# Reviewer #1

**> Comments to Author**

**> Major comments**

> Hindcast vs forecast – A model with good statistical ability to hindcast data through a model might not forecast well (predict/project in future scenarios). So is there an option for looking at how well the model predicts uncalibrated data – i.e validation/cross validation. I think this would be a useful extension as it can help model users/developers understand how well their simulation parameters (as calibrated by hand/visualisation or with 'calib.assit') perform under forecasting.

We agree that a good hindcast model might not forecast well, but DYRESM-CAEDYM is a strongly process-based model with a long history of carefully defining the (narrow) ranges of calibrated parameters in the model. The 'calib_assit' function allows users to rerun the calibrated model with a new set of input data and then quantify the performance of the model with the 'objective_fun' function that calculates as many as five goodness-of-fit metrics. Given that the parameter ranges are small, the model performs well, and the model uses a strongly process-based approach, the model provides an excellent basis for supporting forecasting.

> I enjoyed having a play with the R package, but I was disappointed that I could not use the 'calib.assit' function. I think that you need to develop a way to make this usable after package installation. Perhaps you could port DYRESM-CAEDYM to R. Or enable the calibration to run off a DYRESM-CAEDYM output. It does seem a bit counter intuitive that you are providing an open source tool (dycdtools) that can't actually be run unless you have a licence for DYRESM-CAEDYM (although it appears to be free for academic work). Perhaps an easy workaround is to include installation instructions for DYRESM-CAEDYM or automatically install DYRESM-CAEDYM as part of the package build. The package could then provide a toy/small example of DYRESM-CAEDYM and how you can run the 'calib.assit' function.

> Page 5. I would like to have had a look at how the calibration function works. I think at a bare minimum the authors need to provide a template output or adapt the code so it can run without having to use the DYRESM-CAEDYM executables. Perhaps a port of DYRESM-CAEDYM into R?

We have negotiated with the licence holder of DYRESM-CAEDYM to make the executables of the model freely available. We have now added a URL (https://github.com/SongyanYu/ExampleData_dycdtools) in the Data and software availability section for readers to access the executables and the example data used to support the case study. Instructions on how to use the model executables and the example data are provided in the 'README' file at the URL. The updated code-to-reproduce now has an example showing how to apply the 'calib.assist' function to automatically calibrate the model.

> I really liked the plotting functions. Really nice addition, I think, if you include some of the skill-goodness-of-fit statistics alongside these plots (as shown in the fig below). Then that would be a really useful contribution.

> Fig from Olsen, E., Fay, G., Gaichas, S., Gamble, R., Lucey, S. and Link, J.S., 2016. Ecosystem model skill assessment. Yes we can!. PloS One, 11(1), p.e0146467. Which plots observed vs skill metrics.

> (*the provided Fig not shown in the response letter*)

> Fig. 5 & 6. It might be nice to add in a row or column that represents the RMSE, correlation or some other metric (see the inserted figure from Olsen et al). For example in Fig.5 it is obvious that 2002-07-25 observation fit very well to the model, but it is less clear with the other dates/data.

We are glad that the plotting functions were well received. We assume that the reviewer was suggesting adding values of goodness-of-fit statistics to the figures, although it should be noted that values of model performance metrics can be calculated with the 'objective_fun' function in our package.

For Fig. 6, we have added the key model performance metric e.g., RMSE as suggested by the reviewer. We are hesitant to do the same to Fig.5 as it only shows comparison between simulation and observation for one day.

**Minor comments**

> General Lake Model (GLM); confusing R model acronym – just a comment.

Agreed.

> Page 1 – third para: Sounds a bit like a regularisation problem.

We assume that the reviewer was concerned about potential overfitting of DYRESM-CAEDYM in these reviewed literatures due to the calibration of more than 15 parameters. This might be the case, but it is worth noting that DYRESM-CAEDYM is a strongly process-based model, so the calibrated parameters have an intended physical representation but may vary within reasonably discrete bounds. We consider these physically meaningful parameters limit the potential for overfitting.

> Page 2: "For example, users are not able to compare observations and simulations in the same figure using the current GUI in DYRESM, undermining the ability of visual checks to assess goodness-of-fit to inform parameter calibration. " – I think this is a valuable addition to interpreting simulation runs. But it makes me wonder are there other skill tests that can be done to see how well the model does?

Common ways to identify model performance include 1) calculating suitable goodness-of-fit statistics to compare the simulation results to the observations, and 2) visually checking the patterns of the simulation and the observation. Our package is useful for both ways as the 'objective_fun' function can be used to calculate several model performance metrics and the plotting functions are good for visual checking.

> Fig. 3. Spell out what NSE is. Is it possible to get more contrast in the colour of each panel, maybe using facet_wrap and make the scale free?

The full name of NSE was provided prior to Fig. 3 (in the paragraph underneath Table 1), but we have now also provided the full name of NSE in the caption of Fig. 3.

We tried 'facet_wrap' and a free scale, but the resulting figure (Figure R1) was similar to the original one (Figure R2).

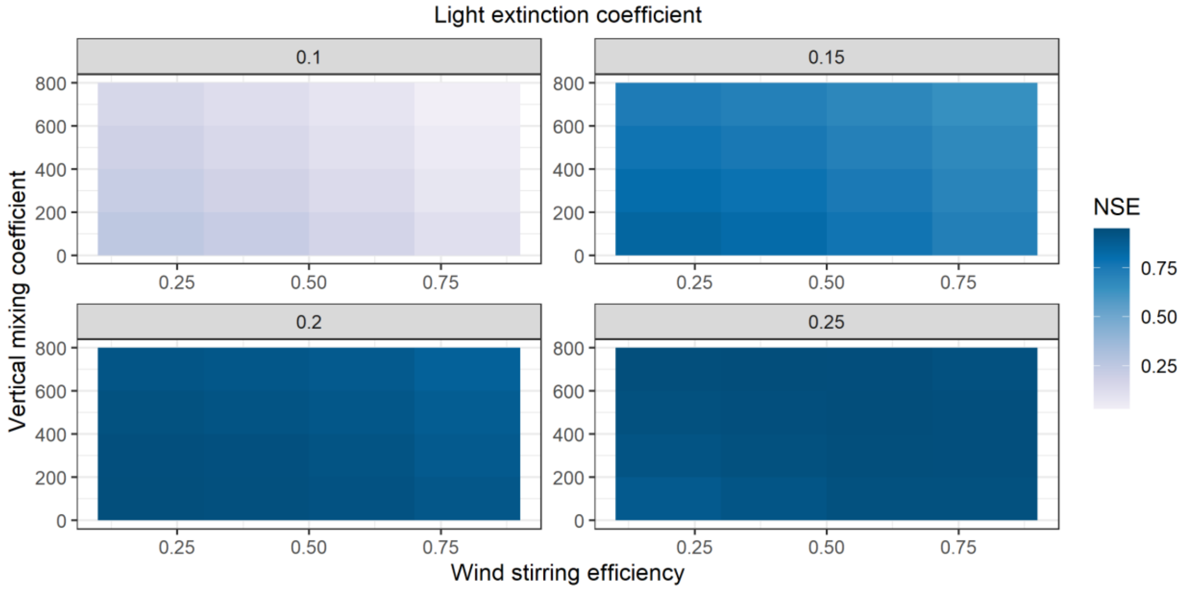

Figure R1. The 'facet_wrap' version of original Fig. 3 in the manuscript.

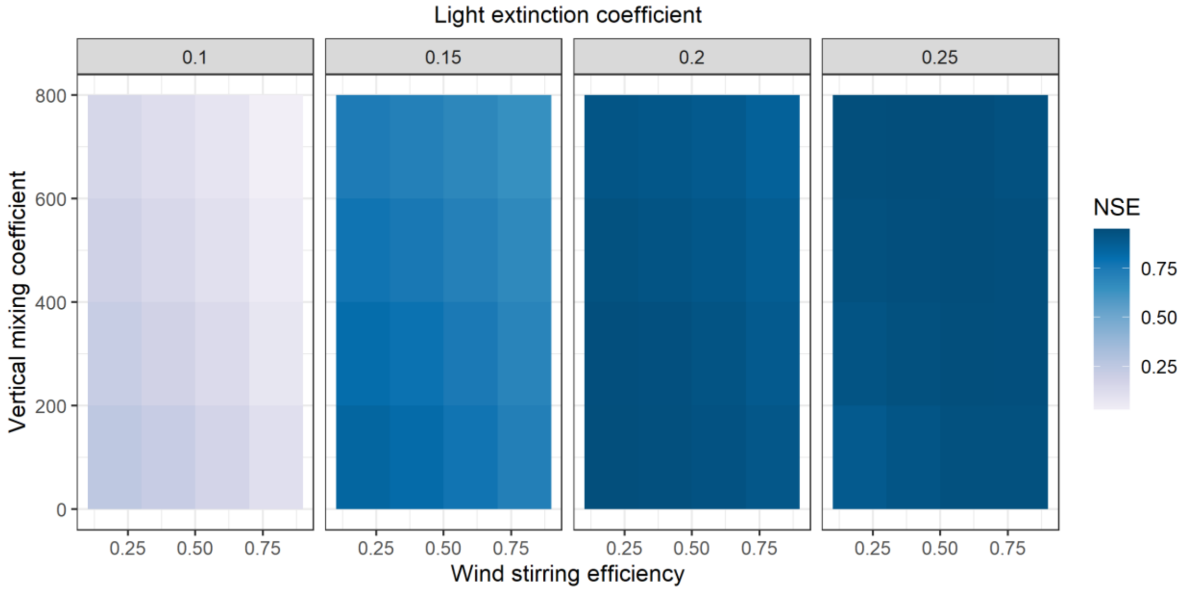

Figure R2. Original Fig. 3 in the manuscript.

> Fig.7. Binning might be a way to show which depth regions/data are easier/harder to calibrate.

Our package has two other plotting functions (‘plot_prof’ and ‘plot_ts’) to identify which depth regions are easier or harder to calibrate (see Fig. 5 & 6 in the revised manuscript). Fig. 7 was intended to show a simulated profile of a variable (e.g., temperature) with the observed variable, complementing Figs 5 & 6 where comparisons were made for each day or each depth.

> Page 11. It seems unlikely to me that anyone would run 10^3 simulations, but rather would choose a subset of these possibilities, say a subset of 10,000 combinations, if being pragmatic and then would start to tune the model based on those simulations.

We agree. The example mentioned was meant to demonstrate how to balance automatic calibration and manual calibration. The process of selecting a subset of those possibilities can be part of manual calibration, while running the model on the selected possibilities can be facilitated by automatic calibration.

## Reviewer #2

> * Overview: Is the article important to R the community? Is the approach sensible? Are technology and methods up-to-date?

> This article and accompanying R package provide a convenient set of functions to facilitate parameter calibration and results visualizations for DYRESM-CAEDYM. The article does a nice job of providing the rationale for the package and the potential limitations of automatic versus manual calibrations. The case example and interpretation are clear. As someone who has never used DYRESM-CAEDYM, I was able to follow along well.

Thank you for the positive comments.

> * Article: Is anything unclear? Is anything missing? Are the examples helpful? Is there sufficient background, including alternative approaches in R?

> Article should include a link to the package repository and provide instructions on submitting issues.

We have added a new section ‘Data and software availability’, in which a link to the package repository and instructions on submitting issues have been provided.

> Should there be an arrow in Figure 1 connecting the bottom shaded block to the flow chart?

Yes, agreed. We have now updated Fig. 1 as suggested.

> The ggplot heatmap you generated is not part of the package—I understand the rationale for not wanting to fully automate the process of parameter selection, but it might be nice to provide one or two convenience functions to assist users in exploring the calibration results. Something like a generalizable figure or a formatted table would be a nice way to connect the workflow you’ve laid out.

In trying to have a general figure we had to consider: 1) that the number of parameters to be calibrated varies (typically from 2 to 6 based depending on the case), while it is very difficult to display 4 or more dimensional information, and the heatmap presented in the manuscript only showed 3 dimensions; 2) that parameter combinations to be tested can be either exhaustive or random (depending on the setting of the argument ‘combination’ of the ‘calib.assist’ function), which makes it difficult to develop a generalizable figure.

Instead, the calibration outputs are currently produced as a formatted table (a csv file), and users can easily use the ‘Conditional Formatting’ function in MS Excel to visualise calibration results and select suitable parameter values. See Table R1 for an example.

**Table R1**. Calibration results produced by the 'calib.assist' function with conditional formatting in MS Excel to colour code Nash-Sutcliffe Efficiency (NSE) values. Higher values are in shades of green, while lower values are in shades of red. Three parameters are tested: wse – wind stirring efficiency, vmc – vertical mixing coefficient, and lec – light extinction coefficient (1/m).

| wse | Vmc | lec | NSE |
|---|---|---|---|
| 0.2 | 100 | 0.1 | 0.82 |
| 0.4 | 100 | 0.1 | 0.78 |
| 0.6 | 100 | 0.1 | 0.73 |
| 0.8 | 100 | 0.1 | 0.67 |
| 0.2 | 300 | 0.1 | 0.78 |
| 0.4 | 300 | 0.1 | 0.75 |
| 0.6 | 300 | 0.1 | 0.70 |
| 0.8 | 300 | 0.1 | 0.64 |
| 0.2 | 500 | 0.1 | 0.75 |
| 0.4 | 500 | 0.1 | 0.70 |
| 0.6 | 500 | 0.1 | 0.66 |
| 0.8 | 500 | 0.1 | 0.61 |
| 0.2 | 700 | 0.1 | 0.70 |
| 0.4 | 700 | 0.1 | 0.67 |
| 0.6 | 700 | 0.1 | 0.62 |
| 0.8 | 700 | 0.1 | 0.58 |

> 'All example data are provided in a public data repository for users to familiarise themselves with model runs, calibration and visualisation.' Please provide a link to this repository. Ideally, it should be part of the package as a vignette.

The link to the example data repository has now been provided in the manuscript and an instruction on how to apply the package to the example data was provided in the 'README' file of the example data.

Our understanding of a vignette is that it is a long-form guide to the package, quite similar to the manuscript we presented here. If the manuscript is published in R Journal we will provide its DOI in the example data repository. If the manuscript is not accepted for publication, we will convert the manuscript to a vignette (following the workflow of making a vignette suggested by Hadley Wickham's R packages) and publish it in Songyan's personal website (https://songyanyu.github.io) for users to access.

> "In this simulation, both the wind stirring efficiency and vertical mixing coefficient are in the parameter (.par) file, while the light extinction coefficient is in the configuration (.cfg) file." These files are not in the code 'code-to-reproduce' directory. The directory only includes reproducible examples for figure generation. There should be an example of the calibration function that

someone with access to DYRESM-CAEDYM would be able to use to reproduce your methods. Again, this should be a vignette in the package.

The example data, including the relevant configuration files, have now been provided. The calibration assistant function (‘calib.assist’) can now work with the example data.

> You found the light extinction coefficient of 0.25 m-1 to be optimal in your calibration. As this is the maximum value in the range of values you tested, should you explore higher values? Or is there a physical or ecological reason to cap the calibration at this value?

The tested coefficient values fall within the normal range of light extinction coefficient for a clear lake like Okareka (i.e., the study lake in the manuscript), with 0.25 $m^{-1}$ commonly regarded as the upper limit. Where measured values were available, the value in the example could be increased.

> The package needs improvement to be useable for a broader audience. Below are some issues I’ve noted, but I suggest reading through Hadley Wickham’s R packages. You may also want to check out best practices from rOpenSci. Please also run goodpractice::gp() on your package directory to see a number of suggestions for improvement. In general, think about how to increase flexibility and how to lower the barrier of entry to new users.

> The package needs better documentation, including the readme, package help page (https://r-pkgs.org/man.html#man-packages), and vignettes. A lot if this can be taken from the paper

We thank the reviewer for the insightful suggestions. We have thoroughly checked the package and made some substantial changes to the package functions to improve flexibility (version update from 0.3.1 to 0.4.2). As mentioned previously, we follow Hadley Wickham’s R packages to make a vignette for our package based on the case study presented in the manuscript.

We have run goodpractice::gp() and found that all the given suggestions could be classified into the following groups:

1) Write unit test
2) Avoid long code lines
3) Avoid calling setwd()
4) Avoid 1:nrow(…)
5) Avoid ‘T’ and ‘F’, always use ‘TRUE’ and ‘FALSE’.

These suggestions are quite similar to some of the following comments from the reviewer. We followed all the suggestions except ‘avoid calling setwd()’ because we believe setwd() is the only solution in our case, but we used ‘on.exit()’ in the function to restore the working directory.

We have also made significant changes to improve the readme (see https://github.com/SongyanYu/dycdtools for detail) and package help page.

> Test coverage is only 1%. Not every line of code needs to be covered but you should aim for more testing (https://r-pkgs.org/man.html#man-packages)

We have now added more unit tests for the package and the text coverage increased to 19%.

> The GitHub repo package name should be lowercase to match cran. Some functions are case sensitive to this.

The GitHub repo package name has now been changed to 'dycdtools' as suggested.

> Some function names have periods and others underscores. I recommend being consistent, and underscores are generally preferred. Check out these packages to help with code styling: https://r-pkgs.ord/r.html#code-style

We have now replaced period with underscore in function names.

> Several your functions have defaults that shouldn't be there, such as file names and dates. Defaults should only be specified if they are generalizable. There should be a good reason for defaults (and reasons should be documented in function help), and someone using the package should be able to run functions using the defaults without error.

Thank you for the suggestion. We have now deleted all unnecessary defaults.

> The plots are very nice, but some are ggplot and others base. It would be best to be consistent, ideally all in ggplot.

The contour plot is achieved through the base function of 'filled.contour', while the other three visualisation functions use ggplot. We tried to use ggplot to make the contour plot but have not improved upon the current graphical output and therefore have not changed the current function.

> Plots should be returned as an object to give the user the option to adapt them. For example, someone should be able to run p <- plot_cont_com(…) and then be able to modify the object p.

We agree. All visualisation functions have now been revised to return an object that can be modified by users.

> Avoid setwd() whenever possible. Perhaps the here package would help.

The here package works for a file project but does not suit the situation where setwd() is used in the package. We found setwd() is the only solution in our case and we used 'on.exit()' to restore the working directory.

> What's the difference between 'Fun_Contour-plot_sim.R' and 'Fun_Contour-plot_sim_obs.R'?

The 'Fun_Contour-plot_sim.R' only visualises model simulation results, while the 'Fun_Contour-plot_sim_obs.R' visualises both model simulation and local observation in the same figure (Fig. 4)

> Some specific comments on Yu_dycdtools.R below

> > Lines 18-19: when I try running ext.output() with another variable I get this error. Maybe because DO wasn't part of the simulation? If that's the case, the function should return an informative error.
>
> Yes, DO was not part of the simulation, but we have now revised the function to return an informative error – reporting which variables get extracted and which variables are not in the model outputs.
>
> > Lines 21-24: why not include this as part of ext.output()? Or if it's not necessary, as it seems like the downstream functions accept matrices, why include this in the demo? You can call all the individual matrices from the var.values list.
>
> Lines 21-24 convert each element of the list output from ext.output() to a data frame, and should not be included in ext.output as these generated data frames still have to be returned as a list.
>
> We agree that these lines are not essential and have thus deleted from the demo.
>
> > Line 37: it's best practice to subset with names instead of index (more readable, less prone to error)
>
> We agree. In the revised demo we replaced index with names in subsetting.
>
> > Lines 95-107: Returns a warning about missing points being removed. You may want to address missing values before plotting.
>
> The removal of missing points is expected for running Lines 95-107. The simulation results ('Sim' column of below table) are daily while the observation results ('Obs' column in the table below) are roughly monthly (see below table), so when plotting 'Sim' and 'Obs' in the

same figure as running Lines 95-107, NA values in the ‘Obs’ column will be removed, incurring the warning.

| Depth | Date | Sim | Obs |
|---|---|---|---|
| 0 | 2022-01-23 | 21.41 | 21.2 |
| 0 | 2022-01-24 | 21.71 | NA |
| 0 | 2022-01-25 | 22.04 | NA |
| 0 | 2022-01-26 | 21.65 | NA |
| … | … | … | … |

> calib.assist()

> Can you modify object.fun() to be able to return multiple measures at once? That would help clean up some of the code in calib.assist().

We agree. We have now revised the object.fun() and cleaned up calib.assist().

> Why do the input calibration parameters have to be a csv file? Can you allow the user to enter a vector or dataframe? More flexibility will increase usability. It should also be clearer what the inputs should look like (and maybe less rigid).

In the revised package, users are now allowed to either provide a csv file or a data frame for the calib.assist function to find essential information for model calibration.