

SOFTWARE METAPAPER

MurCSS: A Tool for Standardized Evaluation of Decadal Hindcast Systems

Sebastian Iling¹, Christopher Kadow¹, Oliver Kunst¹ and Ulrich Cubasch¹

¹ Freie Universität Berlin, Berlin, Germany

MurCSS (Murphy-Epstein decomposition and Continuous Ranked Probability Skill Score) is a tool for standardized evaluation of decadal hindcast-prediction systems written in Python using CDO [1] and can be downloaded at <https://github.com/illing2005/murcss>. It analyzes decadal hindcast experiments in a deterministic and probabilistic way following and extending the framework suggested by Goddard et al. [2]. It was developed as part of the MiKlip (a major project for decadal climate prediction funded by BMBF in Germany) evaluation system to improve the comparability within the project during development stages and interim test phases. It is easily applicable by other modeling groups working on decadal prediction because it complies with international standards.

Keywords: Decadal prediction, skill score, MiKlip, MESS, CRPSS, model evaluation, verification

Funding Statement: *MurCSS* was developed during the research project MiKlip (FKZ: O1LP1160A) funded by the German Federal Ministry of Education and Research in Germany (BMBF).

(1) Overview

Introduction

In recent years decadal climate predictions have become more and more popular in the climate science community. Typically an ensemble of so-called hindcast experiments with starting dates between 1960 and today are performed enabling a verification against observational data. But the evaluation and validation of decadal climate prediction systems is both a scientific and a technical challenge in the current climate research.

There are several convincing arguments for using a standardized tool for decadal model evaluation. For instance, model development stages and interim test phases of (in this case decadal climate predictions) can be assessed easily. It also simplifies the comparison of decadal prediction systems developed by different modeling groups.

Goddard et al. [2] proposed a framework for verification of decadal hindcasts addressing two key questions:

1. Do the initial conditions in the hindcast lead to more accurate predictions of the climate, compared to uninitialized climate change projections?
2. Is the prediction model's ensemble spread an appropriate representation of forecast uncertainty on average

The first question is addressed through the Murphy-Epstein decomposition of the Mean Squared Error Skill Score (*MSESS*) [3],[4] which is based on the Mean Squared

Error and can be used to compare two different model outputs (e.g. initialized and uninitialized). The *MSESS* is a deterministic skill measure of the ensemble mean. For the second question Goddard et al. [2] suggest to use a modified version of the Continuous Ranked Probability Skill Score (*CRPSS*) [5], which compares the average ensemble spread with the mean squared error. We extended the probabilistic part with the logarithmic ensemble spread score [6], which is a direct estimate for the skill of the mean ensemble spread.

MurCSS follows this framework and offers a standardized and reproducible way to calculate the above mentioned metrics for decadal prediction systems including a bootstrap method to determine significance levels. A detailed explanation of the tool, its output, and a tutorial how to use the tool can be found on our web-page (<https://www-miklip.dkrz.de/about/murcss/>).

Implementation and architecture

MurCSS is a command line tool written in Python [7] using the Python-CDO-Interface [8]. The plots are produced and saved using the python library Matplotlib [9].

The tool architecture can be separated into three components (see **Fig. 1**), file input/preparation, metric calculation, and plotting routines. The file input searches for valid files and prepares them for the actual metric calculation. This component makes use of the MiKlip file database which is based on the international standards CMOR [10] and NetCDF [11]. It is also possible using the simplified file input component (`findFilesCustom.py`) which

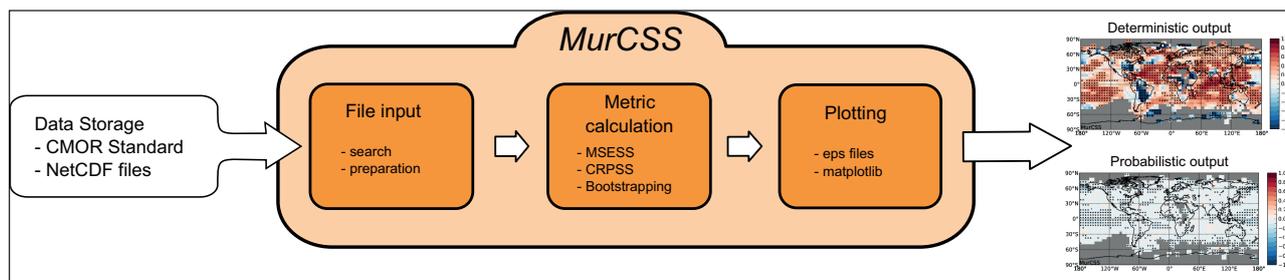


Fig. 1: The sketched work-flow of MurCSS with sample plots depicted on the right-hand side.

can be easily adjusted to the local database structure. The main part is the metric calculation, where the actual skill score calculation takes place. Most calculations are performed using CDO or the NumPy package, applying multiprocessing whenever possible. After processing the results are stored in the common NetCDF data format. In the last step these files are visualized using plotting routines.

Quality control

We developed the following strategy to assure the quality of the system.

- We assembled a set of test-data.
- Unit tests, which were constantly extended as soon as new problems were observed.
- Users within MiKlip helped as beta-testers to improve the tool.

(2) Availability

Operating system

MurCSS has been developed and tested on Linux. Although the tool should run on every system where Python and CDO are installed.

Programming language

Python 2.7, CDO 1.5.4

Dependencies

- Python Modules:
 4. Matplotlib >= 1.1.0
 5. Basemap
 6. NumPy >= 1.5.0
 7. SciPy >= 0.8.0
 8. CDOpy
- CDO
- NetCDF Libraries

Archive

Name

Figshare

Persistent identifier

<http://dx.doi.org/10.6084/m9.figshare.976060>

License

GNU General Public License 3

Publisher

Sebastian Illing

Date published

02/03/14

Code Repository

Name

GitHub

Identifier

<https://github.com/illing2005/murcss>

License

GNU General Public License 3

Date published

02/03/14

Language

English

(3) Reuse potential

MurCSS is already frequently used within the MiKlip project. The direct access to our file database and the standardized score calculation allows a quick assessment of the latest model improvements. The standardized output also facilitates the comparison of results obtained by different working groups within MiKlip. By now *MurCSS* has been used in a number of publications, for instance Pohlmann et al. [12] or Kadow et al. [13], and more in preparation.

Although it was developed to improve the comparability within the MiKlip project, we encourage other modeling groups working on decadal climate predictions to use our tool to benefit from the advantages that come with a standardized evaluation tool. In order to use the tool in projects outside of MiKlip *MurCSS* comes with a simplified file input component (`findFilesCustom.py`) which can be easily adjusted to the local data-structure in `murcss_config.py`.

The tool can also be applied to other disciplines of climate modeling, at the time of writing, for instance, we are working on an extension of *MurCSS* to seasonal forecast systems. Furthermore the modular architecture allows an easy extension to other skill scores or metrics of interest.

