IS-ENES3 Deliverable D9.4

Seamless Evaluation with the ESMValTool

ESMValTool version supporting regional climate models and different timescales

Reporting period: 01/07/2020 – 31/12/2021

Authors: Saskia Loosveldt Tomas (BSC), Javier Vegas Regidor (BSC), Kim Serradell (BSC), Francesco Repola (CMCC), Veronica Villani (CMCC), Paola Mercogliano (CMCC), Pasquale Schiano (CMCC), Stéphane Sénesi (CNRS-IPSL), Klaus Zimmermann (SMHI)

Reviewers: Rémi Kazeroni (DLR), Sylvie Joussaume (CNRS-IPSL)

Release date: 16/12/2021

ABSTRACT

This deliverable describes the improvements implemented in the ESMValTool in order to achieve a seamless evaluation of climate datasets. The aim of the work consisted on expanding the ESMValTool from a framework designed to analyse global scale projections, to also support regional climate models participating in CORDEX and decadal experiments in the CMIP6 DCPP activity. Technical developments within the tool include adding attribute fixes to allow the concatenation of CORDEX data and improving the regridding support as well as the handling of the time ranges.

Revision table			
Version	**Date**	**Name**	**Comments**
Release for review	26/11/2021	Saskia Loosveldt Tomas	First version of the document
First revision	02/12/2021	Rémi Kazeroni	Comments from first reviewer
Second revision	10/12/2021	Sylvie Joussaume	Comments from second reviewer
Final version	15/12/2021		
Dissemination Level			
PU	Public		X
CO	Confidential, only for the partners of the IS-ENES3 project		

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 824084

Table of contents

Executive Summary

This report covers the developments done in the ESMValTool to support regional modelling experiments and climate predictions. In particular, the added capabilities allow ESMValTool to directly use CORDEX (Coordinated Regional Downscaling Experiments) and DCPP (Decadal Climate Prediction Project) experiments, improving the accessibility of both datasets to scientists. Other regional modelling or climate predictions experiments will also benefit from this work, but out-of-the-box support is not guaranteed.

The developments described in this document comprehend not only the required improvements to allow ESMValTool to load and preprocess the CORDEX and DCPP data but also the creation of diagnostics showcasing these added capabilities. The basic development includes additions to allow the tool to correctly find and interpret the extra tags required by these new experiments (compared to CMIP), improvements to the recipe definition format to simplify the addition of these datasets and a complete rework of the way the time coordinate is managed in the recipes. It is worth to note that the work on those new experiments has also led to fixes and improvements that will benefit other use cases of the tool, including some that were requested a long time ago.

The diagnostics described in this work are not only a showcase of these new core functionalities but also are meant to be examples of how users can leverage these new capabilities to create their own analysis. This is specially important for the future, as the continuous improvement of the work presented here relies on the community using it and finding ways to drive it further to accomplish their goals with the guidance and help from the development team.

1. Objectives

The Earth System Model Evaluation Tool (ESMValTool) (Righi et al. 2020) is a powerful and extensible community diagnostics package whose main objective is to facilitate the analysis of the Coupled Model Intercomparison Project (CMIP) (Eyring et al. 2016) projections and, in general, any model experiments. Its ability to automate data preprocessing along with the flexibility provided by the way it manages the addition of diagnostic scripts has proved invaluable for the analysis of the recent CMIP phase 6 (CMIP6) projections as stated by its use for the elaboration of the Sixth Assessment Report from the Intergovernmental Panel on Climate Change (IPCC).

In an effort to bring the tool to a wider community, the IS-ENES3 project committed to expand its capabilities to other types of experiments, in particular to Regional Climate Models (RCMs) and climate predictions. Those particular use cases align perfectly with the ESMValTool philosophy as they will allow the tool to analyse the results from the Coordinated Regional Climate Downscale Experiment (CORDEX) (Gutowski Jr. et al. 2016) and the Decadal Climate Prediction Project (DCPP) (Boer et al. 2016).

The objective of the work described in this deliverable is to set the foundations of a more versatile ESMValTool that can be easily used for those two new use cases by giving the tool the basic capabilities to handle those new kinds of experiments. Those developments are currently under extensive review and expected to be included in version 2.5 of the ESMValCore and the ESMValTool, scheduled for February 2022.

From that point on, the community is expected to keep refining and expanding these functionalities as new user requirements emerge while using these new features.

2. Application to Regional Climate Models

RCMs supplement coarser-resolution global models providing high-resolution climate simulations for limited regional domains, necessary for impact assessment and adaptation planning at regional and local scale. Additionally, in CMIP6 some global simulations will start to reach these high resolutions (≈ 25 km) within the High Resolution Model Intercomparison Project (HighResMIP). We will extend ESMValTool so that it can be applied to output from both RCMs and Earth System Models. To this end, we will tackle the challenges posed by a range of different curvilinear coordinate systems that are common in RCMs and by limited domains. Furthermore, we will develop solutions to deal with the vast amounts of data that stem from high-resolution RCM simulations (up to 1-2km).

Using the same tool for both RCMs and ESMs unifies analysis and simplifies the intercomparison of climate data generated by global and regional climate models considerably. To achieve these goals, we will leverage our prior experience in the CORDEX project and, consequently, ESMValTool is likely to play an important role in the future evaluation in CORDEX.

● 2.1 Computation of ETCCDI Extreme Events indices from EURO-CORDEX datasets

This section is going to describe the methodology adopted to compute ETCCDI (Expert Team On Climate Change Detection and Indices) indexes from regional CORDEX datasets, including the code edits and minor model fixes. Concerning the regional CORDEX datasets, the different RCMs included in the CORDEX Program were used over the European domain (EURO-CORDEX) with a highest resolution of 0.11 degree (about 12 km), and forced by different global climate models. Further information about the initiative is available in the report "Guidance for EURO-CORDEX climate projections data use" published by the EURO-CORDEX community (Hennemuth et al. 2017).

In the ESMValTool's repository a set of diagnostics able to compute ETCCDI indices from global climate dataset already exists; our aim was to use the same code for EURO-CORDEX. To achieve this purpose, we have applied several changes to the ESMValCore preprocessor module, and we have rewritten part of the native diagnostic, creating an alternative version to evaluate this regional dataset.

These improvements are tested on the following software versions: ESMValCore: 2.3.0; ESMValTool: 2.3.0; Synda: 3.8.

2.2 Changes to ESMValCore's preprocessor module

The preprocessor module in the ESMValTool core library, in the 2.3 version, still provides partial support for regional datasets. It is thought mainly to process global datasets which are mapped onto a regular lat/lon grid, further improvements about regridding are discussed in section 2.5. The idea is to make the tool able to automatically retrieve the dataset from ESGF nodes and to pre-process it, preventing any issues related to irregular grid projections (rotated pole, Mercator….).

2.2.1 CORDEX model's name convention in configuration file

The first minor code edit concerns the section related to CORDEX dataset in the configuration file called *config-develop.yml*. During the processing of these datasets it emerged that it could be useful to keep domain and driver keys also in the output file name of datasets, so line 239 was updated from the contents shown in Table 1 to the contents in Table 2.

```
{short_name}_{dataset}_{exp}_{ensemble}_{rcm_version}_{mip}
```

Table 1: Original line 239.

```
{short_name}_{domain}_{driver}_{dataset}_{exp}_{ensemble}_{rcm_version}_
{mip}
```

Table 2: Edited line 239.

Pull Request #1303 (PR) in the ESMValCore repository about this improvement was opened on GitHub and we are planning to merge this PR for the next release.

2.2.2 Synda query for CORDEX datasets

In the version 2.3, ESMValtool is not yet able to download the EURO-CORDEX at runtime because the support to compose a correct Synda-query is still unimplemented. We added this missing feature to enable the automatic download of data from ESGF nodes, in the file *esmvalcore/preprocessor/_download.py* as indicated in Table 3.

```
elif project == "CORDEX":
    query = {
        'institute':  variable.get('institute'),
        'rcm_name' :  variable.get('rcm_name'),
        'rcm_version' : variable.get('rcm_version'),
        'project': variable.get('project'),
        'frequency': variable.get('frequency'),
        'ensemble': variable.get('ensemble'),
        'experiment': variable.get('exp'),
        'variable': variable.get('short_name'),
        'domain' : variable.get('domain'),
        'driving_model' : variable.get('driver')}
```

Table 3: *elif* block in the function *_synda_search_cmd*, added to support the new functionality.

The block introduces the correct set of keys to get any regional CORDEX dataset, including Euro-CORDEX.

An example of a typical query to include in the section "datasets" of any recipe can be found in Table 4:

```
datasets:
  - { project: CORDEX, variable : pr , domain: EUR-11, driver: ICHEC-EC-EARTH,
      exp: historical, ensemble: r12i1p1, institute: CLMcom,
      rcm_name: CCLM4-8-17,dataset: CLMcom-CCLM4-8-17, rcm_version: v1,
      mip: day, start_year: 1949,
      end_year: 1955}
```

Table 4: Example of a *datasets* section loading EURO-CORDEX data.

This feature enhancement was submitted on GitHub in PR #1293 further details on the source code can be found there. In this test case Synda has been used to retrieve data from ESGF Nodes because it represented the state of art when the test case was developed. The latest versions of ESMValtool have restricted the support for Synda. However, also the most recent driver, ESGF-Python, can be used to retrieve data as reported in Obtaining input data — ESMValTool 2.4.0 documentation.

2.2.3 Fixing unmatching attributes in pairwise concatenation

One of the tasks of the preprocessor is to concatenate the datasets split on multiple files, for instance the EURO-CORDEX are typically split into multiple NetCDF files and each of them contains 5 years of climate data. The preprocessor operates a pairwise concatenation several times instead of merging all files together at once. This should prevent errors given by discontinuous time axes. However, we found an issue coming from this methodology. Sometimes an attribute unmatches between partially merged datasets due to implicit operations done by the Iris (v3.0.1) backend. The issue emerged from the datasets provided by CCLM (Cosmo Climate Limited-area Modeling Community). The pairwise concatenation causes an implicit type conversion from big endian encoding to little endian, which implies a type unmatching between the partially merged file and the files still unmerged. A possible solution is being discussed in an already existing pull request on GitHub (PR #1068), its target is to fix the endianness to get a match between each pair. From our side, we propose an alternative and more generic solution. The idea is to fix any problem related to either type or attribute unmatching during concatenation, thus we created a more generic *_equalize_attributes* function. If the preprocessor throws an exception during a pairwise concatenation, *_equalize_attributes* tries to equalize the attributes and types of the two Iris cubes by taking as reference the attributes of the first cube. If there are exceeding attributes, these will be removed. The file *esmvalcore/preprocessor/_io.py* was edited as shown in Table 5.

```
def _by_two_concatenation(cubes):
    """Perform a by-2 concatenation to avoid gaps."""
    concatenated = iris.cube.CubeList(cubes).concatenate()
    if len(concatenated) == 1:
        return concatenated[0]
    logger.debug("Found unmatching attributes in cubes during
    pairwise concatenation. I'm trying to equalise them!")
    cubes_adj = _equalise_attributes(cubes)
    concatenated = iris.cube.CubeList(cubes_adj).concatenate()
    if len(concatenated) == 1:
        return concatenated[0]
    logger.debug("Found unmatching attributes in cubes during
    pairwise concatenation. I'm trying to concatenate it as
    overlapping cubes.")
    concatenated = _concatenate_overlapping_cubes(concatenated)
    if len(concatenated) == 2:
        _get_concatenation_error(concatenated)
    else:
        return concatenated[0]
```

Table 5: Edits inthe function *_by_two_concatenation*, found in the ESMValCore *_io* module.

More details about the source code are available in a new pull request opened on GitHub at PR #1311.

2.3 ETCCDI Extreme Events from EURO-CORDEX datasets

After the preprocessors issues have been addressed, ESMValTool is ready to preprocess EURO-CORDEX datasets. Since climate change is one of the most current subjects in the Earth System Science, we have chosen to compute ETCCDI indexes from climatological datasets retrieved from ESGF nodes. Instead of implementing the diagnostics from scratch, we choose to adapt an existing one.

In the official documentation there is an implementation of Extreme Events Indices (ETCCDI) done by CICERO and other institutions. The *recipe_extreme_events.yml* enables the user to compute any ETCCDI index on a global domain and to compare the performances of the selected datasets by time-plots or Gleckler-plots. The statistics are written in R language, but most of the computation is performed by NCO and CDO command line operators called from R scripts.

2.3.1 The new diagnostics

The original script is located in the *esmvaltool/diag_scripts/extreme_events* directory, which can be found in the ESMValTool main branch. The changes made to the original code concern the adaptation of the code to CORDEX name convention, which is quite different from the CMIP5 name convention for global datasets. Since the source code is strongly focused on the processing of global datasets, the functions were partially rewritten. The new diagnostic was called e*xtreme_events_cordex* to distinguish it from the "global" version; it is possible to call it from any recipe.

Hence, several indicators and metrics of error are displayed in only a single view. The plot parameters could be set up in the recipe, a complete description of the keys that could be added in a recipe is available at the following link. The implementation of this code was submitted on GitHub, the code edits are being discussed under PR #2321.

2.4 Guide: How to create a simple recipe about ETCCDI index.

In this brief guide we are going to describe how to create a simple recipe using our diagnostic, following the contributing guidelines that are already available in the ESMValTool official documentation. The aim is to create a time-plot and a Gleckler's plot to compare ETCCDI indices computed from different EURO-CORDEX datasets.

2.4.1 Selection of CORDEX datasets

The catalog of the EURO-CORDEX datasets is available on any ESGF node. For instance, it is possible to access the following node https://esgf-data.dkrz.de/projects/esgf-dkrz/ and perform search using the graphical interface.

Each CORDEX dataset is represented by the fields defined in Table 7:

```
- {project: CORDEX, domain: EUR-11, driver: ICHEC-EC-EARTH, exp: historical,
ensemble: r12i1p1, institute: CLMcom, rcm_name: CCLM4-8-17, dataset: CLMcom-
CCLM4-8-17, rcm_version: v1, mip: day, start_year: 1981, end_year: 2000}
```

Table 7: Dataset tags required to load CORDEX data. (1) *project* is the project name. (2) *domain*: EUR-11 stands for the Europe domain at 11km. (3) The *driver* is the driving global model. (4) *exp* could be either historical or a scenario. (5) *ensemble* is the ensemble member code[3]. (6) the name of the institute. (7) *rcm_name* is the name of the regional climate model. (8) *rcm_version* is the model version. (9) *mip* is the time frequency. (10) *start_year* is the first year. (11).*end_year* is the last year.

We selected 13 datasets, listed in Table 8, which contained the precipitation variable and added them to the recipe. The statistics are computed over a period of 20 years. The YAML anchor *cordex_exp* contains tags with common values between datasets.

```
- {<<: *cordex_exp, driver: ICHEC-EC-EARTH, institute: CLMcom, rcm_name: CCLM4-8-17, dataset: CLMcom-CCLM4-8-17, rcm_version: v1 }
- {<<: *cordex_exp, driver: CNRM-CERFACS-CNRM-CM5, rcm_name: RCA4, dataset: SMHI-RCA4, rcm_version: v1}
- {<<: *cordex_exp, driver: ICHEC-EC-EARTH, institute: KNMI, rcm_name: RACMO22E, dataset: KNMI-RACMO22E}
- {<<: *cordex_exp, driver: ICHEC-EC-EARTH, institute: SMHI, rcm_name: RCA4, dataset: SMHI-RCA4, rcm_version: v1}
- {<<: *cordex_exp, driver: ICHEC-EC-EARTH, institute: KNMI, rcm_name: RACMO22E, dataset: KNMI-RACMO22E, rcm_version: v1,}
- {<<: *cordex_exp, driver: IPSL-IPSL-CM5A-MR, institute: SMHI, rcm_name: RCA4, dataset: SMHI-RCA4, rcm_version: v1}
- {<<: *cordex_exp, driver: MOHC-HadGEM2-ES, institute: CLMcom, rcm_name: CCLM4-8-17, dataset: CLMcom-CCLM4-8-17, rcm_version: v1}
- {<<: *cordex_exp, driver: MOHC-HadGEM2-ES, institute: KNMI, rcm_name: RACMO22E, dataset: KNMI-RACMO22E, rcm_version: v2}
- {<<: *cordex_exp, driver: MOHC-HadGEM2-ES, institute: SMHI, rcm_name: RCA4, dataset: SMHI-RCA4, rcm_version: v1}
- {<<: *cordex_exp, driver: MPI-M-MPI-ESM-LR, institute: CLMcom, rcm_name: CCLM4-8-17, dataset: CLMcom-CCLM4-8-17, rcm_version: v1}
- {<<: *cordex_exp, driver: MPI-M-MPI-ESM-LR, institute: MPI-CSC, rcm_name: REMO2009, dataset: MPI-CSC-REMO2009, rcm_version: v1}
- {<<: *cordex_exp, driver: MPI-M-MPI-ESM-LR, institute: SMHI, rcm_name: RCA4, dataset: SMHI-RCA4, rcm_version: v1a}
- {<<: *cordex_exp, driver: MPI-M-MPI-ESM-LR, institute: MPI-CSC, rcm_name: REMO2009, dataset: MPI-CSC-REMO2009, rcm_version: v1}
```

Table 8: Datasets used to produce Figure 1 and Figure 2.

2.4.2 Diagnostic section

In the third section of the recipe, we specify the diagnostic and the plot parameters, as presented in Table 9 below:

```
Diagnostic

  extreme_events:
    description: calculate extreme events over Europe

    variables:
        pr:
          mip: day

    scripts:
      main:
        script: extreme_events_cordex/extreme_events.R
        reference_datasets: "ICHEC-EC-EARTH_CLMcom-CCLM4-8-
           17_historical_r12i1p1_v1"
        regrid_dataset: "ICHEC-EC-EARTH_CLMcom-CCLM4-8-17"
        mip_name: EUROCORDEX
        timeseries_idx:["cddETCCDI_yr",'r95pETCCDI_yr', 'r99pETCCDI_yr',
           'rx5dayETCCDI_mon', 'sdiiETCCDI_yr']
        gleckler_idx:["cddETCCDI_yr",'r95pETCCDI_yr', 'r99pETCCDI_yr',
           'rx5dayETCCDI_mon', 'sdiiETCCDI_yr']
        ts_plt: true
        glc_plt: true
        base_range: [1981, 2000]
        analysis_range: [1981, 2000]
```

Table 9: Definition of the diagnostics used to produce Figure 1 and Figure 2.

Here is defined a diagnostic named "extreme_events" which involves the precipitation variable (daily frequency). The diagnostic refers to the script *extreme_events_cordex/extreme_events.R*. The main block contains all parameters to be passed to our script.

1. *reference_datasets*: A list of datasets to be used as reference (e.g. observations)
2. *regrid_dataset*: Before compute the average value for a statistic, all datasets are re-gridded to a common reference grid.
3. *mip_name*: The name to give to the ensemble of all models into the time-plot.
4. *timeseries_idx*: ETCCDI indices for which a time-plot must be created
5. *gleckler_idx*: ETCCDI indices to include in Gleckler's plot.
6. *ts_plot/glk_plt*: a logical value to enable or disable the time-plot/gleckler-plot
7. *base_range/analysis_range*: Some ETCCDI indices require a normalization with respect to a reference period.

The complete recipe is present in the etccdi_for_cordex branch, it has the name recipe_extreme_events_cordex.yml, hence, it could be run directly from command line once the branch will be merged to the main branch.

2.4.3 Run the recipe

To run the recipe, you have to launch the command from the shell shown in Table 10:

```
esmvaltool run recipe_extreme_events_cordex.yml --config_file=config-user.yml
--synda-download
```

Table 10: Command to run the ESMValTool, activating the download of files activating the --synda-download flag for the retrieval of the datasets.

The data will be retrieved in an automatic way by Synda from ESGF nodes. If the data is already present in the local storage, the path must be specified in the configuration file as explained in the ESMValTool's documentation.

2.4.4 The outputs

In the */plot* directory of the output of the recipe run, it will be possible to find the following plots.

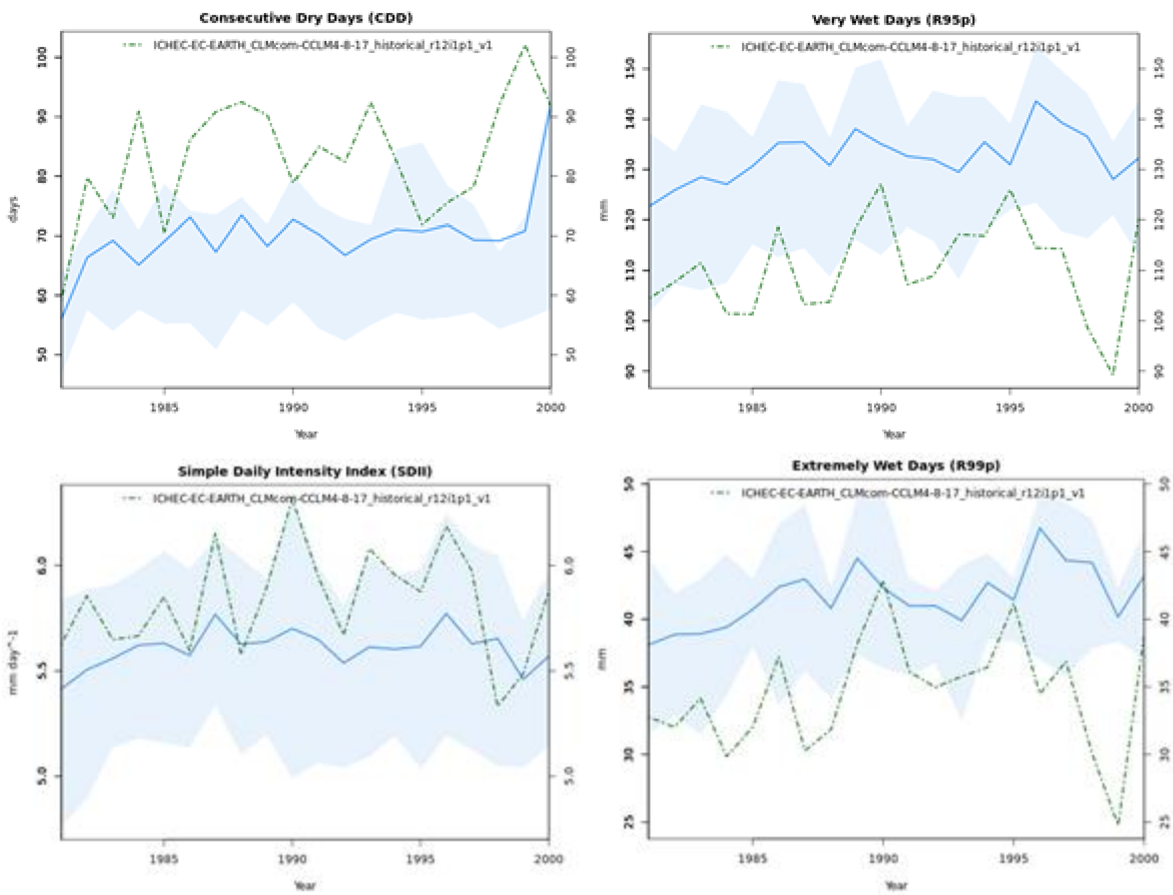

Figure 1: Time series of the CDD index (top left), R95p index (top right), SDII index (bottom left) and R99p index (bottom right) over 1981-2000 for a selection of EURO-CORDEX regional models, the EURO-CORDEX multi-model mean (thick blue line) and reference datasets "ICHEC-EC-EARTH_CLMcom-CCLM4-8-17_historical_r12i1p1_v1" (dashed line in green). Shading is used to reproduce the multi-model spread.

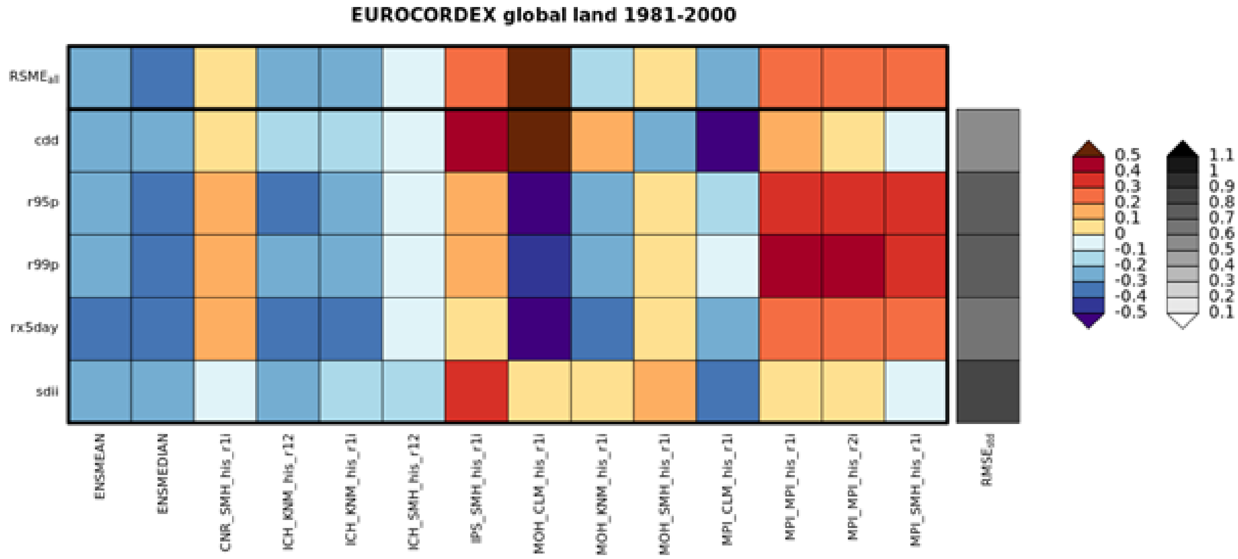

Figure 2: Portrait plot of relative error metrics for precipitation extreme indices calculated with a selection of EURO-CORDEX regional models, evaluated over 1981-2000.

• 2.5 Improved Regridding Support

As has been hinted at in Section 2.2, one challenge with regional models is their use of different grids than global models. This is a natural consequence of their regional focus. Where all commonly used regular global grids exhibit singularities or otherwise undesirable features somewhere, often close to the poles, regional grids can be chosen such that these undesirable features are outside or even far away from the area of interest. Within CORDEX, the principal kind of grid chosen is that of regular rotated poles. Additionally, a few regional models that are part of CORDEX use other projections, such as the Lambert-Conformal projection as their native grid.

Preliminary support for these kinds of grids has already been introduced into ESMValTool in an earlier version (Righi et al., 2020), however, further developments were deemed necessary.

To maximize the impact of these developments, they have been carried out in collaboration with Iris developers, making them available not only for ESMValTool, but for the wider circle of users of the Iris library (Iris v3.1.0, 2021) and have been placed in their own github repository at https://github.com/SciTools-incubator/iris-esmf-regrid.

The new regridding support brings improvements in performance both in terms of required computing time and memory efficiency, which will help with the analysis of high resolution and multi-model data.

It will also increase the accuracy and flexibility in choosing different regridding methods, such as first and second order conservative regridding, by better integration with the underlying ESMF regridding library.

This work benefits from the recent improvements in the handling of so-called fx variables, that is variables that contain cell measures such as cell area, that have been released with ESMValTool versions 2.3 and 2.4.

Looking beyond the regridding needs for regional models, this approach to regridding has allowed developers in other projects to integrate regridding even for fully irregular grids into the same framework. These grids are gaining popularity for the next generation of global models, such as the FESOM ocean model and the ICON atmospheric model.

• 2.6 Interactions with other Projects

It is worth noting that the support for regional models developed within IS-ENES3 is timely, since also other projects and actors have started applying ESMValTool to regional models. This includes the Horizon 2020 project EUCP (https://www.eucp-project.eu), which is developing analyses partly based on the work presented here, as well as the SME Predictia (https://predictia.es/en) that is applying the tool in the private sector and helping with its development for impact applications.

3. Flexible use across different types of experiments and timescales

The ESMValTool was designed to evaluate mainly climate projection experiments. These particular experiments have some peculiarities that are not shared by other kinds of experiments regarding the time handling:

- Projections are usually initialized on January 1st of the selected year and run to December 31st of the last simulated year.
- All the model runs for a projection experiment have the same start date

Prediction experiments do not comply those assumptions:

- They can be initialized on different days of the year (i.e. May 1st and November 1st for seasonal predictions to predict summer / winter seasons).
- Predictions are not run until the end of the last year but for a limited time period, which can range from several months for seasonal predictions to 5-10 years for the decadal ones.
- Model runs from the same prediction experiment have different start dates to get as many different pairs of initialization conditions and observations as possible to properly assess the performance of the model.

Due to the enormous variety of predictions available, the work is focused on supporting the decadal predictions from DCPP but in a way that will allow the usage of ESMValTool for other kinds of predictions. The community is encouraged to build upon this work and add support for new predictions as needed.

3.1 Support for sub-experiment IDs

One of the differences between the support required by CMIP experiments with respect to DCPP ones, is the need for a new tag called *sub_experiment* that is used to identify the start date of hindcast and forecast experiments. Up until version 2.4 of the ESMValTool, the tag can be defined in the dataset section of the recipe, but its value has to be specified for every *sub_experiment* entry as seen in Table 11:

```
datasets:
  {<<: *dcpp_exp1, sub_experiment: s1960}
  {<<: *dcpp_exp1, sub_experiment: s1961}
  {<<: *dcpp_exp1, sub_experiment: s1962}
  {<<: *dcpp_exp1, sub_experiment: s1963}
  {<<: *dcpp_exp1, sub_experiment: s1964}
  ...
  {<<: *dcpp_exp1, sub_experiment: s2014}
```

Table 11: Example of a dataset definition using the sub-experiment tag for ESMValTool v2.4 and earlier. The entry *dcpp_exp1 is a YAML anchor defining other tags needed to call a dataset.

Recipes defined using this syntax will quickly become too extensive to be readable, especially if they use multiple models or experiments, as it is usual within ESMValTool.

To simplify this dataset definitions, a modification was introduced to the tool in Pull Request (PR) #771, that allows users to define the required *sub_experiment* in a more concise and simplified way, shown in Table 12, letting ESMValTool expand it to the full range of values:

```
{<<: *dcpp_exp1, sub_experiment: s(1960:2019)}
```

Table 12: Example of a dataset section using the compact sub_experiment tag.

A similar feature was already in place for the *ensemble* tag and was used as a reference for the implementation of this new feature to keep consistency across tags. When both of them are combined, the definition of a DCPP dataset with 20 members and all its *sub_experiments* can be done in one line (Table 13) instead of 1200 which would correspond to the 60 start dates with 20 ensemble members each:

```
{<<: *dcpp_exp1, ensemble: r(1:20)i1p1f1, sub_experiment: s(1960:2019)}
```

Table 13: Example of a dataset where both the members and the sub_experiment are called in a compact syntax.

● 3.2 Improving time range definitions

Once a way to handle the *sub_experiment* tag is developed, the next objective is to provide a procedure and syntax that facilitates the handling of the time range associated with each sub-experiment. In Table 13 all references to the time range were avoided to exemplify the feature described, but given the case in which a user wanted to load the whole ten years of each sub-experiment, the ESMValTool would still require a specification for the start and end years. This case is presented in Table 14:

```
{<<: *dcpp_exp1, sub_experiment: s1960, start_year: 1960, end_year: 1969}
{<<: *dcpp_exp1, sub_experiment: s1961, start_year: 1961, end_year: 1970}
{<<: *dcpp_exp1, sub_experiment: s1962, start_year: 1962, end_year: 1971}
{<<: *dcpp_exp1, sub_experiment: s1963, start_year: 1963, end_year: 1972}
{<<: *dcpp_exp1, sub_experiment: s1964, start_year: 1964, end_year: 1973}
…
{<<: *dcpp_exp1, sub_experiment: s2019, start_year: 2019, end_year: 2028}
```

Table 14: Example of a dataset definition using the start_year and end_year tags in DCPP data for ESMValTool v2.4 and earlier.

The work developed in PRs #1133 and #1214 aims to substitute the use of the *start_year* and *end_year* tags to define the time range with a new *timerange* tag based on ISO 8601 standard for date- and time-related data.

When using the *timerange* tag to specify the start and end points, possible values can be defined as follows:

- A start and end point specified with a resolution up to seconds (YYYYMMDDThhmmss)
  - *timerange: '1980/1982*'. Spans from 01/01/1980 to 31/12/1982.
  - *timerange: '198002/198205*'. Spans from 01/02/1980 to 31/05/1982.
  - *timerange: '19800302/19820403'*. Spans from 02/03/1980 to 03/04/1982.
  - *timerange: '19800504T100000/19800504T110000*'. Spans from 04/05/1980 at 10h to 11h.
- A start point or end point, and a relative period with a resolution up to seconds (P[n]Y[n]M[n]DT[n]H[n]M[n]S).
  - *timerange: '1980/P5Y*'. Starting from 01/01/1980, spans 5 years.
  - *timerange: 'P2Y5M/198202'*. Ending at 28/02/1982, spans 2 years and 5 months.
- A wildcard to load all available years, the first available start point or the last available end point.
  - *timerange:* '*'. Finds all available years.

- *timerange: '*/1982'*. Finds first available point, spans to 31/12/1982.
- *timerange: '*/P6Y*'. Finds first available point, spans 6 years from it.
- *timerange: '198003/*'*. Starting from 01/03/1980, spans until the last available point.
- *timerange: 'P5M/*'*. Finds the last available point, spans 5 months backwards from it.

The example in Table 14 can now be written in a single line, summarized in Table 15:

```
{<<: *dcpp_exp1, sub-experiment: s(1960:2019), timerange: ‘*’}
```

Table 15: Example of a dataset definition using the timerange tag in DCPP data. The wildcard implies that all data available will be loaded.

But the new syntax also makes it possible to specify as one liner new and richer use cases in a way that is more readable and less error prone than previous versions of the tool shown in Tables 16 and 17 :

- Get the first two years after initialization

```
{<<: *dcpp_exp1, sub-experiment: s(1960:2019), timerange: ‘*/P2Y’}
```

Table 16: Example of a dataset definition using the timerange tag and duration periods from a starting point in DCPP data.

- Get the last three years after initialization

```
{<<: *dcpp_exp1, sub-experiment: s(1960:2019), timerange: ‘P3Y/*’}
```

Table 17: Example of a dataset definition using the timerange tag and duration period from an endpoint in DCPP data.

Although not required for DCPP experiments, the syntax allows *timerange* specifications with resolution up to seconds to make it suitable for shorter term predictions, like seasonal or subseasonal, which will usually require time range definitions with resolution up to months or even days.

The work carried in both pull requests contains a complete overhaul of the code used to select the relevant data files according to the request date range and of the actual selection of the requested time after the data is loaded (#1133).

It is also important to note that this work can also help the resolution of old and long-standing issues, in Issue #76 and #345.

• 3.3 Practical examples

A recipe and diagnostic were developed with the purpose to showcase the implemented changes in the handling of the time ranges. The developments can be found in PR #2241. The recipe loads the first member of variable Near-Surface Air Temperature (*tas*) of the *dcppA-hindcast* experiment for a run of the *EC-Earth3* model. This experiment is composed of sub-experiments initialised at start-dates ranging from 1960 to 2018. Each *sub-experiment* comprises 11 years of data initialised in November for each one of the start dates, as shown in Table 18.

```
tas_Amon_EC-Earth3_dcppA-hindcast_s1960-r1i1p1f1_gr_196011-196110.nc
...
tas_Amon_EC-Earth3_dcppA-hindcast_s1960-r1i1p1f1_gr_197011-197110.nc

...

tas_Amon_EC-Earth3_dcppA-hindcast_s2018-r1i1p1f1_gr_201811-201910.nc
...
tas_Amon_EC-Earth3_dcppA-hindcast_s2018-r1i1p1f1_gr_202811-202910.nc
```

Table 18: Input file syntax for monthly data in a DCPP experiment. Each file spans 12 months from the sub_experiment date, starting from November to October.

The recipe also loads *tas* for the observational dataset *ERA-Interim*. Both datasets are pre-processed using the ESMValTool preprocessor `area_statistics`. Once the global mean is computed, the resulting time series are plotted in a simple diagnostic script in order to obtain the comparison between the observational values and the model values.

The first example, defined in Table 19, will load all the data available for *sub_experiments* s1980 to s2018, since the *timerange* tag is set to a wildcard. The choice of this range for the *sub_experiments* is due to the fact that the *ERA-Interim* data stored in the BSC's archive ranges from January 1980 to December 2018. The datasets in the recipe are defined using tags relevant to the BSC's Data Reference Syntax (DRS).

```
datasets:
  - {project: ECEARTH, expid: a1ua, dataset: EC-Earth3, exp: dcppA-hindcast,
sub_experiment: 's(1980:2018)', ensemble: 'r1i1p1f1', timerange: '*'}

  - {project: OBS, type: recon, institute: ecmwf, dataset: erainterim,
freq_folder: monthly_mean, freq_base: '', timerange: '198011/201812'}
```

Table 19: Dataset section definition for Example 1. Loads a DCPP experiment and an observational dataset. The time range is set to a wildcard to load all available data in the DCPP experiment. For the observational dataset, the requested data spans November 1980 to December 2018.

Since the model data is initialized in November 1980, the start of the *timerange* for *ERA-Interim* also gets set to this value in order to match the model data. Regarding the end of the time range, the last *sub_experiment* spans to October 2029, whereas the *ERA-Interim* data spans until December 2018. This difference in the time ranges is represented in Figure 5.

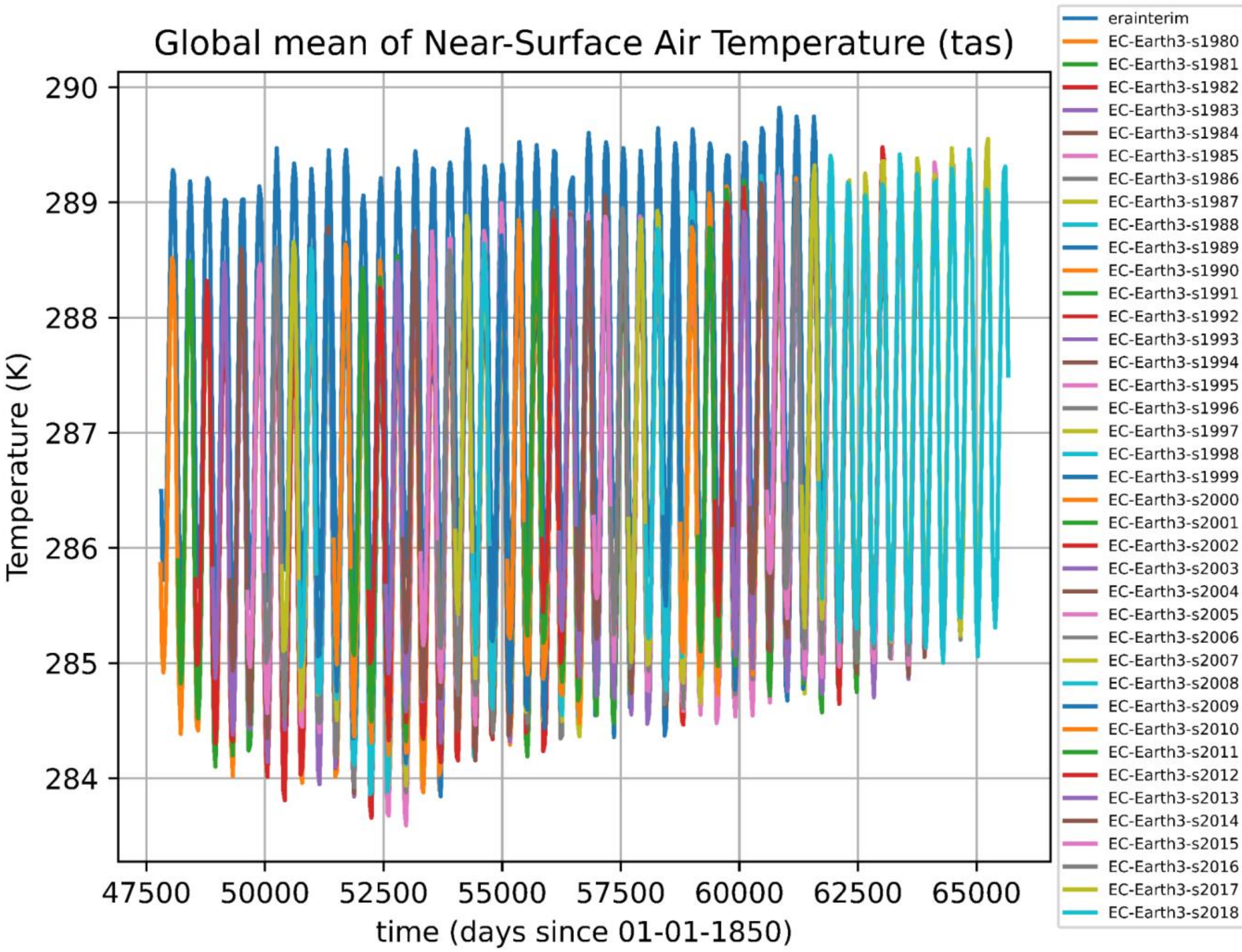

Figure 3: Time series representing the global mean of Near-Surface Air Temperature for the datasets defined in Table 19. As defined in the recipe, both the observational dataset and the model dataset start at the same point, but the model data spans a larger time range.

As a second example, defined in Table 20, the model datasets get re-defined to only load data up until December 2018, therefore matching the range for the ERA-Interim data. The data in *sub_experiment* s2007 spans until October 2018, therefore all previous *sub_experiment*s load all data available setting the *timerange* tag to a wildcard. On the other hand, *sub_experiment*s from s2008 to s2018, which range beyond the *ERA-Interim timerange*, load data from the first available point in each *sub_experiment* until December 2018, as showcased in Figure 6.

```
datasets:
  - {project: ECEARTH, expid: a1ua, dataset: EC-Earth3, exp: dcppA-hindcast,
sub_experiment: 's(1980:2007)', ensemble: 'r1i1p1f1', timerange: '*'}

  - {project: ECEARTH, expid: a1ua, dataset: EC-Earth3, exp: dcppA-hindcast,
sub_experiment: 's(2008:2018)', ensemble: 'r1i1p1f1', timerange: '*/201812'}

  - {project: OBS, type: recon, institute: ecmwf, dataset: erainterim,
freq_folder: monthly_mean, freq_base: '', timerange: '198011/201812'}
```

Table 20: Dataset section definition for Example 2. Loads a DCPP experiment and an observational dataset. The time range is set to a wildcard to load all available data for sub_experiments s1980 to s2007. For the remaining experiments, the time range is set to span up to December 2018 in order to match the time range of the observations.

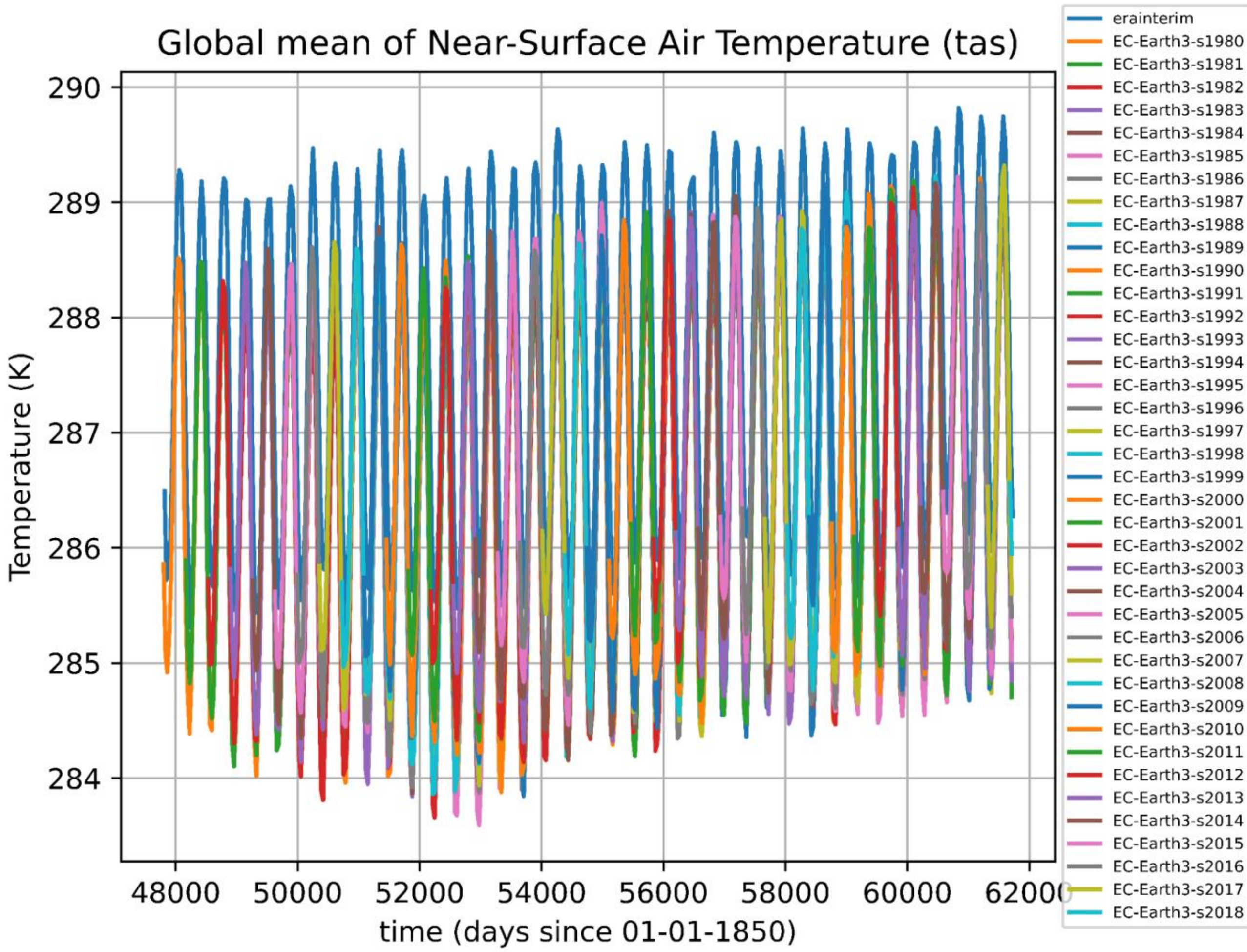

Figure 4: Time series representing the global mean of Near-Surface Air Temperature for the datasets defined in Table 20. As defined in the recipe, both the observational dataset and the model dataset start and end at the same point.

Finally, in order to showcase the use of duration periods, the recipe defined in Table 21 is set to only load one year from the first available point per *sub_experiment*. Results are illustrated in Figure 7.

```
datasets:
  - {expid: a1ua, dataset: EC-Earth3, project: ECEARTH, exp: dcppA-hindcast,
sub_experiment: 's(1980:2018)', ensemble: 'r1i1p1f1', timerange: '*/P1Y'}

  - {project: OBS, type: recon, institute: ecmwf, dataset: erainterim,
freq_folder: monthly_mean, freq_base: '', timerange: '1980/2018'}
```

Table 21: Dataset section definition for Example 3. Loads a DCPP experiment and an observational dataset. For the experiments, the time range is set to a duration period that will span one year from the first available point. The observations will span from January 1980 to December 2018.

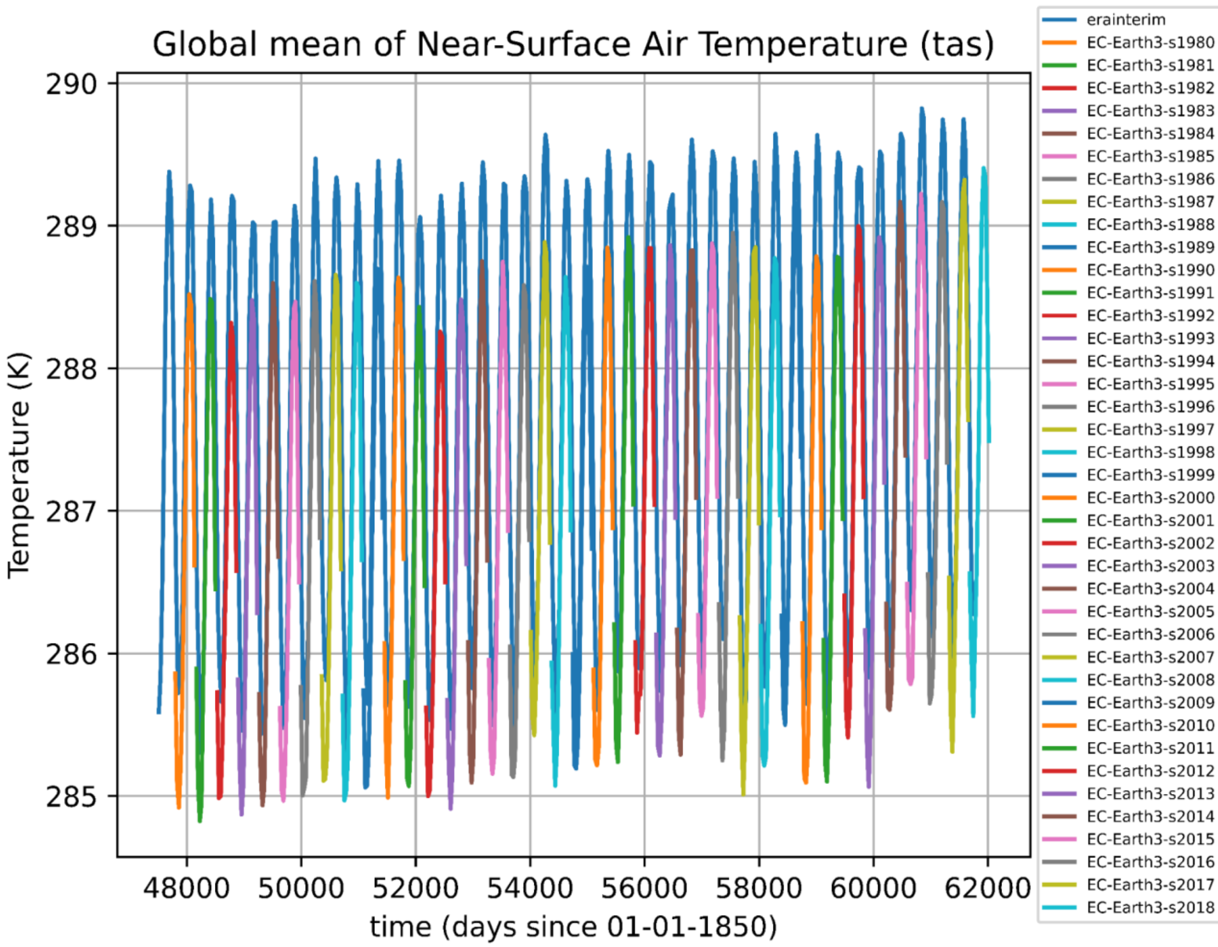

Figure 5: Time series representing the global mean of Near-Surface Air Temperature for the datasets defined in Table 21. The model data spans one year per sub-experiment, from November to October.

3.4 Wildcards and reproducibility

While the use of wildcards in the *timerange* tag is a convenient asset in order to define datasets in a compact way, the data that gets loaded is subjected to its availability in the local storage. Which means that the results may differ depending on the archive in which the data is retrieved from.

In order to avoid ambiguities and make the recipes reproducible across different platforms, the work in PR [#1133]() makes the ESMValTool create a copy of the recipe in which the wildcards in the *timerange* are filled with the start and end values that have been found. This copy, the contents of which are summarized in Table 22, is stored in the `/run` sub-directory of the recipe output directory under the name *recipe_XXX_filled.yml*, where *recipe_XXX.yml* is the name of the original recipe.

```
...
    variables:
      tas:
        mip: Amon
        preprocessor: pptas
        grid: gr
        additional_datasets:
        - dataset: EC-Earth3
          project: ECEARTH
          exp: dcppA-hindcast
          timerange: 198011/199110
          ensemble: r1i1p1f1
          sub_experiment: s1980
          expid: a1ua
          institute:
          - EC-Earth-Consortium
        - dataset: EC-Earth3
          project: ECEARTH
          exp: dcppA-hindcast
          timerange: 198111/199210
          ensemble: r1i1p1f1
          sub_experiment: s1981
          expid: a1ua
          institute:
          - EC-Earth-Consortium
        - dataset: EC-Earth3
          project: ECEARTH
          exp: dcppA-hindcast
          timerange: 198211/199310
```

```
          ensemble: r1i1p1f1
          sub_experiment: s1982
...
```

Table 22: Filled recipe generated for the dataset section in Table 19. The wildcard in the time range has been filled with the corresponding start and end dates, obtained from the files that have been found.

4. Conclusions

The work presented in this deliverable has allowed ESMValTool usage for two new kinds of climate experiments: regional experiments and/or predictions. These new developments allow ESMValTool users to easily use the tool to analyse CORDEX and DCPP data, expanding the usefulness of the tool to WCRP Model Intercomparison Projects.

The work also contains two diagnostics showcasing these added capabilities. Those were developed to prove that the work allows users to produce results that were not feasible in previous versions of the tool and also to provide working examples that users can use to kickstart their own diagnostic work. It needs to be noted that the developments in the diagnostic require ESMValCore pull requests that will be merged for version 2.5.

The developments presented on this deliverable have also helped to fix some long-standing issues that were part of the ESMValTool. In particular, the improved time range management provides users with a set of new tools to define their study periods that will be useful outside the scope of climate predictions and regional modelling. This is an excellent example on how community tool users can benefit from the work done to improve them even if that work's main objective is not of direct use to them.

As a final note, this work has been done by contributors with highly different levels of expertise on ESMValTool. Some of the contributors to this deliverable were already members of ESMValTool’s Core Development Team while others were complete newcomers to the tool. This fact can be seen as a sign of good health for ESMValTool as it shows that newcomers can make key contributions to the tool with a bit of guidance from the experts. That is specially important due to the fact that the continuous improvement of the features added in this work rely on ESMValTool users identifying and/or contributing new features as they find them during their daily work.

References

Boer, G. J., Smith, D. M., Cassou, C., Doblas-Reyes, F., Danabasoglu, G., Kirtman, B., Kushnir, Y., Kimoto, M., Meehl, G. A., Msadek, R., Mueller, W. A., Taylor, K. E., Zwiers, F., Rixen, M., Ruprich-Robert, Y., and Eade, R.: The Decadal Climate Prediction Project (DCPP) contribution to CMIP6, Geosci. Model Dev., 9, 3751-3777, doi:10.5194/gmd-9-3751-2016, 2016

Eyring, V., Bony, S., Meehl, G. A., Senior, C. A., Stevens, B., Stouffer, R. J., and Taylor, K. E.: Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization, Geosci. Model Dev., 9, 1937-1958, doi:10.5194/gmd-9-1937-2016, 2016.

Gutowski Jr., W. J., Giorgi, F., Timbal, B., Frigon, A., Jacob, D., Kang, H.-S., Raghavan, K., Lee, B., Lennard, C., Nikulin, G., O'Rourke, E., Rixen, M., Solman, S., Stephenson, T., and Tangang, F.: WCRP COordinated Regional Downscaling EXperiment (CORDEX): a diagnostic MIP for CMIP6, Geosci. Model Dev., 9, 4087-4095, doi:10.5194/gmd-9-4087-2016, 2016.

Hennemuth TI, Jacob D, Keup-Thiel E, Kotlarski S, Nikulin G, Otto et al J (2017) Guidance for EUROCORDEX climate projections data use. Version1.0 - 2017.08. Link: https://euro-cordex.net/imperia/md/content/csc/cordex/euro-cordex-guidelines-version1.0-2017.08.pdf

Met Office. 2021. *Iris: A Python Package for Analysing and Visualising Meteorological and Oceanographic Data Sets*. V3.1.0. Exeter, Devon. http://scitools.org.uk/.

Righi, M., Andela, B., Eyring, V., Lauer, A., Predoi, V., Schlund, M., Vegas-Regidor, J., Bock, L., Brötz, B., de Mora, L., Diblen, F., Dreyer, L., Drost, N., Earnshaw, P., Hassler, B., Koldunov, N., Little, B., Loosveldt Tomas, S., and Zimmermann, K.: Earth System Model Evaluation Tool (ESMValTool) v2.0 - technical overview, Geosci. Model Dev., 13, 1179-1199, doi: 10.5194/gmd-13-1179-2020, 2020.