

SOFTWARE METAPAPER

Validator – a Web-Based Interactive Tool for Validation of Ocean Models at Oceanographic Stations

Hagen Radtke¹, Florian Börgel¹, Sandra-Esther Brunnabend¹, Anja Eggert^{1,2}, Madline Kniebusch¹, H. E. Markus Meier^{1,3}, Daniel Neumann¹, Thomas Neumann¹ and Manja Placke¹

¹ Section of Physical Oceanography and Instrumentation, Leibniz Institute for Baltic Sea Research Warnemünde (IOW), Seestr. 15, Rostock, DE

² Leibniz Institute for Farm Animal Biology (FBN), Wilhelm-Stahl-Allee 2, Dummerstorf, DE

³ Swedish Meteorological and Hydrological Institute, Norrköping, SE

Corresponding author: Hagen Radtke (hagen.radtke@io-warnemuende.de)

Numerical ocean models, like other geoscientific models, are a strongly simplified representation of real oceans. They are used as tools to answer research questions about the real-world systems. Therefore, their thorough validation is essential to ensure that the conclusions drawn from the model experiment are valid in reality. We demonstrate a software which allows an interactive model validation through a web interface based on the R Shiny framework. At pre-defined stations, different kinds of plots can be rendered within a few seconds, according to the user's choice, allowing a live validation of different model parameters even in model simulations which are still running. This makes it different from validation approaches which generate a pre-defined set of plots after the calculations have finished and make it particularly useful for model tuning purposes. Observation data can be read in from text files or can be extracted from a database.

Once set up, the validation tool requires no technical skills to use. It can be used for single- or multi-model validation and allows saving the generated plots as high-resolution images suitable for use in scientific publications.

A Linux operating system is required for the Validator app, but via a virtual machine, the software can run on Windows or MacOS hosts as well. A Dockerfile is supplied which allows to test the software with example data without installation.

Keywords: Validation; Ocean model; Station data; Database; Model intercomparison

Funding statement: This software was developed at the Leibniz Institute for Baltic Sea Research Warnemünde (IOW) on institutional funding.

(1) Overview

Introduction

Functionality

The software aims at comparing ocean model output, which may be both physical or biogeochemical, to oceanographic observations.

The user selects

- a data set containing a pre-defined list of oceanographic stations, i.e., all stations of a particular measurement network,
- an oceanographic station from this dataset, either via a drop-down list or via a map,
- a variable to compare between model results and measurements, e.g., temperature,
- between one and four models whose results will be compared to the observed data, and
- a plot type of their choice.

Five different kinds of plots can be selected, see **Figure 1:**

- Time series
 - Model data are averaged vertically over a chosen depth interval and drawn as curves over time. Measurements from within the depth range are drawn as points.
- Vertical profile
 - Model data are time-averaged over a selected period and month range, e.g., Dec-Feb, and drawn as curves over depth. Individual measurements are drawn as points.
- Scatter plot
 - Model results at the same time and location as the measurements are plotted against these observations.
- Taylor diagram

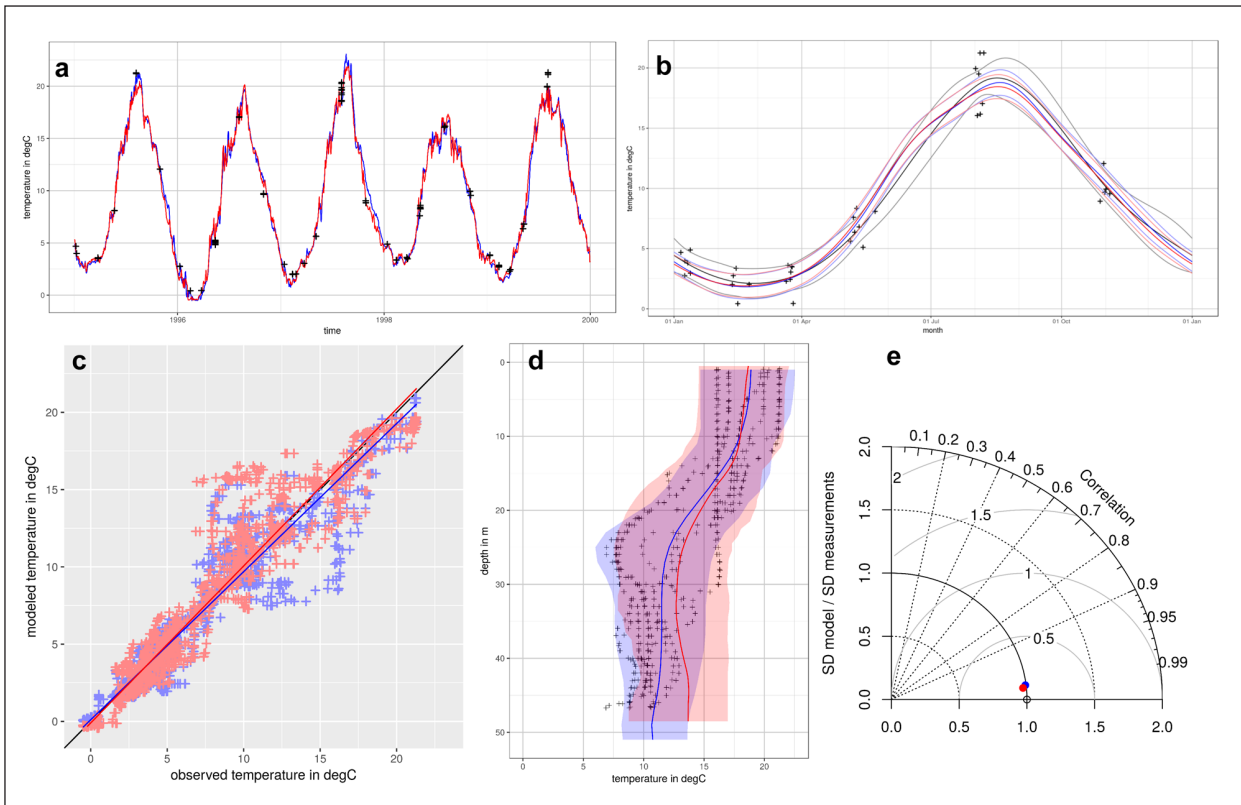


Figure 1: Examples of plots generated by Validator: **(a)** time series, **(b)** nonlinear seasonal trend, **(c)** scatter plot, **(d)** vertical profile, **(e)** Taylor diagram.

- The standard deviation in model data and measurements, the correlation between them and the standard error are shown in the same diagram following [1].
- Seasonal and/or long-term signal extracted with a Generalised Additive Mixed Model (GAMM)
 - Model data are vertically averaged to create a time series. Then, a GAMM model [2] is used to fit smooth splines through measurements and model results. Both long-term and periodic n l splines can be selected. The latter allows for a comparison of the seasonal cycle between model and measurements. This can be used to compare seasonality and long-term trends in the data.

A set of additional options exists depending on the plot type, e.g., showing the 5th and 95th percentile in a vertical profile plot. Also, the user can select a sub-range of time or depth, which allows a detailed exploration of individual aspects of model behaviour.

Implementation overview

The software is implemented as an R Shiny app [3], making it easy to use via an interactive web-based front-end. The back-end can run both on a private computer or on a server. It requires RStudio or Shiny Server, respectively, and a Linux operating system, possibly in a virtual machine. A Dockerfile is supplied such that the software can be used with example data without the need to install the software itself. The use of the Dockerfile and installation

instructions are explained in the user's guide. The software can load model data from a predefined list of models given in a text file. The same applies for measurement datasets. The app requires read access to the data over the (server-side) file system, an online upload of data is not intended. Model data have to be provided in the NetCDF format [4] and measurement data as text files or via an SQL database. We refer to the user's guide for details on the format of the required data.

Alternatives

This software is, to our best knowledge, the first example of a web-based open-source solution aimed at validating ocean model output. There are, however, alternative software packages which could be used to obtain similar validation results.

The first alternative is the use of visualisation software for geodata. This includes the programs NOAA Ferret [5] and GrADS [6], which are specialised software solutions for geodata visualisation. The diversity of statistics and plots which can be produced by these programs strongly exceeds the possibilities in our software. This, however, comes at the expense of a dedicated script language which needs to be learned in order to perform the required tasks.

A second alternative is the use of more general script languages like R [7] or MATLAB [8], for which packages for model data access or validation routines exist. An example for accessing model results from the R programming language is the package rNOMADS [9]. An example for a library of scripts for doing a numerous set of validation

plots is the open-source Python package VACUMM by Ifremer [10].

A third alternative to our Validator is the use of software which aims at assigning model scores as a measure of the goodness of the fit, rather than providing graphical evaluation. Software of this kind can be used for model intercomparison and assessing which model performs best compared to a predefined set of measurements. Examples for these tools are the NOAA model skill assessment software [11] or the ESMValTool [12], which is designed for obtaining performance metrics of Earth system models from the Coupled Model Intercomparison Project (CMIP6).

Implementation and architecture

The software provides a user interface generated by the R Shiny framework. In this framework, a Shiny server acts as a web server which creates an Asynchronous JavaScript and XML (AJAX) based user interface from R code. This user interface contains widgets, in our case drop-down boxes and radio buttons which allow the user to select

- a measurement dataset containing a pre-defined list of oceanographic stations, i.e., all stations of a particular measurement network,
- an oceanographic station from this dataset, either via a drop-down list or via a map,
- one to four models whose results will be compared to the observed data,
- a variable to compare between model and measurements, e.g., temperature,
- a plot type of their choice, and
- range settings (time, depth, season) depending on the type of plot.

When the user selects one option, the AJAX widgets communicate this choice to the Shiny server, where the chosen values are stored in so-called “reactive expressions”. These reactive expressions are passed as arguments to the plotting function, which will be explained in detail later. The result of the plotting function is the desired diagram, which is shown as output in the validator application. The concept of the “reactive expressions” is that an update of

their values, e.g., when the user selects a new oceanographic station, causes all functions depending on them to be re-evaluated. In this case, this means drawing a new plot.

The plotting function launches a separate process using the “mcparrallel” function in R (system package “parallel”). This keeps the user interface responsive during the plotting process. Should the user select different options (i.e., demand a different plot) before the plotting process is finished, it will be aborted and a new plotting process will be started instead.

There is an exception to this rule, which prevents the starting of too many plotting processes in a short time. E.g., this would happen if the user uses the up/down arrows at a numeric input field to change its value (e.g., from a depth of 5 m to 10 m by clicking the “up” arrow 5 times. The exception says that if a new plot is desired too quickly after the last one (in our implementation less than 0.8 seconds after the last plotting started), plotting will be delayed until no new plot has been requested during a period of these 0.8 seconds. So, the program waits until the user has finished their choice. **Figure 2** shows a flow chart which visualises the course of action when a plotting process is started.

The plotting function itself consists of five steps:

1. Checking the consistency of the input arguments
2. Loading model data
3. Loading measurement data
4. If required, discarding the months which are not part of the selected seasonal range
5. Calling the actual plot function, depending on the plot type

These steps will be outlined in the following subsections.

The same plotting function will also be executed when the user clicks a “save the plot as ...” button, in which case it will provide a PNG image file with publication-ready resolution for download.

Checking the consistency of input arguments

The plotting process only continues if the input passes some basic checks, e.g., if the input type is numeric or if upper range limits are below lower range limits.

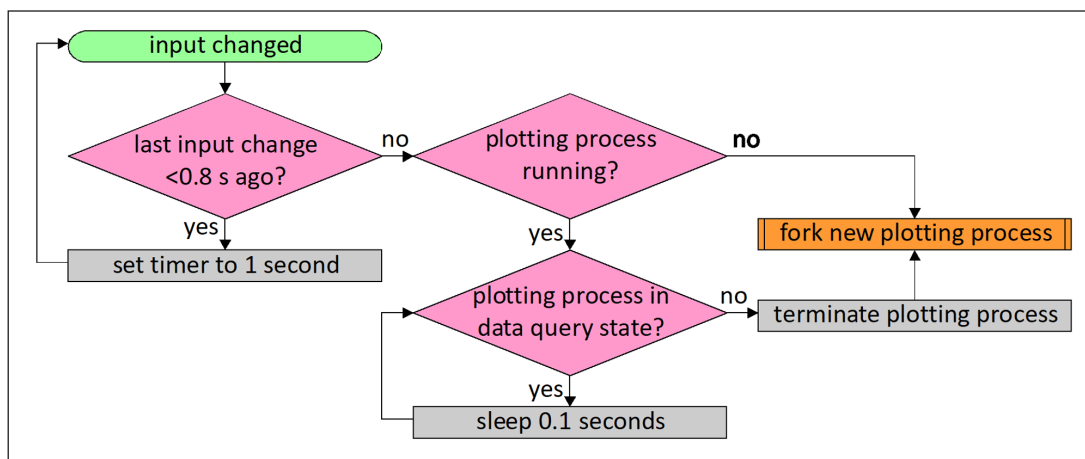


Figure 2: Flow chart diagram showing the course of action when a plotting process will be started as a separate process.

Loading model data

The required time and depth range of the selected variable is read from the input file and loaded into memory, for up to four selected models. Model data need to be provided as one NetCDF file per oceanographic station. For format details we refer to the user's guide.

Loading measurement data

Measurement data can be either loaded from an ASCII file or from a database. Both methods are illustrated in the example setup we provide. We use the open-source data base MariaDB, but any database which supports SQL queries should work. R connects to the data base via the package RMariaDB. Two tables need to be provided in the data base: The "measurements" table contains the observation values and the time and depth in which they were taken, while the "variables" table contains details about the observed variable, e.g., the name and unit. Details can be found in the user's guide. After the data are loaded from a data base, they are stored in a cache directory to minimise data base requests. In summary, this takes the following steps:

1. If data for the selected station and temporal range do not exist in the cache directory, perform a corresponding data base request and save the acquired data in ASCII format. During the processing of the data base request, the process cannot be aborted when the user selects a new plot, but its creation will be delayed.
2. Data will be loaded from the corresponding ASCII file in the cache directory.
3. The selected subrange of depth and time will be selected.

Select seasonal range

The user can select a range of months (e.g., Dec to Feb) which they want to include in the plot output. Both measurements and model data outside the given range will be discarded.

Calling the actual plot function, depending on the plot type

The actual plot function will create a plot showing both the model data (of up to four models) and observations. The "ggplot2" package is used. The first three plot types (time series, vertical profile, scatter plot) require no further explanation.

Choosing the option "Taylor diagram" will plot a Taylor diagram [1] displaying three measures of the model fit to the measurements at the same time: Pearson's correlation coefficient, root mean squared error and the standard deviation.

Choosing the option "seasonal/long term trend" will fit each one Generalised Additive Mixed Model (GAMM) [2] through the observations and every model's data. The data are modelled by fitting two smooth splines: One of them is periodic with a one-year period describing the seasonality of the measured variable, the second one describes the long-term behaviour. The random error

terms are represented by a first-order autoregressive model (continuous AR-1) [2]. Plotting the resulting splines allows to independently evaluate whether or not (a) the seasonality of the observed variable and (b) the (non-linear) long-term trend match between model(s) and observations.

Quality control

No additional testing has been carried out on the code apart from its regular practical application at our institute since 2017. As the functionality of the program is only to generate plots from existing data, it is itself of rather low complexity and therefore not very error prone.

The last two plot types require statistical methods to be applied to the data before plotting. These algorithms are of a higher complexity. Our program uses existing R functions and packages for this task, and we rely on their correctness, which is supported by the open-source availability of the code and the frequent use of the functions by a broad community.

Examples with sample input data are provided together with the model code, and the resulting figures can be compared to those shown in the user's guide to check whether the program is working properly.

(2) Availability**Operating system**

The software is written in R, but still requires a Linux environment due to the use of forking (function `mcparrallel`) in the parallel package. However, it can be run both locally or as a web server application. When running as a web server, e.g., on a virtual machine, clients can use the software from any operating system with a modern web browser (works at least with Mozilla Firefox 65.0 and Microsoft Edge 42.17134.1.0).

Programming language

Software is written in R using the R Shiny framework. It requires R version 3.5.1.

Additional system requirements

If live-extraction of measurements from a database is used, hard disk space in the order of 1 GB can be required for cached files, depending on the number of observations.

Dependencies

The software requires installation of the following R packages:

- ggplot2 2.2.1
- RNetCDF 1.9-1
- shiny 1.0.5
- plotrix 3.7-2
- RCurl 1.95-4.10
- mgcv 1.8-23

If measured data are live-extracted from a database, additional packages may be required for database access. If the example MySQL database is used, the package RMariaDB 1.0.6 is required.

List of contributors

All of the authors contributed to the creation or testing of the software.

Software location

Archive

Name: IOW institutional repository

Persistent identifier: doi: 10.12754/prog-2019-0001-01

Licence: MIT

Publisher: Hagen Radtke

Version published: 1.0.1

Date published: 16/05/19

Code repository

Name: GitHub

Identifier: <https://github.com/hagenradtke/validator>

Licence: MIT

Date published: 16/05/19

Language

The software is in English only.

(3) Reuse potential

The software in its present state can be used by any group of ocean modellers to validate their models against oceanographic observations. All that needs to be done is to save observed and modelled data in the required formats as specified in the user's guide. Users can also connect their own measurement database to the software if they provide an R subroutine to obtain the data from the database. Also, adding other plot types like, e.g., a Hovmöller diagram [13] is straightforward.

The program could also be extended to include measurements not taken at a fixed oceanographic station, e.g., satellite or glider data. This would, however, require substantial modifications. Anyone interested in extending the present software is encouraged to contact the corresponding author.

Acknowledgements

We would like to thank Steffen Bock from the IT department at our institute for setting up and maintaining the R Shiny server. We are grateful to the R Core team [7] and the developers of the R Shiny framework [14] as well as the other packages stated above [15–20] for providing their contributions as free software. The development of the Validator supports the Earth System Science for the Baltic Sea region of the Baltic Earth program (Earth System Science for the Baltic Sea region, see <http://www.baltic.earth>).

Competing Interests

The authors have no competing interests to declare.

References


1. **Taylor, K E** 2001 Summarizing multiple aspects of model performance in a single diagram. *Journal of Geophysical Research: Atmospheres*, 106(D7): 7183–7192. DOI: <https://doi.org/10.1029/2000JD900719>
2. **Pinheiro, J and Bates, D** 2000 *Mixed-Effects Models in S and S-PLUS*. New York: Springer-Verlag. DOI: <https://doi.org/10.1007/978-1-4419-0318-1>
3. **Chang, W, Cheng, J, Allaire, J J, Xie, Y and McPherson, J** 2017 shiny: Web Application Framework for R [online]. Available from: <https://CRAN.R-project.org/package=shiny> [Accessed 17 May 2018].
4. **Unidata** 2017 NetCDF [online]. Available from: <https://www.unidata.ucar.edu/software/netcdf/> [Accessed 17 May 2018].
5. <https://ferret.pmel.noaa.gov/Ferret/>.
6. <http://cola.gmu.edu/grads/>.
7. **R Core Team** 2018 R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.
8. <https://www.mathworks.com/products/matlab.html>.
9. **Bowman, D C and Lees, J M** 2015 Near Real Time Weather and Ocean Model Data Access with rNOMADS. *Comput. Geosci*, 78(C): 88–95. DOI: <https://doi.org/10.1016/j.cageo.2015.02.013>
10. **Raynaud, S and Charria, G** 2017 VACUMM – VACUMM v3.5.0 documentation [online]. Available from: <http://www.ifremer.fr/vacumm/> [Accessed 17 May 2018].
11. **Zhang, A, Hess, K W, Wei, E and Myers, E** 2006 Implementation of model skill assessment software for water level and current in tidal regions. *NOAA technical report NOS CS 24* [online]. Available from: <https://repository.library.noaa.gov/view/noaa/2204> [Accessed 17 May 2018].
12. **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, Senftleben, D, Sterl, A, Ulft, L H, van Walton, J, Wang, S and Williams, K D** 2016 ESMValTool (v1.0) – a community diagnostic and performance metrics tool for routine evaluation of Earth system models in CMIP. *Geoscientific Model Development*, 9: 1747–1802. DOI: <https://doi.org/10.5194/gmd-9-1747-2016>
13. **Hovmöller, E** 1949 The trough-and-ridge diagram. *Tellus*, 1(2): 62–66. DOI: <https://doi.org/10.3402/tellusa.v1i2.8498>
14. **Chang, W, Cheng, J, Allaire, J J, Xie, Y and McPherson, J** 2018 shiny: Web Application Framework for R. R package version 1.1.0. <https://CRAN.R-project.org/package=shiny>.
15. **Wickham, H** 2009 ggplot2: Elegant Graphics for Data Analysis. New York: Springer. DOI: <https://doi.org/10.1007/978-0-387-98141-3>
16. **Michna, P and Woods, M** 2017 RNetCDF: Interface to NetCDF Datasets. R package version 1.9-1. <https://CRAN.R-project.org/package=RNetCDF>.
17. **Lemon, J** 2006 Plotrix: a package in the red light district of R. *R-News*, 6(4): 8–12.

18. **Lang, D T** and **CRAN team** 2018 RCurl: General Network (HTTP/FTP/...) Client Interface for R. R package version 1.95-4.10. <https://CRAN.R-project.org/package=RCurl>.
19. **Wood, S N** 2011 Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)*, 73(1): 3–36 DOI: <https://doi.org/10.1111/j.1467-9868.2010.00749.x>
20. **Müller, K, Ooms, J, James, D, DeRoy, S, Wickham, H** and **Horner, J** 2018 RMariaDB: Database Interface and 'MariaDB' Driver. R package version 1.0.6. <https://CRAN.R-project.org/package=RMariaDB>.

How to cite this article: Radtke, H, Börgel, F, Brunnabend, S-E, Eggert, A, Kniebusch, M, Meier, H E M, Neumann, D, Neumann, T and Placke, M 2019 Validator – a Web-Based Interactive Tool for Validation of Ocean Models at Oceanographic Stations. *Journal of Open Research Software*, 7: 18. DOI: <https://doi.org/10.5334/jors.259>

Submitted: 14 February 2019 **Accepted:** 20 May 2019 **Published:** 12 June 2019

Copyright: © 2019 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

 *Journal of Open Research Software* is a peer-reviewed open access journal published by Ubiquity Press

OPEN ACCESS 