Demonstration and Practical Work

1. Setting up ESMValTool
2. ESMValTool structure
3. Workflow
4. Namelist structure
5. Observations
6. Demonstration
7. Run your first ‘hello world’ diagnostic
8. Example diagnostics:

- - SeaIce
  -Adding new model into diagnostic
- - CVDP
- - Clouds etc…

1. Setting up ESMValTool

Lets install your instance of ESMValTool

Go to your home folder on ciclad:

```
cd /home/tp/tp##
cp /home/nkadygrov/get_ESMVal_tool.sh .
./get_ESMVal_tool.sh ESMValTool_form
```

(You can select any name, not only ESMValTool_form)

This script will copy ESMValTool to your home folder and creates working dir with working files for diagnostics:

```
/data/tp/tp##/work_form/...
```

These files were created during the diagnostic run, but it takes time, so they are precalculated for the training (not results)

Also, the user specific configuration file will be prepared:

```
~/ESMValTool_form/config_private_IPSL_form.xml
```

You may need to load NCL module version 6.3.0

Check which NCL version is in use and load the module:

```
ncl -V
module load ncl/6.3.0
```

2. ESMValTool structure

The structure of ESMValTool is as follows:

```
ls ~/ESMValTool_form/
```

`diag_scripts/` – main folder with the code of the diagnostics

`reformat_scripts/` – reformat routines used during the run and also to reformat new observations (with detailed description how to get the data)

`variable_defs/` – description of the variables. NCL scripts which define the variables with units conversion if needed. Also any new variables have to be added and described there.

`interface_data/` – Intermediate files, temporal, based on templates files for different language updated on-the-fly. The main reason, why you can't run several diagnostics in the same time... ncl.interface – main file used by all the diagnostics

`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)

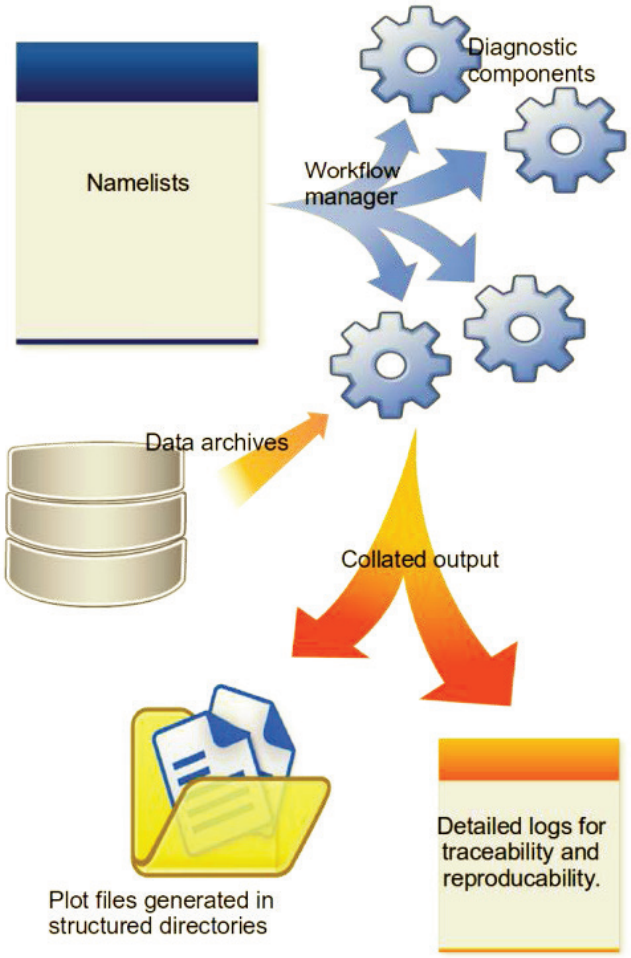
`nml/` – Namelists for specifying general parameters, input data and diagnostics to run. Here all the diagnostics are stored

`plot_scripts/` – plotting routines

3. Workflow

Workflow Structure and namelists

The workflow manager: *Python* script
`~/ESMValTool_form/main.py`
runs a set of diagnostics on data provided by, for instance, a data archive
`/prodigfs/projects/CMIP5/…`
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



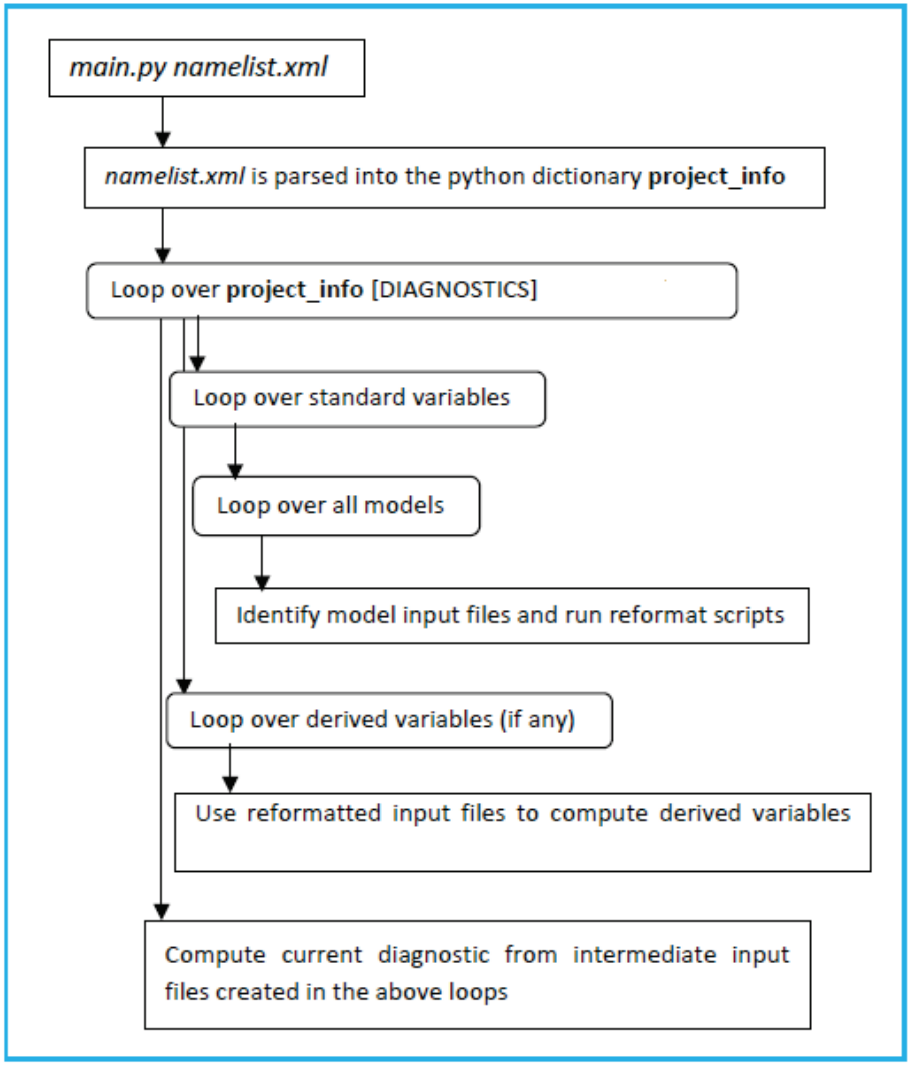
Namelists acting as interfaces between the user and the scripts and configuration files. **Namelist specifies a list of diagnostics to run, global flags and a list of models and observations that are used within the diagnostics**. Namelists are text files written in **XML** and can be modified by the user. For any given namelist "*namelist_###.xml*", the tool is invoked from the command line, from the ESMValTool root folder, via:

```
python main.py nml/namelist_###.xml
```

Workflow

The Python “workflow manager” ***main.py*** will parse the namelist (***namelist_##.xml***) and **call all diagnostic scripts listed in the namelist**.

This sequence 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



4. Namelist structure

Open namelist _perfmetrics_IPSL_form.xml in `~/ESMValTool_form/nml/`

Download the detailed manual:
https://github.com/ESMValGroup/ESMValTool/raw/master/doc/ESMValTool_Users_Guide.pdf

Basic structure of a namelist:

config xml file *(optional – paths settings, now your* `/data/tp/tp##` *accounts)*

<GLOBAL>
controls the general settings
</GLOBAL>

<MODELS>
defines the models/observations and years to be processed and their pathnames
</MODELS>

<DIAGNOSTIC>
defines which diagnostics are run. Each diagnostic is enclosed in an opening <diag> and closing </diag>-tag
</DIAGNOSTIC>

Namelist structure

More on <GLOBAL> tag:

Some of these tags in table (e.g., regridding_dir) are specific to some diagnostics and not generally defined in all namelists.

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

Namelist Structure

More on <MODELS> tag:

Each data set is specified by a <model> line with the first entry of each model line being the "**project specifier**" . 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_WPSIPSL 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> e.g.:

*<variable **MIP**="Omon"> fgco2 </variable>*
*<model> **CMIP5_WPSIPSL** MPI-ESM-LR **MIP_VAR_DEF** esmHistorical r1i1p1 1960 2005 @{MODELPATH} </model>*

The project specifier "CMIP5_WPSIPSL" 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>, the trailing asterisk is a placeholder for the start/end date of the data set.

Namelist Structure

More on <MODELS> tag:

… 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.

For IPSL the new class **CMIP5_WPSIPSL** was created to follow the IPSL DRS structure **starting** from the **path.** Otherwise (if using CMIP5) workflow will search for files directly **in path**. The full list of project specifiers and corresponding arguments are given in user guide.

The <model>-tag may also take the optional attribute "*id*":
<model id="string">

Example:
*<model **id**="obs"> OBS ERA-Interim reanaly 1 1998 2004 @{OBSPATH}/ERA-Interim/ESMVal_ERA-Interim </model>*
The attribute *id* specifies a string that can be used to refer to the model in other places of the namelist.

Namelist Structure

More on the <DIAGNOSTICS>-tag

Each <diag> entry refers to one or several scripts in the folder **diag_scripts/** complemented by a variable name and the corresponding (input) field type

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).

All variables may be found in: `variable_defs/` folder

Namelist Structure

More on the <DIAGNOSTICS>-tag

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 including possible preprocessing (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 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 but will be ignored by other <diag>-sections.

Namelist Structure

More on the <DIAGNOSTICS>-tag

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

Namelist Structure

More on the <DIAGNOSTICS>-tag

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)

5. Observations

Most of the diagnostics in ESMValTool works with CMIP5 data, located in `/prodigfs/projects/CMIP5/…` and available through CMIP5_WPSIPSL project specifier.

For observations, the main specifiers are: **OBS**, **obs4mips** and **ana4mips**.

The set of observations utilized by ESMValTool is huge including satellites and campaign data (see Table S9 of the user guide)

For most of the observations there are reformatting scripts with detailed instructions how to download and process data. The reformat routines are located in **reformat_scripts/obs/** folder. Some scripts used at IPSL are located in **/reformat_scripts_ipsl/** folder.

The root folder for the observations prepared for ESMValTool at IPSL is:

/prodigfs/ESMValData/OBS/

Typically, the naming convention of the subfolders is ESMVal_###, but this is not obligatory

Available observations

Name	Description	Vars	RES, lat, lon, lev	Period
AERONET	Aerosol optical depth at 550 nm	od550aer	T0M	1992-2016
AURA-MLS-OMI	Tropospheric column ozone	tropoz	T2Ms, 120x288	2005-2015
AURA-TES	Ozone mixing ratio	tro3	T3M, 83x90x15	2005-2009
CASTNET	Aerosol surface level concentrations	sconccl	T0M	1987-2016
		sconcna		
		sconcnh4		
		sconcno3		
		sconcso4		
CMAP	Precipitation	pr	T2Ms, 72x144	1979-2015
	CPC Merged Analysis of Precipitation			
	(excludes NCEP Reanalysis)			
CRU	ECV	pr, tas	T2Ms, 360x720	1901-2012
Dong08-ARGO-monthly	Derived ocean mixed layer depth	mlotst	TO2Ms	Climo, 35x360
EMEP	Aerosol surface level concentrations	concso4	T0M	1970-2016
		concno3		
		concnh4		
		concpm2p5		
		concpm10		
ESACCI-AEROSOL	Aerosol optical depth at 550 nm	od550aer	T2Ms, 180x360	1997-2011
ESACCI-CLOUD	Total cloud fraction	clt	T2Ms, 360x720	
	Liquid water path	clwvi		
	Ice water path	clivi		
ESACCI-OZONE	Total column ozone	toz	T2Ms, 180x360	1997-2010
ESACCI-SIC	Sea ice concentration, NH, SH	sic	T2Ms, 432x432	2003-2010
ESACCI-SIC	Sea ice concentration, NH, SH	sic	T2Ms, 432x432	1992-2008
ESACCI-SOILMOISTURE	Degree of saturation	dos, dosStderr	T2Ms, 720x1440	1979-2014
	Volumetric Moisture in Upper Portion of Soil Column	sm, smStderr		
ESACCI-SST	Surface Temperature	ts,tsStderr	T2Ms, 3600x7200	1992-2010
			T2Ms, 360x720	
ESRL	CO2 surface level concentrations	co2	T0M	1973-2015
ETH-SOM-FFN	sea surface pCO2	spco2	T02Ms, 180x360	1998-2011
GCP	Land plus Ocean Carbon Fluxes (nbp+fgco2)	co2flux	T0M	1959-2014
GLOBALVIEW	CO Volume Mixing Ratio	vmrco	T0Ms	1991-2008
GPCC	Precipitation	pr	T2Ms, 72x144	1901-2013
			T2Ms, 360x720	

Available observations

Name	Description	Vars	RES, lat, lon, lev	Period
GPCP_daily	Precipitation	pr	T2Ds, 180x360	199610-201510
GPCP	Precipitation	pr+error	T2Ms, 120x288	1979-2013
GPP	gross_primary_productivity_of_carbon	gpp	T2Ms, 360x720	1982-2011
HadCRUT	Near-Surface Air Temperature, var adjusted	tas	T2Ms, 36x72	1850-2013
	temperature anomalies with respect to 1961-1990			
HadCRUT4	Near-Surface Air Temperature, var adjusted	tas	T2Ms, 36x72	1850-2015
	Absolute values!			
HadISST	Sea ice concentrations	sic	T2Ms, 180x360	187001-201507
	Sea surface temperature	ts		
HALOE	Water vapour mixing ratio	vmrh2o	T3M, 34x50x22	1991-2002
IMPROVE	Aerosol surface level concentrations	concso4	T0M	1988-2015
		concno3		
		concnh4		
		concbc		
		concoa		
		concpm2p5		
		concpm10		
LAI3g	Leaf area index	lai	T2Ms, 360x720	1982-2011
LandFlux-EVAL	Evapotranspiration	et, et-sd	T2Ms, 180x360	1989-2005
MERRA	Precipitation	pr	T2Ms, 361x540	197901-201310
MERRA2	Precipitation	pr	T2Ds, 361x576	198001-201608
			T2Ms, 361x576	
			2DHourly, 361x576	
MODIS-L3-C6	Total cloud fraction	clt	T2Ms,180x360	200207-201609
	Liquid water path	clwvi		
	Ice water path	clivi		
	Aerosol optical depth	od550aer		
		iwpStderr		
		lwpStderr		
MODIS-obs4mips	Total cloud fraction	clt	T2Ms, 180x360	200003-201109
MODIS-CFMIP	Ice water path	clivi	T2Ms, 180x360	2003-2014
NOAA-PSD-Interp	TOA Outgoing Longwave Radiation	rlut	T2Ds, 73x144	1975-2013
NCEP	Essential climate variables	ECVs…	73x144x17	1948-2012
NSIDC	Sea Ice Concentration BT and NT satellite data	sic	T2Ms	197811-201412
SRB	Radiative fluxes	rsut	T2Ms, 180x360	1984-2007
		rlut		
		rlutcs		

Available observations

Name	Description	Vars	RES, lat, lon, lev	Period
Tilmes	Ozone mixing ratios	tro3	T1M	Climo?, 26 lev
TRMM_L3	Precipitation, 3B42 Version 7 satellite data	pr	T2Ds, 400x1440	199801-201607
			T2Ds, 400x1440	
			T2Is, 400x1440	
			T2Ms, 400x1440	
			T2Ms, 400x1440	
WHOI-AOFlux	Surface latent heat flux	hfls	T2Ms, 180x360	1958-2015
	Surface sensible latent heat flux	hfss		
WOA09-monthly	Sea water temperature and salinity data	so	T3M, 180x360x24	Climo
		sos	TO2Ms, 180x360	
		to	T3M, 180x360x24	
		tos	TO2Ms, 180x360	
ERA-Interim		psl	T2Ms, 241x480	1979-2012
		tas		
		hus	T3M, 241x480	
		ta		
		ua		
		va		
		zg		
		psl	T2Ms, 241x480	1979-2013
CERES	Surface Radiation Fluxes	rsuscs	3hr, 180x360	201401-201501
		rsus		
		rsdscs		
		rsds		
		rluscs		
		rlus		
		rldscs		
		rlds		
	Top of the Atmosphere Radiation Fluxes	rsutcs		
		rsut		
		rluts		
		rlut		
			Monthly, 180x360	200003-201302
				200003-201209
AIRS	Specific Humidity	hus	Monthly, 180x360	200209-201105
		ta		

Demonstration

Performance metrics for essential climate parameters

Open namelist _perfmetrics_IPSL_form.xml in `~/ESMValTool_form/nml/`

And follow the presentation…

Run your first diagnostic

For the very first run of the ESMVal tool we will use the toy diagnostic MyDiag

Go to your ESMValTool_form folder:

```
cd ~/ESMValTool_form
```

Open nml/namelist_MyDiag_IPSL_form.xml:

```
vim nml/namelist_MyDiag_IPSL_form.xml
```

Set output file type (ps, png, eps)

Check the configuration of the diagnostics:

```
vim nml/cfg_MyDiag/cfg_MyDiag.ncl
```

Check which variable to be used:

```
vim variable_defs/MyVar.ncl
```

And run:

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

To exit vim without saving:

```
:q! enter
```

With saving:

```
:wq enter
```

Run your first diagnostic

Now you can check the results:

Resulting plot will be in your <plot_dir> from <GLOBAL> section:
`/data/tp/tp##/work_form/MyDiag/plots/MyDiag`

The produced files will be in your <wrk_dir>:
`/data/tp/tp##/work_form/MyDiag/`

Now you can change some options, say projection in nml/cfg_MyDiag.ncl

Or, select NCEP data (also with MyVar variable, which is 'ta') and rerun tool (comment all the models but **NCEP**, note that the project is **OBS**)

Or, to see, how to add variables with units conversion:
Select **TRMM** data
(comment all the models but TRMM)
Select **Mypr** variable (comment MyVar)
And run.

Check the variable_defs/Mypr.ncl
Check the /data/tp/tp##/work_form/MyDiag/climo folder.

8. Diagnostics for training

8.1 Sealce Diagnostics (see the description in UserGuide, page 164)
Open namelist_Sealce_IPSL_form.xml in `~/ESMValTool_form/nml/`

Check the namelist blocks: diagnostics, configurations, models and observations (Now it's only one model there to save time).

Look at: `/nml/cfg_SeaIce/cfg_SeaIce_NH.ncl`

Note that reference obs could be defined directly in cfg, not in namelist.

To run:

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

Look at the output plots in your `/data/tp/tp##/work_form/SeaIce/..` folder

8. Diagnostics for training

8.1 Sealce Diagnostics (see the description in UserGuide, page 164)
Now, lets add new model, NOT from CMIP5 archive, to diagnostic. Say, we have results from intermediate version of the new IPSLCM6 model, called CM605-LR-piCtrl-01:

We created new project specifier, class **IPSLCM6**, with the same filename convention as CMIP5, but DRS is: /path_to_the_model/MODELNAME/

In our case it is:
`/data/tp/tp##/work_form/ADDNEWMODEL/IPSLCM6/CM605-LR-piCtrl-01/`

Original file from the model is (for sea ice conc):
`CM605-LR-piCtrl-01_19500101_23391231_1M_sicf.nc`

If needed, units were fixed with ncatted (check global attribute 'history') :
`ncdump -h CM605-LR-piCtrl-01_19500101_23391231_1M_sicf.nc`

And created the link to this file follow filename convention:
`sic_Amon_CM605-LR-piCtrl-01_piControl_typdef_195001_233912.nc`

8. Diagnostics for training

8.1 Sealce Diagnostics (see the description in UserGuide, page 164)

Note, that the variable is not sic, but sicf.
You need to add 'sicf' as altname for sic in:
reformat_scripts/recognized_vars.dat

std_name = sic
alt_name = sicf,siconc

Also, the units may differ. For example, in the same place we have:
pr_Amon_CM605-LR-piCtrl-01_piControl_typdef_195001_233912.nc
With the units "kg/(s*m2)" instead of "kg m-2 s-1" (and the name 'precip' for 'pr' variable, also fixed in reformat_script/recognized_vars)

The alt_unit should be added to standard unit, with scaling factor (check the `reformat_scripts/recognized_units.dat`):

std_unit = kg m-2 s-1
alt_unit = kg /(m^2 s),1, …, kg/m2/s,1, kg/(s*m2),1

8. Diagnostics for training

8.1 Sealce Diagnostics (see the description in UserGuide, page 164)

After that, we can use the new model in ESMValTool diagnostic

Uncomment this line, for IPSLCM6 project in
namelist_SeaIce_IPSL_form.xml :

```
<model>  IPSLCM6       CM605-LR-piCtrl-01  Amon  piControl
typdef  1980 2005 @{WORKPATH}/work_form/ADDNEWMODEL/IPSLCM6/
</model>
```

We will run this diagnostic with qsub.
For that, check the run_ESMValTool_SeaIce.sh script in ESMValTool_form folder

To run the script, we will run qsub, with more requested memory. This is not necessary for this diagnostic, but it is good to know:

qsub -l mem=8gb -l vmem=16gb run_ESMValTool_SeaIce.sh

To check the job status use: qstat –u $USER

8. Diagnostics for training

8.1 Other diagnostics to test:

Highly recommended:
-NCAR's Climate Variability Diagnostics Package (CVDP) :
namelist_CVDP_IPSL_form.xml
After run, check open index.html file (in outputs) with nice results:

```
firefox index.html
```

If your diagnostic (with qsub or other) is still running, I suggest to install second instance of the ESMValTool (it will not copy precalculated files anymore, so it's fast). The running time of CVDP is near 20 minutes.

From your home folder:

```
./get_ESMVal_tool.sh ESMValTool_cvdp
cd ESMValTool_cvdp
python main.py nml/namelist_CVDP_IPSL_form
```

8. Diagnostics for training

8.1 Other diagnostics to run:

namelist_CLOUDS_IPCC_form.xml

namelist_aerosol_CMIP5_IPSL_form.xml

namelist_Evapotranspiration_IPSL_form.xml

namelist_TropicalVariability_IPSL_form.xml

namelist_wenzel14jgr_IPSL_form.xml

If the NCL syntax is not highlighted in vim :
Copy `ncl.vim` file from `/home/nkadygrov/.vim/syntax/`
to your `~/.vim/syntax/` directory
Copy .vimrc file from /home/nkadygrov/ to you home folder
If things do not appear to be working, then add a third line to the `.vimrc` file:
`syntax on`

After re-login
And don't forget to load ncl/6.3.0 module again!