss: contour limits for sensible heat
- contour_limits_rlut: contour limits for longwave radiation
- contour_limits_rsut: contour limits for shortwave radiation
- contour_limits_rlds: contour limits for surface downwelling longwave radiation
- contour_limits_rsds: contour limits for surface downwelling shortwave radiation

Color maps for map plots. You may change the maps as you will, just google “python matplotlib colormaps” for examples. Color maps can be inverted by adding ‘_r’ at the end.

- colourmap_clouds: e.g., Blues
- colourmap_model: e.g., RdYlGn_r
- colourmap_diff: e.g., jet
- colourmap_dev: e.g., cool hot_r

Definition of sub areas, e.g., [SouthernHemisphere_northern]

- lat_min: min. latitude
- lat_max: max. latitude
- lon_min: min. longitude
- lon_max: max. longitude

Definition of months covered by each season – 1 is January and so forth, e.g., [SouthernHemisphere_season_DJF]

- season_months: e.g., 12 1 2

Configuration for cloud vs radiation scatter plots

- lat_min: min. latitude
- lat_max: max. latitude
- lon_min: min. longitude
- lon_max: max. longitude
- points: number of bins

38.4 Variables

- clt: total cloud cover fraction (atmos, monthly mean, time latitude longitude)
- clivi: cloud ice water path (atmos, monthly mean, time latitude longitude)
- clwvi: total condensed water path (atmos, monthly mean, time latitude longitude)
- hfls: latent heat flux (atmos, monthly mean, time latitude longitude)
- hfss: sensible heat flux (atmos, monthly mean, time latitude longitude)
- rlut: top of atmosphere outgoing longwave radiation (atmos, monthly mean, time latitude longitude)

- rlutcs: top of atmosphere outgoing clear-sky longwave radiation (atmos, monthly mean, time latitude longitude)
- rsut: top of atmosphere outgoing shortwave radiation (atmos, monthly mean, time latitude longitude)
- rsutcs: top of atmosphere outgoing clear-sky shortwave radiation (atmos, monthly mean, time latitude longitude)

38.5 Observations and reformat scripts

Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

- CERES-EBAF (obs4mips)
- WHOI-OAFlux

Reformat script: reformat_scripts/obs/reformat_obs_WHOI-OAFlux.ncl

- CloudSat-L3

Reformat script: reformat_scripts/obs/reformat_obs_cloudsat.bash

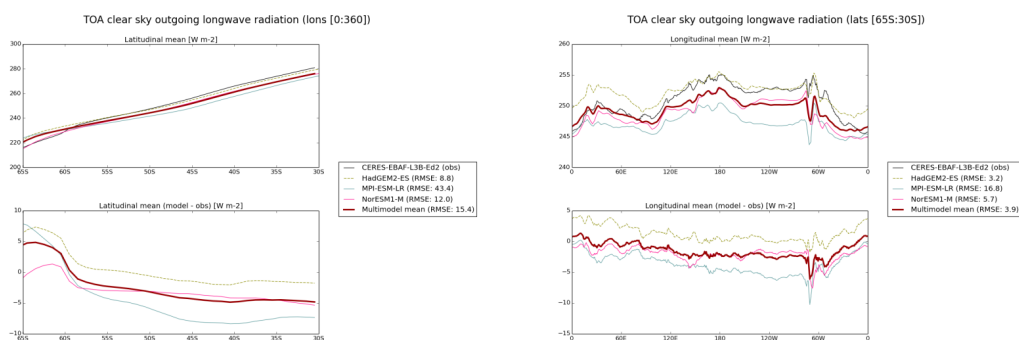
- MODIS-L3-C6

Reformat script: reformat_scripts/obs/reformat_obs_MODIS-L3-C6.ncl

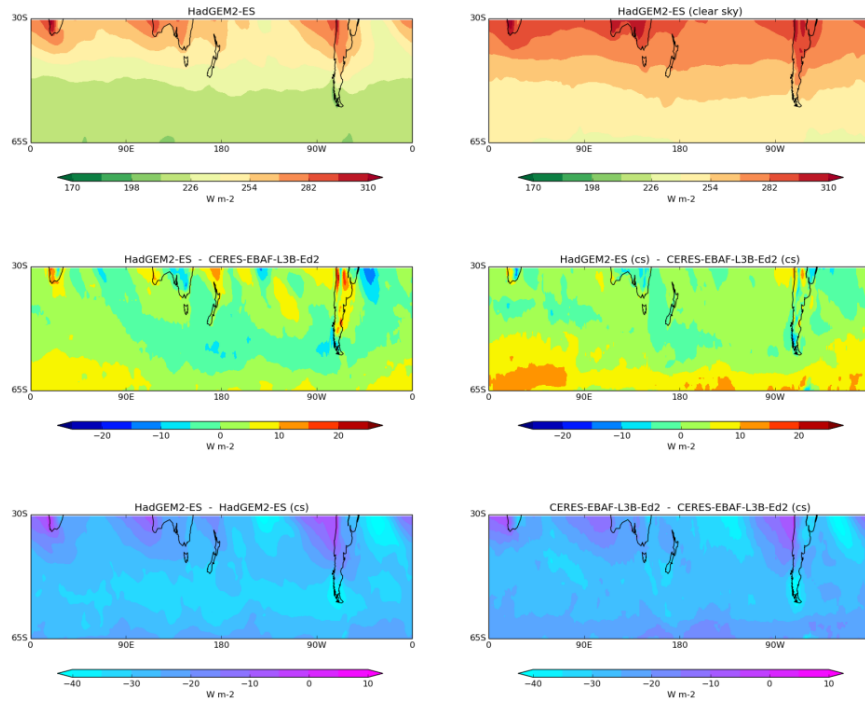
38.6 References

None.

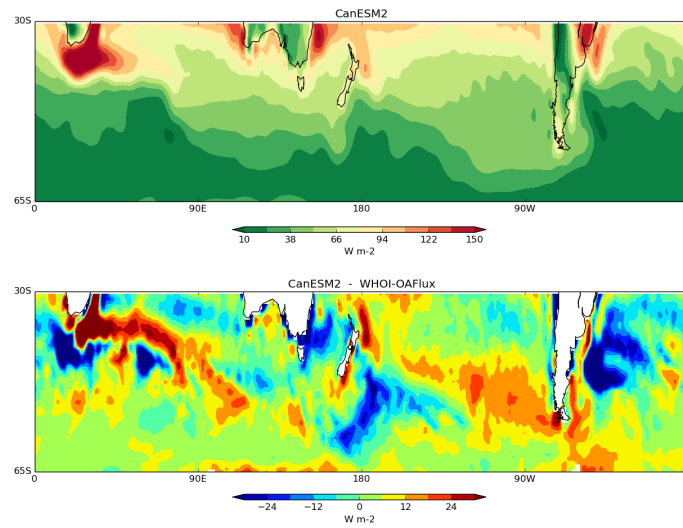
38.7 Example plots



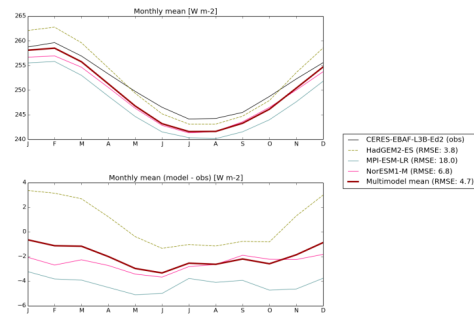
TOA outgoing longwave radiation for months: DJF



Surface latent heat flux for months: DJF



TOA clear sky outgoing longwave radiation (lats [65S:30S], lons [0:360])



39.1 Overview

These diagnostics include polar stereographic (difference) maps to compare the monthly/annual mean sea surface temperature, salinity and wind stress from ESMs with ERA-Interim data. Furthermore, there are scripts to plot the differences in the area mean vertical profiles of temperature and salinity between models and data from the World Ocean Atlas (Antonov et al., 2010; Locarnini et al., 2010). The ocean mixed layer thickness from models can be compared with that obtained from the Argo floats (Dong et al., 2008), again using polar stereographic (difference) maps. Finally, the Antarctic Circumpolar Current strength, as measured by water mass transport through the Drake Passage, is calculated using the same method as in the CDFTOOLS package (CDFtools). This diagnostic can be used to calculate the transport through other section as well, but is only available for EC-Earth/NEMO output for which all grid information is available.

39.2 Available namelists and diagnostics

Namelists are stored in `nml/`

- `namelist_SouthernOcean.xml`

Diagnostics are stored in `diag_scripts/`

- `SouthernOcean_polcon.ncl`: create polar stereographic plots for ocean mixed layer thickness, sea surface salinity and sea surface temperature.
- `SouthernOcean_polcon_diff.ncl`: create polar stereographic plots of the difference between individual models and reference data for ocean mixed layer thickness, sea surface temperature and eastward and northward wind stress. All data are regridded to a common grid using the ESMF regridding software.
- `SouthernOcean_vector_polcon_diff.ncl`: create polar stereographic contour plots of the difference between individual model data and reference data similar to `SouthernOcean_polcon_diff.ncl`, but on top plots vectors (magnitude and direction) for both the individual models and the reference data. Currently it is used for wind stress, but it should be possible to use it for other variables with `u` and `v` components as well. All data are regridded to a common grid using the ESMF regridding software.
- `SouthernOcean_areamean_vertconplot.ncl`: calculate the average sea water salinity and temperature over a subdomain from model data and create a Hovmoller-like diagram with time and depth on the axes. All data are regridded to a common grid using the ESMF regridding software.

- SouthernOcean_transport.ncl: calculate the sea water volume transport across a section from the variables `uo` and `vo` using a similar approach as is done in the CDFTOOLS package. **Currently only available for EC-Earth/Nemo output** as the calculations are performed using `uo` and `vo` on a staggered grid and the grid dimensions of the `u` and `v` grids are required.

39.3 User settings

User setting files (cfg files) are stored in `nml/cfg_SouthernOcean/`

1. SouthernOcean_polcon.ncl, SouthernOcean_polcon_diff.ncl, SouthernOcean_vector_polcon_diff.ncl
 - region: “Antarctic” (entire hemisphere will be evaluated)
 - month: “A” (A = annual mean, 3 = March, 9 = September)
 - styleset: “CMIP5”, “DEFAULT”
 - max_vert: max. number of rows on a panel page (vertical)
 - max_hori: max. number of columns on a panel page (horizontal)
 - grid_min: min. contour value (default = -1.0)
 - grid_max: max. contour value (default = 1.0)
 - grid_step: step between contours (default = 0.2)
 - colormap: color table (from NCL distribution)
 - PanelLabelBar: use single label bar per page (True, False)
 - showunits: display units in figure title (True, False)
 - range_option: 0 = use each model's time range specified in namelist, 1 = use only intersection of all time ranges

Setting specific to SouthernOcean_vector_polcon_diff.ncl

- vcmagnitude: magnitude of vectors (larger value = shorter arrows)
 - vcmindist: controls density of vectors (larger value = less arrows)
 - vccolormod: color of vectors for current model
 - vccolorref: color of vectors for reference model
2. SouthernOcean_areamean_vertconplot.ncl

Required diag_script_info attributes

- region: name of region (“Southern Ocean”)
- West: western boundary of area
- East: eastern boundary of area
- South: southern boundary of area
- North: northern boundary of area
- styleset: plot style (“CMIP5”, “DEFAULT”)

Optional diag_script_info attributes

- grid_min: min. for plot scale
- grid_max: max. for plot scale
- grid_step: step size for plot scale
- colormap: e.g., WhiteBlueGreenYellowRed, rainbow

- `dbg_plots`: create additional plots for debugging purposes (True, False)

3. SouthernOcean_transport.ncl

Required diag_script_info attributes

- `styleset`: plot style ("CMIP5", "DEFAULT")
- `lon1`: longitude of start of section
- `lat1`: latitude of start of section
- `lon2`: longitude of end of section
- `lat2`: latitude of end of section
- `section_name`: name used in title, e.g., "Drake passage"

39.4 Variables

- `hfds` (atmos, monthly mean, longitude latitude time) = `hfls` + `hfss` + `rsns` + `rlns`
- `mlotst` (ocean, monthly mean, longitude latitude time)
- `so` (ocean, monthly mean, longitude latitude olevel time)
- `sos` (ocean, monthly mean, longitude latitude time)
- `tauu` (atmos, monthly mean, longitude latitude time)
- `tauv` (atmos, monthly mean, longitude latitude time)
- `to` (ocean, monthly mean, longitude latitude olevel time)
- `tos` (ocean, monthly mean, longitude latitude time)
- `uo` (ocean, monthly mean, longitude latitude olevel time)
- `vo` (ocean, monthly mean, longitude latitude olevel time)
- `wfpe` (atmos, monthly mean, longitude latitude time) = `pr` + `evspsbl`

39.5 Observations and reformat scripts

Note: (1) `obs4mips` data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-`obs4mips` data for download instructions.

- Dong et al. (2008): Southern Ocean mixed-layer depth from Argo float profiles. Reformat script: `reformat_scripts/obs/reformat_obs_Dong08-ARGO-monthly.ncl`.
- ERA-Interim, global atmospheric ECMWF reanalysis: SST, wind stress, precipitation, evaporation, etc.

Reformat scripts:

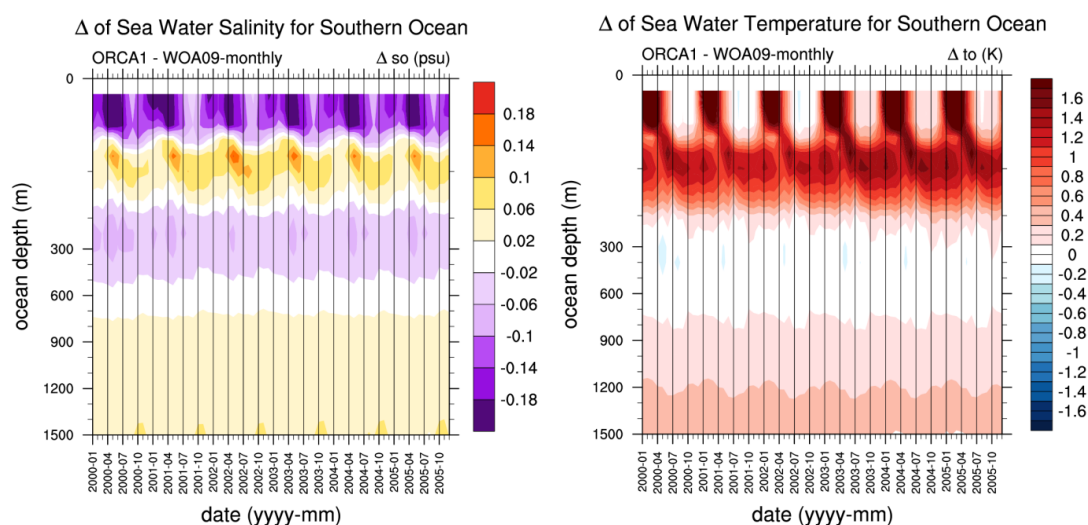
- variables `tos`, `tauu`, `tauv`: `reformat_scripts/obs/reformat_obs_ERA-Interim.ncl`
- variables `pr`, `evspsbl`, `hfls`, `hfss`, `rsns`, `rlns`: `reformat_scripts/obs/reformat_obs_ERA-Interim-surffluxes.ncl`
- World Ocean Atlas 2009: sea water temperature and salinity

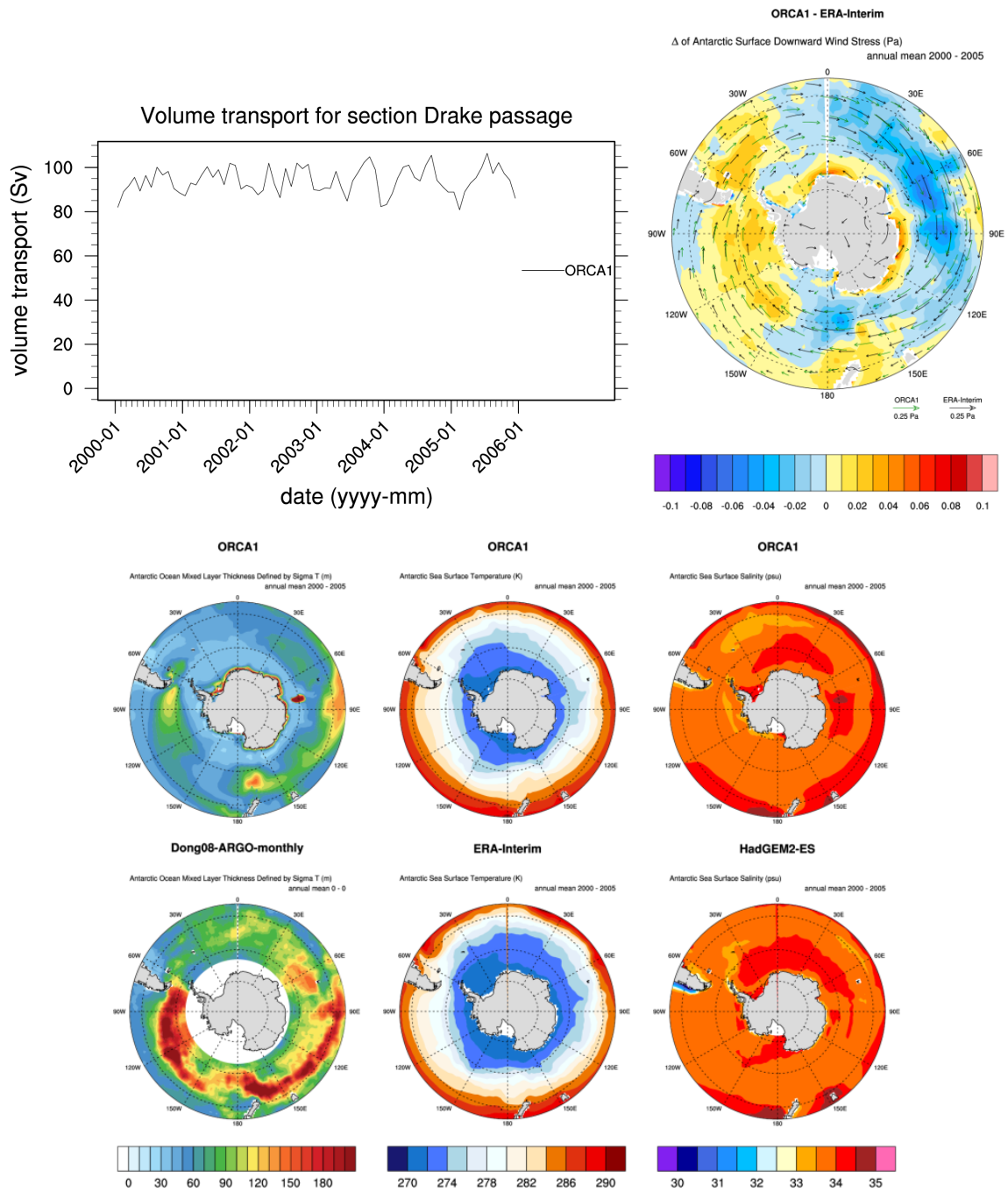
Reformat script: `reformat_scripts/obs/reformat_obs_WOA09.ncl`

39.6 References

- Antonov, J. I., D. Seidov, T. P. Boyer, R. A. Locarnini, A. V. Mishonov, H. E. Garcia, O. K. Baranova, M. M. Zweng, and D. R. Johnson (2010). World Ocean Atlas 2009, Volume 2: Salinity. S. Levitus, Ed. NOAA Atlas NESDIS 69, U.S. Government Printing Office, Washington, D.C., 184 pp (available at ftp://ftp.nodc.noaa.gov/pub/WOA09/DOC/woa09_vol2_text.pdf).
- CDFtools: <http://servforge.legi.grenoble-inp.fr/projects/CDFTOOLS>
- Dong, S., J. Sprintall, S. T. Gille, and L. Talley (2008). Southern Ocean mixed-layer depth from Argo float profiles, J. Geophys. Res., 113, C06013, doi: 10.1029/2006JC004051.
- ERA-Interim: <http://www.ecmwf.int/en/research/climate-reanalysis/era-interim>
- ESMF regridding with ncl: <http://www.ncl.ucar.edu/Applications/ESMF.shtml>
- Locarnini, R. A., A. V. Mishonov, J. I. Antonov, T. P. Boyer, H. E. Garcia, O. K. Baranova, M. M. Zweng, and D. R. Johnson (2010). World Ocean Atlas 2009, Volume 1: Temperature. S. Levitus, Ed. NOAA Atlas NESDIS 68, U.S. Government Printing Office, * Washington, D.C., 184 pp. (available at ftp://ftp.nodc.noaa.gov/pub/WOA09/DOC/woa09_vol1_text.pdf).
- nco: <http://nco.sourceforge.net>
- World Ocean Atlas 2009: https://www.nodc.noaa.gov/OC5/WOA09/pr_woa09.html

39.7 Example plots





Standardized Precipitation Index (SPI)

40.1 Overview

For each month, the precipitation over the preceding TIMESCALE months, x , is summed. Then a two-parameter Gamma distribution of cumulative probability, $\text{Gamma}_{\alpha,\beta}$, is fitted to the strictly positive TIMESCALE month sums, such that the probability of a non-zero precipitation sum being below a certain value x corresponds to $\text{Gamma}_{\alpha,\beta}(x)$. We estimate shape parameter α and scale parameter β with a maximum likelihood approach. If the estimation does not converge, α and β are approximated using empirical relations (Bordi et al., 2001). Accounting for TIMESCALE month periods of no precipitation, occurring at a frequency q , the total cumulative probability distribution of a precipitation sum below x , $H(x)$, becomes $H(x) = q + (1 - q) * \text{Gamma}_{\alpha,\beta}(x)$. In the last step, a precipitation sum x is assigned to its corresponding Standardized Precipitation Index (SPI) value by computing the quantile $q_{N(0,1)}$ of the standard normal distribution at probability $H(x)$. The SPI of a precipitation sum x , thus, corresponds to the quantile of the standard normal distribution which is assigned by preserving the probability of the original precipitation sum, $H(x)$.

40.2 Available namelists and diagnostics

Namelists are stored in `nml/`

- `namelist_SPI.xml`

Diagnostics are stored in `diag_scripts/`

- `SPI.r`

40.3 User settings

User setting files (cfg files) are stored in `nml/cfg_SPI/`

1. `SPR.r`
 - `begin.ref.year`: first year of the reference period
 - `end.ref.year`: last year of the reference period
 - `timescale`: valid values are 3, 6 and 12 months

- seasons: “ann”, “djf”, “mam”, “jja”, “son”
- spi_colorbar_max: color bar range (= -spi_colorbar_max ... +spi_colorbar_max)
- my.colors: colors for contour plot, e.g., colorRampPalette(c(“brown”, “orange”, “white”, “lightblue”, “blue”))
- png_width: width of png image
- png_height: height of png image
- png_units: units of png dimensions (“px” = pixels, “in” = inches, “cm” = centimeters, “mm” = millimeters)
- png_pointsize: the default size of plotted text in points (1/72 inch)
- png_bg: background color, e.g., “white”

40.4 Variables

- pr (atmos, monthly mean, longitude latitude time)

40.5 Observations and reformat scripts

Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

40.6 References

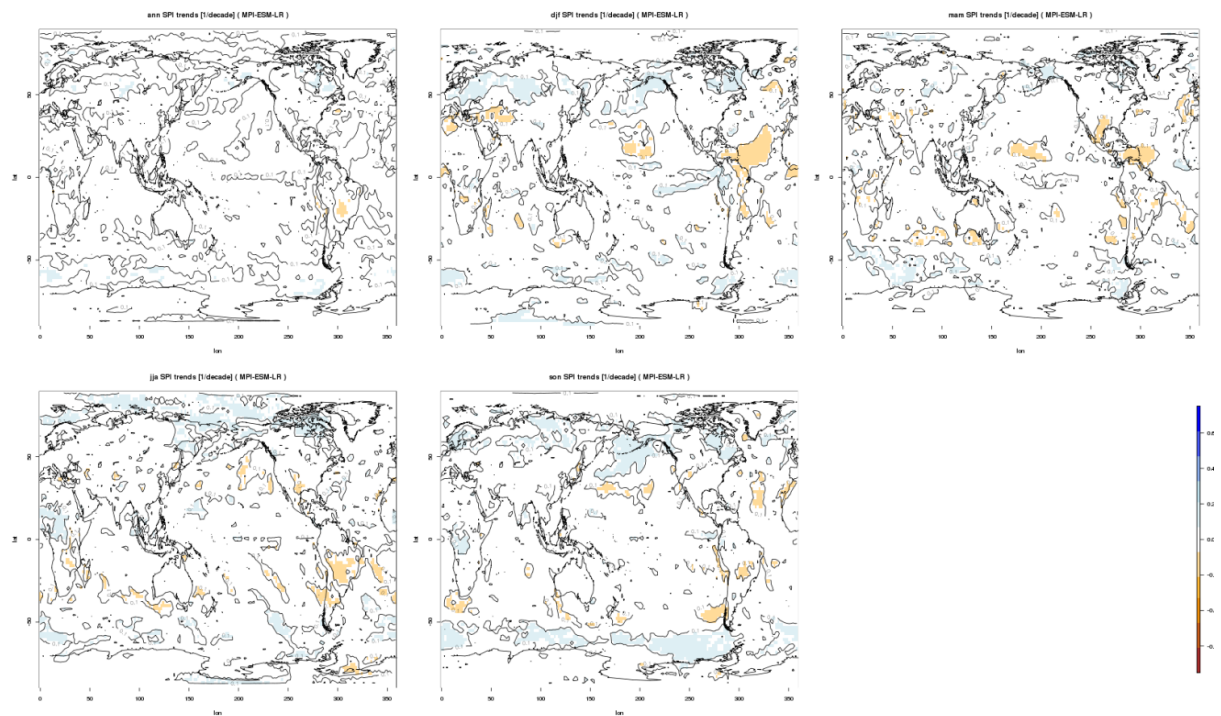
A very good explanation of SPI

- Lloyd-Hughes, B. and Saunders, M. A. (2002), A drought climatology for Europe. *Int. J. Climatol.*, 22, 1571-1592. doi: 10.1002/joc.846.

Other standard SPI references

- Guttman, N. B. (1999), ACCEPTING THE STANDARDIZED PRECIPITATION INDEX: A CALCULATION ALGORITHM. *JAWRA Journal of the American Water Resources Association*, 35, 311-322. doi: 10.1111/j.1752-1688.1999.tb03592.x.
- McKee, T. B., N. J. Doesken, and J. Kliest, 1993: The relationship of drought frequency and duration to time scales. In *Proceedings of the 8th Conference of Applied Climatology*, 17-22 January, Anaheim, CA. American Meteorological Society, Boston, MA. 179-184.
- McKee, T. B., N. J. Doesken, and J. Kliest, 1995: Drought Monitoring with Multiple Time Scales. 9th AMS Conference on Applied Climatology, 15-20 January 1995, Dallas, Texas.

40.7 Example plots



41.1 Overview

The available diagnostics are motivated by the work of Li and Xie (2014). In particular, this diagnostics reproduces their Fig. 5 for models and observations/reanalyses, calculating equatorial mean (5°N-5°S), longitudinal sections of annual mean precipitation (pr), skin temperature (ts), horizontal winds (ua and va) and 925 hPa divergence (derived from the sum of the partial derivatives of the wind components extracted at the 925 hPa pressure level (that is $du/dx + dv/dy$). Latitude cross sections of the model variables are plotted for the equatorial Pacific, Indian and Atlantic Oceans with observational constraints provided by the TRMM-3B43-v7 for precipitation, the HadISST for SSTs, and ERA-interim reanalysis for temperature and winds. Latitudinal sections of absolute and normalized annual mean SST and precipitation are also calculated spatially averaged for the three ocean basins. Normalization follows the procedure outlined in Fig. 1 of Li and Xie (2014) whereby values at each latitude are normalized by the tropical mean (20°N-20°S) value of the corresponding parameter (e.g., annual mean precipitation at a given location is divided by the 20°N-20°S annual mean value). Finally, to assess how models capture observed relationships between SST and precipitation the co-variability of precipitation against SST is calculated for specific regions of the tropical Pacific. This analysis includes calculation of the Mean Square Error (MSE) between model SST/precipitation and observational equivalents.

41.2 Available namelists and diagnostics

Namelists are stored in nml/

- namelist_TropicalVariability.xml

Diagnostics are stored in diag_scripts/

- TropicalVariability.py
- TropicalVariability_EQ.py
- TropicalVariability_wind.py

41.3 User settings

User setting files (cfg files) are stored in nml/cfg_Tropical Variability/

1. TropicalVariability.py, TropicalVariability_EQ.py, TropicalVariability_wind.py

Settings for all diagnostics

- plot_equatorial: switch for equatorial plots (True, False)
- plot_scatter: switch for scatter plots (True, False)
- plot_zonal_means: switch for zonal mean plots (True, False)
- plot_clouds: True, False
- plot_fluxes: True, False
- mask_unwanted_values: mask values (e.g., missing values) exceeding “mask_limit_low” and “mask_limit_high” (True, False)
- mask_limit_low: lower threshold used for creating a mask (if “mask_unwanted_values” = True)
- mask_limit_high: upper threshold used for creating a mask (if “mask_unwanted_values” = True)
- plot_grid: provides a background grid for relevant plots (True, False)

Settings for equatorial mean plots with precipitation, temperature and winds

- areas: region to process (“Atlantic”, “Indian”, “Pacific”); each region is defined separately

Definition of regions (“areas”)

- lat_min: min. latitude of region
- lat_max: max. latitude of region
- lon_min: min. longitude of region
- lon_max: max. longitude of region
- prec_min: range of values for precipitation (min.)
- prec_max: range of values for precipitation (max.)
- temp_min: range of values for temperature/SST (min.)
- temp_max: range of values for temperature/SST (max.)
- wind_min: range of values for wind speed (min.)
- wind_max: range of values for wind speed (max.)
- div_min: range of values for divergence (min.)
- div_max: range of values for divergence (max.)

Settings for temperature/precipitation scatter plots

- areas: region to process (“West-Pacific”, “Central-Pacific”, “East-Pacific”)
- seasons: season (annual DJF MAM JJA SON)
- seasonal_limits: if you want to use your own limits (True) or let the code decide (False - values based on observations)

Definition of regions (“areas”) for scatter plots

- lat_min: min. latitude of region
- lat_max: max. latitude of region
- lon_min: min. longitude of region
- lon_max: max. longitude of region
- season_limits_annual: “seasonal” limits (annual means), 4 integer numbers giving ‘min. temp.’ ‘max. temp.’ ‘min. precip.’ ‘max. precip.’, e.g., 300 303 4 10
- season_limits_DJF: same as season_limits_annual, but for Dec-Jan-Feb (DJF)

- season_limits_MAM: same as season_limits_annual, but for Mar-Apr-May (MAM)
- season_limits_JJA: same as season_limits_annual, but for Jun-Jul-Aug (JJA)
- season_limits_SON: same as season_limits_annual, but for Sep-Oct-Nov (SON)

Definition of the seasons (each season defined separately)

- season_months: numbers of the months covered by the corresponding season, e.g., “12 1 2” for the season “DJF”

Settings for zonal means of SST and precipitation

- areas: region to process (Pacific Atlantic Indian)

Definition of the regions (“areas”)

- lat_min: min. latitude of region
- lat_max: max. latitude of region
- lon_min: min. longitude of region
- lon_max: max. longitude of region

41.4 Variables

- ts: skin temperature (atmos, monthly mean, time latitude longitude)
- pr: precipitation (atmos, monthly mean, time latitude longitude)
- ua: u-wind (atmos, monthly mean, time plevel latitude longitude)
- va: v-wind (atmos, monthly mean, time plevel latitude longitude)

41.5 Observations and reformat scripts

Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

- HadISST: skin Temperature (ts) / sea surface temperature (SST)

Reformat script: reformat_scripts/obs/reformat_obs_HadISST.ncl

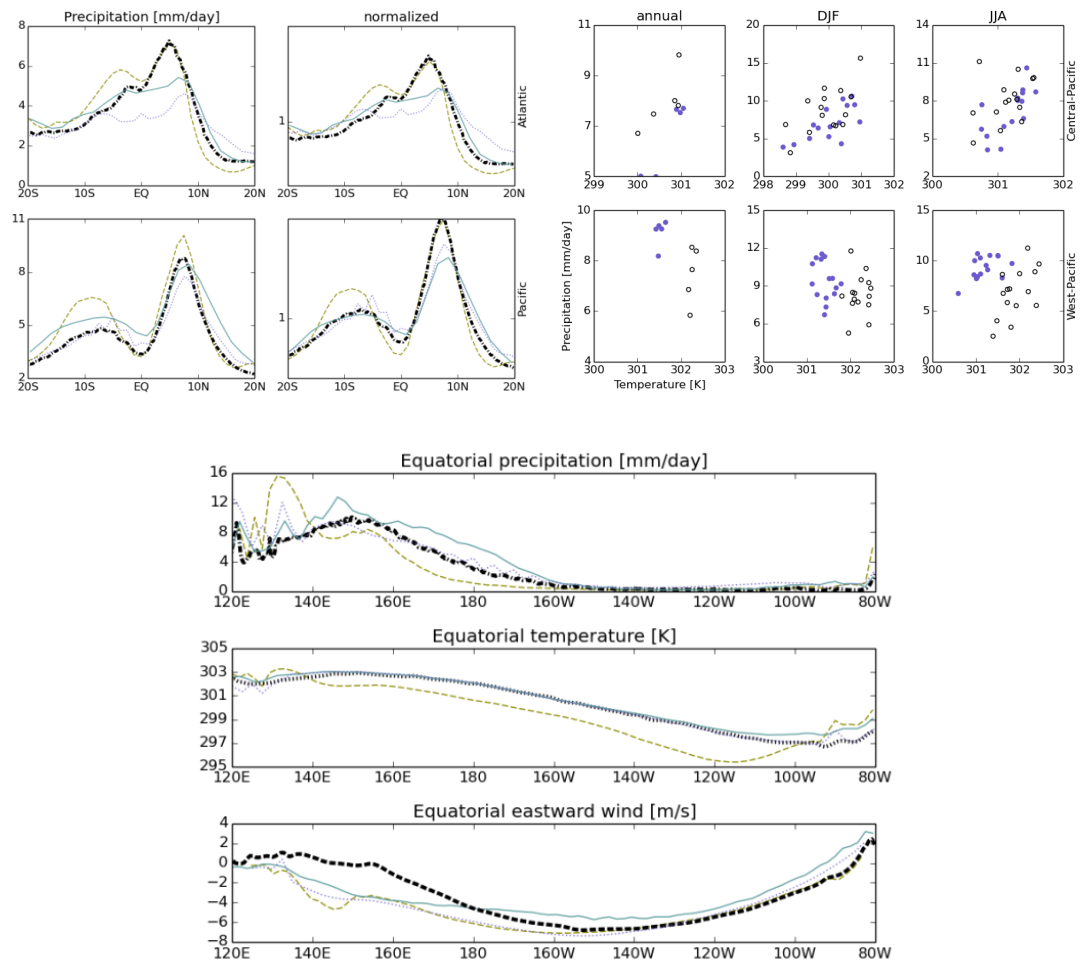
- TRMM-L3 (pr, monthly means - obs4mips)
- ERA-Interim (u-wind, v-wind)

Reformat script: reformat_scripts/obs/reformat_obs_ERA-Interim.ncl

41.6 References

- Li, G. and S.-P. Xie (2014), Tropical Biases in CMIP5 Multimodel Ensemble: The Excessive Equatorial Pacific Cold Tongue and Double ITCZ Problems. J. Climate, 27, 1765-1780. doi: <http://dx.doi.org/10.1175/JCLI-D-13-00337.1>.

41.7 Example plots



West African Monsoon (WAM)

42.1 Overview

West Africa is a critical region for climate models (e.g., Cook and Vizzy, 2006; Roehrig et al., 2013). Roehrig et al. (2013) show that although state-of-the-art CMIP5 models can capture many features of the West African monsoon, they have not yet reached a sufficient degree of maturity that makes them trustable to anticipate climate changes and their impacts in this region, especially with regard to rainfall. Therefore, along the process of climate model development and evaluation, it is crucial for model developers and users to have at their disposal a set of synthetic and simple diagnostics that provide them an overall vision of the representation of the West African monsoon in their model or set of models. Such diagnostics are implemented in this namelist.

42.2 Available namelists and diagnostics

Namelists are stored in `nml/`

- `namelist_WAMonsoon.xml`
- `namelist_WAMonsoon_daily.xml`

Diagnostics are stored in `diag_scripts/`

- `WAMonsoon_10W10E_1D_basic.ncl`: same as `WAMonsoon_10W10E_3D_basic.ncl` but for a 2D variable (e.g., precipitation, potential temperature at 850 hPa).
- `WAMonsoon_10W10E_3D_basic.ncl`: computes the zonal average over 10°W-10°E of a 3-dimensional variable (e.g., zonal wind, meridional wind, potential temperature). It is then averaged over the JJAS season and plotted as a latitude-level transect over West Africa.
- `WAMonsoon_autocorr.ncl`: similar to `WAMonsoon_isv_filtered.ncl`, except that it computes the 1-day autocorrelation of intraseasonal anomalies of any field (e.g., precipitation), as a measure of convection persistence.
- `WAMonsoon_contour_basic.ncl`: computes the average of any 2-dimensional field over the JJAS season and plots it as a latitude-longitude map zoomed over West Africa. It is used for precipitation, 2-m air temperature and 850-hPa potential temperature (heat low signature).
- `WAMonsoon_precip_IAV.ncl`: plots the interannual variability of precipitation averaged over JJAS and over the Sahel (10°N-20°N, 10°W-10°E).

- `WAMonsoon_isv_filtered.ncl`: filters any 2-dimensional field, computes the standard deviation of the filtered field over the JJAS season and plots it on a latitude-longitude map zoomed over West Africa. High-pass and band-pass filter based on the Lanczos filtering method are available. Basically, this is used for computing the 90-day high pass or 3-10-day bandpass filtered precipitation or outgoing longwave radiation standard deviation, namely the intraseasonal and synoptic (African easterly waves) standard deviation. Data are first interpolated on a common grid before any computations. We advise a $1^\circ \times 1^\circ$ grid for precipitation (reference is GPCP version 1.2, 1DD, available on this grid) and a $2.5^\circ \times 2.5^\circ$ grid for OLR (reference is from NOAA satellites, and available on this grid).
- `WAMonsoon_precip_seasonal.ncl`: computes the mean monthly annual cycle of any 2-dimensional variable averaged over a given latitude-longitude box and plots it for the given models and reference data set. It is used for precipitation and 2-m air temperature averaged over the Sahel (10°N - 20°N , 10°W - 10°E).
- `WAMonsoon_wind_basic.ncl`: computes the average of zonal and meridional wind component over the JJAS season and plots it as a latitude-longitude map for a given level (200 and 700 hPa). Zonal wind is in shading and total wind is in vector. The map is zoomed over West Africa.

42.3 User settings

User setting files (cfg files) are stored in `nml/cfg_WAMonsoon/`

1. `WAMonsoon_10W10E_1D_basic.ncl`

Required diag_script_info attributes

- `latrange`: min. and max. latitude for plot (2-element array)
- `lonrange`: min. and max. longitude for plot (2-element array)
- `minmax`: lower and upper limit of values in plot (2-element array)
- `season`: season, typically “JJAS”

Optional diag_script_info attributes

- `cn_levels_mean_basic`: contour levels for mean plot (n-element array)
- `cn_levels_mean_diff_basic`: contour levels for difference plot (n-element array)
- `cn_levels_stddev_basic`: contour levels for standard deviation plot (n-element array)
- `cn_levels_stddev_diff_basic`: contour levels for difference standard deviation plot (n-element array)
- `legendPos`: position of legend (“TopRight”, “BottomRight”, “TopLeft”, “BottomLeft”)
- `multi_model_mean`: calculate multi-model mean (“y”, “n”)
- `plotype_lbres`: plot labelbar (True, False)
- `stylesheet`: stylesheet to use (“CMIP5”, “DEFAULT”)
- `x_gridlines`: display gridline along x-axis (“y”, “n”)
- `y_gridlines`: display gridline along y-axis (“y”, “n”)

2. `WAMonsoon_10W10E_3D_basic.ncl`

Required diag_script_info attributes

- `cn_levels_mean_basic`: contour levels for mean plot (n-element array)
- `cn_levels_mean_diff_basic`: contour levels for difference plot (n-element array)
- `latrange`: min. and max. latitude for plot (2-element array)
- `lonrange`: min. and max. longitude for plot (2-element array)
- `levrange`: max. and min. pressure (Pa) for plot (2-element array)
- `plotype_lbres`: handle for labelbar settings (True, False)

- season: season, typically “JJAS”
- plot_stddev: switch for plotting standard deviation (True, False)
- diff_colormap: filename and full path for colormap for difference plot, e.g., “diag_scripts/aux/WAMonsoon/cmap_difference_theta.rgb”

Optional diag_script_info attributes

- cn_levels_mean_basic: contour levels for mean plot (n-element array)
- cn_levels_mean_diff_basic: contour levels for difference plot (n-element array)

3. WAMonsoon_autocorr.ncl

Required diag_script_info attributes

- cn_levels_corr_basic: contour levels for correlation plot (n-element array)
- cn_levels_corr_diff_basic: contour levels for correlation difference plot (n-element array)
- latrange: min. and max. latitude for plot (2-element array)
- lonrange: min. and max. longitude for plot (2-element array)
- sahel_latrange: min. and max. latitude of “Sahel” region (2-element array)
- sahel_lonrange: min. and max. longitude of “Sahel” region (2-element array)
- season: season, typically “JJAS”
- destgrid: destination grid for ESMF regridding (“1x1”, “2.5x2.5”)
- stylesheet: stylesheet to use (“CMIP5”, “DEFAULT”)
- my_region: label for Monsoon region, e.g., “WA”
- filter_hp: value for high pass filter
- filter_type: “hp” = high pass filter, “bp” = band pass filter
- filter_weights: filter weights
- filter_min, filter_max: required for band pass filter only

4. WAMonsoon_contour_basic.ncl

Required diag_script_info attributes

- cn_levels_mean_basic: contour levels for mean plot (n-element array)
- cn_levels_mean_diff_basic: contour levels for difference plot (n-element array)
- latrange: min. and max. latitude for plot (2-element array)
- lonrange: min. and max. longitude for plot (2-element array)
- season: season, typically “JJAS”
- my_region: label for Monsoon region, e.g., “WA”
- plot_stddev: switch for plotting standard deviation (True, False)
- cn_levels_stddev_basic: contour levels for standard deviation plot (only required if plot_stddev = True) (n-element array)
- cn_levels_stddev_diff_basic: contour levels for standard deviation difference plot (only required if plot_stddev = True) (n-element array)
- diff_colormap: filename and full pathname of colormap for difference plots, e.g., “diag_scripts/aux/WAMonsoon/cmap_difference.rgb”

5. WAMonsoon_precip_IAV.ncl, WAMonsoon_precip_seasonal.ncl

Required diag_script_info attributes

- `cn_levels_mean_basic`: contour levels for mean plot (n-element array)
- `cn_levels_mean_diff_basic`: contour levels for difference plot (n-element array)
- `cn_levels_stddev_basic`: contour levels for standard deviation plot (n-element array)
- `cn_levels_stddev_diff_basic`: contour levels for difference standard deviation plot (n-element array)
- `diff_colormap`: filename and full pathname of colormap for difference plots, e.g., “diag_scripts/aux/WAMonsoon/cmap_difference.rgb”
- `latrange_seasonal`: min. and max. latitude of crop region for plot (2-element array)
- `lonrange_seasonal`: min. and max. longitude of crop region for plot (2-element array)
- `season`: season, typically “JJAS”
- `stylesheet`: stylesheet to use (“CMIP5”, “DEFAULT”)
- `multi_model_mean`: calculate multi-model mean (“y”, “n”)
- `m_y_region`: label for Monsoon region, e.g., “WA”
- `supporting_gridlines`: display supporting grid line (“y”, “n”)

6. WAMonsoon_isv_filtered.ncl

Required diag_script_info attributes

- `season`: season, typically “JJAS”
- `destgrid`: destination grid for ESMF regridding (“1x1”, “2.5x2.5”)
- `stylesheet`: stylesheet to use (“CMIP5”, “DEFAULT”)
- `latrange_basic`: min. and max. latitude for plot (2-element array)
- `lonrange_basic`: min. and max. longitude for plot (2-element array)
- `diff_colormap`: filename and full pathname of colormap for difference plots, e.g., “diag_scripts/aux/WAMonsoon/cmap_difference.rgb”
- `cn_levels_stddev_basic`: contour levels for standard deviation plot (n-element array)
- `cn_levels_stddev_diff_basic`: contour levels for difference standard deviation plot (n-element array)
- `plot_norm`: plot normalized standard deviation (True, False)
- `cn_levels_stddev_norm_basic`: contour levels for normalized standard deviation plot (n-element array)
- `cn_levels_stddev_norm_diff_basic`: contour levels for normalized standard deviation difference plot (n-element array)
- `sahel_latrange`: min. and max. latitude of “Sahel” region (2-element array)
- `sahel_lonrange`: min. and max. longitude of “Sahel” region (2-element array)
- `filter_type`: “hp” = high pass filter, “bp” = band pass filter
- `filter_hp`: value for high pass filter
- `filter_weights`: filter weights
- `filter_min`, `filter_max`: required for band pass filter only
- `my_region`: label for Monsoon region, e.g., “WA”

7. WAMonsoon_wind_basic.ncl

Required diag_script_info attributes

- `stylesheet`: stylesheet to use (“CMIP5”, “DEFAULT”)
- `season`: season, typically “JJAS”
- `latrange_basic`: min. and max. latitude for plot (2-element array)

- lonrange_basic: min. and max. longitude for plot (2-element array)
- cn_levels_mean_basic: contour levels for mean plot (n-element array)
- cn_levels_mean_diff_basic: contour levels for difference plot (n-element array)
- diff_colormap: filename and full pathname of colormap for difference plots, e.g., “diag_scripts/aux/WAMonsoon/cmap_difference_wind.rgb”
- cn_levels_stddev_basic: contour levels for standard deviation plot (n-element array) (if plot_stddev = True)
- cn_levels_stddev_diff_basic: contour levels for difference standard deviation plot (n-element array) (if plot_stddev = True)
- plottype_lbres: plot labelbar (True, False)
- my_region: label for Monsoon region, e.g., “WA”
- use_for_contour: switch for kind of contour (“speed”, “zonal” (ua), “meridional” (va))
- plot_stddev: plot standard deviation (True, False)

42.4 Variables

- pr (atmos, monthly mean, longitude latitude time)
- tas (atmos, monthly mean, longitude latitude time)
- rlut (atmos, monthly mean, longitude latitude time)
- rsut (atmos, monthly mean, longitude latitude time)
- rlutcs (atmos, monthly mean, longitude latitude time)
- rsutcs (atmos, monthly mean, longitude latitude time)
- rlds (atmos, monthly mean, longitude latitude time)
- rsds (atmos, monthly mean, longitude latitude time)
- ua (atmos, monthly mean, longitude latitude plev time)
- va (atmos, monthly mean, longitude latitude plev time)
- ta (atmos, monthly mean, longitude latitude plev time)
- pr (atmos, daily mean, longitude latitude time)
- rlut (atmos, daily mean, longitude latitude time)

42.5 Observations and Reformat Scripts

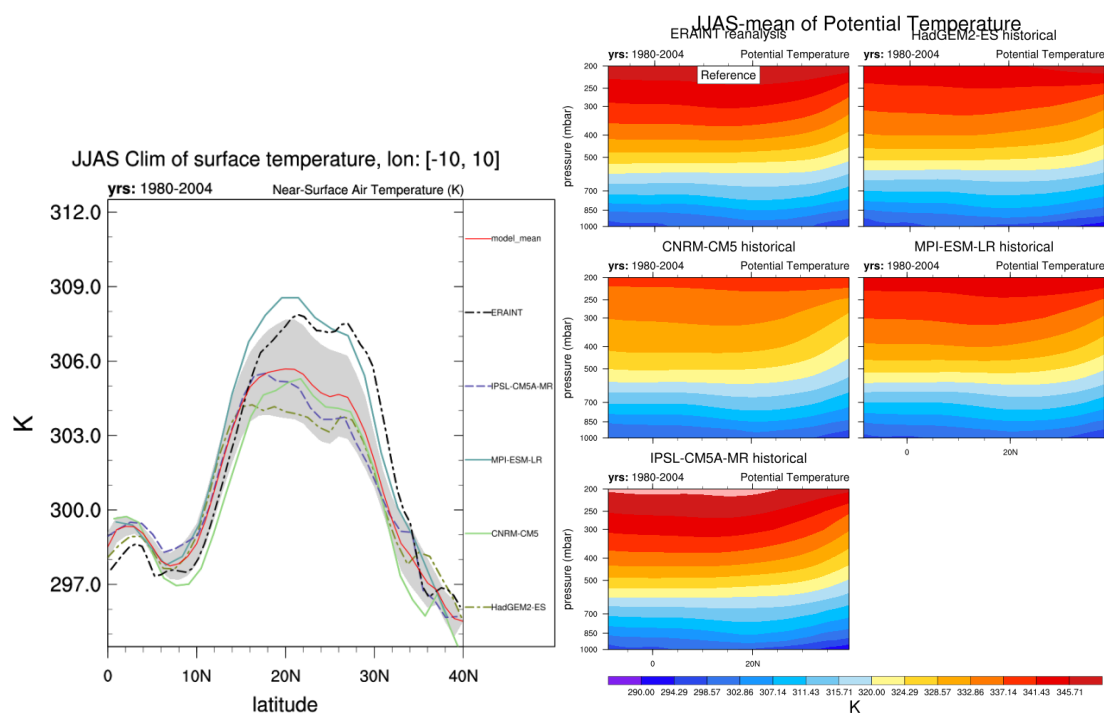
Note: (1) obs4mips data can be used directly without any preprocessing; (2) see headers of reformat scripts for non-obs4mips data for download instructions.

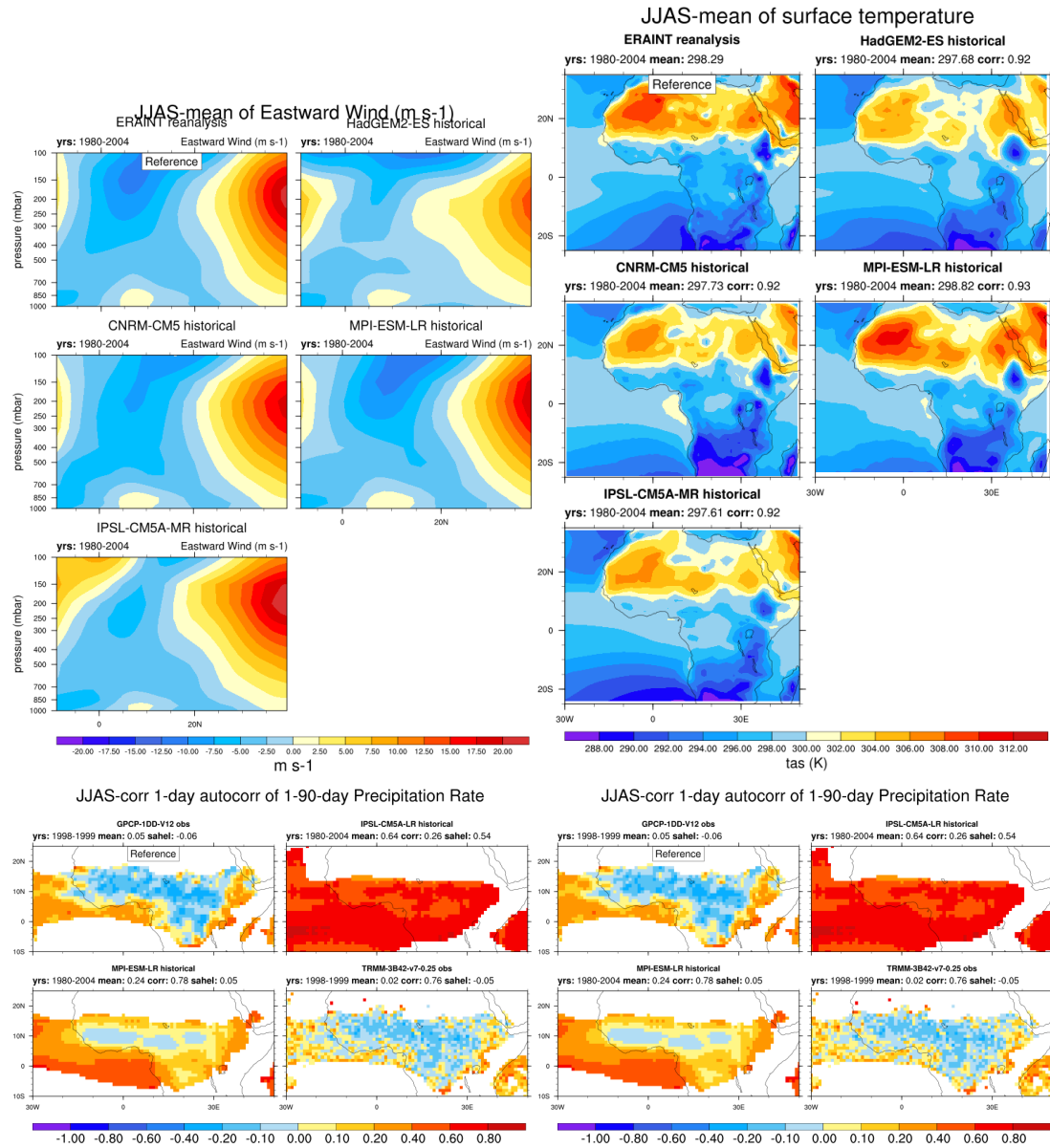
- ERA-Interim Reanalysis (tas, ua, va)
Reformat script: reformat_scripts/obs/reformat_obs_ERA-Interim.ncl
- GPCP monthly (pr) – obs4mips
- CERES-EBAF (TOA and derived surface radiation fluxes) – obs4mips
- GPCP Version 1.2, daily and 1°x1° (pr) – obs4mips
- Daily NOAA OLR
Reformat script: reformat_scripts/obs/reformat_obs_NOAA-PSD-Interp.ncl

42.6 References

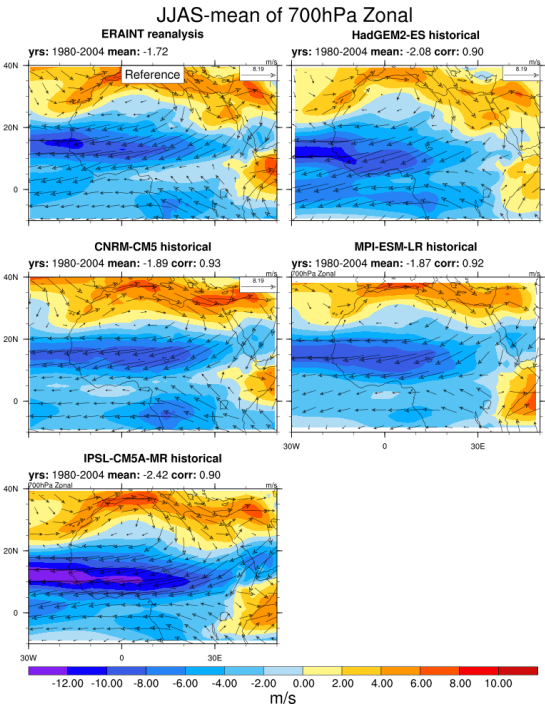
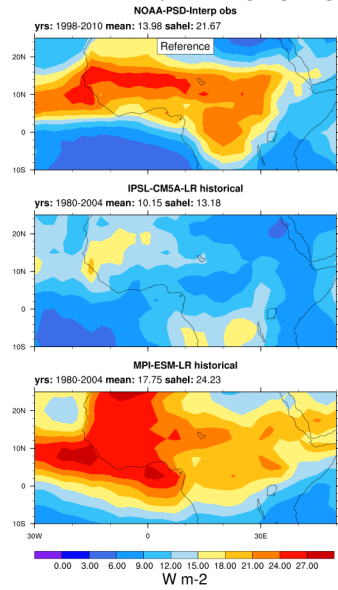
- Cook, K. H. and E. K. Vizi, 2006: Coupled model simulations of the West African monsoon system: Twentieth- and twenty-first-century simulations. *J. Climate*, 19, 3681-3703.
- Roehrig, R., D. Bouniol, F. Guichard, F. Hourdin, and J.-L. Redelsperger, 2013: The Present and Future of the West African Monsoon: A Process-Oriented Assessment of CMIP5 Simulations along the AMMA Transect. *J. Climate*, 26, 6471-6505. doi: <http://dx.doi.org/10.1175/JCLI-D-12-00505.1>.

42.7 Example plots

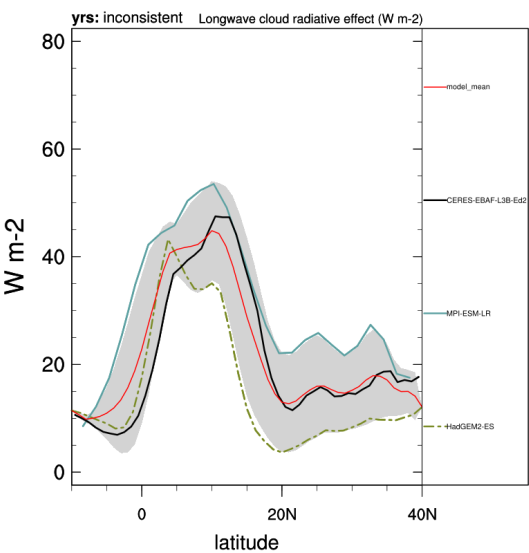




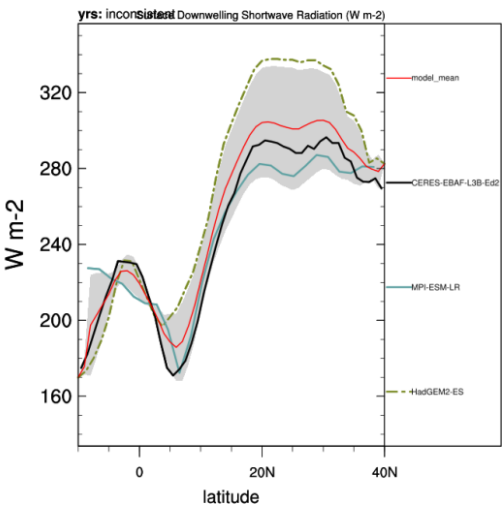
JJAS-stddev stddev of 3-10-day TOA Outgoing Longwave Radiation



JJAS Clim of Longwave cloud radiative effect, lon: [-10, 10]



JJAS Clim of Surface Downwelling Shortwave Radiation, lon: [-10, 10]



CHAPTER 43

Indices and tables

- `genindex`
- `search`