dycdtools: an R Package for Assisting Calibration and Visualising Outputs of an Aquatic Ecosystem Model

by Songyan Yu, Christopher G. McBride, Marieke A. Frassl, Matthew R. Hipsey and David P. Hamilton

Abstract The high complexity of aquatic ecosystem models (AEMs) necessitates a large number of parameters that need calibration, and visualisation of their multifaceted and multilayered simulation results is necessary for effective communication. Here we present an R package "dycdtools" that contains calibration and post-processing tools for a widely applied aquatic ecosystem model (DYRESM-CAEDYM). The calibration assistant function within the package automatically tests a large number of combinations of parameter values and returns corresponding values for goodness-of-fit, allowing users to narrow parameter ranges or optimise parameter values. The post-processing functions enable users to visualise modelling outputs in four ways: as contours, profiles, time series, and scatterplots. The "dycdtools" package is the first open-source calibration and post-processing tool for DYRESM-CAEDYM, and can also be adjusted for other AEMs with a similar structure. This package is useful to reduce the calibration burden for users and to effectively communicate model results with a broader community.

Introduction

Aquatic ecosystem models are important tools to understand the structure and function of aquatic ecosystems, fill observation gaps, and support scientific management of water quality of inland waters (Jakeman et al., 2006). Processed-based aquatic ecosystem models represent the major physical, chemical and biological processes as a series of mathematical equations, offering opportunities for exploratory or predictive management applications (Frassl et al., 2019; Özkundakci et al., 2011). These models are increasingly used to simulate observed data and forecast changes that may occur under scenarios in a changing climate (Elliott, 2012; Nielsen et al., 2017; Rousso et al., 2020). A typical example of such models is DYRESM-CAEDYM (DYnamic REservoir Simulation Model – Computational Aquatic Ecosystem Dynamics Model), which has been developed from a one-dimensional coupled hydrodynamic-ecological lake model (Hamilton and Schladow, 1997) and has been widely used to simulate water quality and ecosystem processes for a large number of lakes and reservoirs (Cui et al., 2016; Lewis et al., 2004; Takkouk and Casamitjana, 2016).

Aquatic ecosystem models, particularly those with a large number of physical and biogeochemical parameters, need calibration before they can be used for reliable simulation outputs (Luo et al., 2018). For example, water quality simulations with DYRESM-CAEDYM usually involve calibration of 20-30 parameters (e.g. 18 parameters in Luo et al. (2018); 28 parameters in Schladow and Hamilton (1997)). Parameter values are usually chosen via trial and error (Takkouk and Casamitjana, 2016), parameter ranges in the literature and modeller experience (Lehmann and Hamilton, 2018; Robson and Hamilton, 2004), laboratory experimental data (Robson and Hamilton, 2003), or small-scale field measurements (Burger et al., 2008). The calibration process of stepwise iterative manual adjustment of parameters is labour intensive and time consuming, partly due to the interdependent nature by which state variables respond to individual parameter adjustments (Lehmann and Hamilton, 2018). The success of calibration (i.e., ability to accurately reproduce a set of observed data) strongly relies on the skill and experience of the modeller, and the calibration process is often considered to be complete when the difference between observations and simulations is within an acceptable level of statistical compliance (Hipsey et al., 2020).

Numerous development efforts are underway to provide assistance in the calibration process. A promising approach is to take advantage of increasing computational power to overcome some shortcomings of traditional manual calibration methods and to automatically trial a large number of possible combinations of parameters in silico. For example, a model independent parameter estimator tool, PEST (Parameter ESTimation), is able to carry out nonlinear parameter estimation for most environmental simulation models (Doherty et al., 1994; Dohorty, 2018). PEST has been widely used for parameter calibration and to quantify errors in surface water and groundwater models (Christensen and Doherty, 2008; Gallagher and Doherty, 2007; White et al., 2014), but has rarely been applied in numerical lake models. In addition, multiple calibration assistant tools have been developed for specific lake models, but many are not open-source or freely accessible to model users, limiting their potential application to a broad community. Developing tools in a more open way to facilitate aquatic ecosystem model calibration and output processing can facilitate accessibility and create user-friendly tools, as evidenced in the lake modelling community (Frassl et al., 2019). Such tools have been developed for a range of lake models, including the Freshwater Lake Model (FLake), General Lake Model (GLM), General Ocean Turbulence Model (GOTM) and Simstrat in the **LakeEnsembleR** package (`https://github.com/aemon-j/LakeEnsemblR`). However, currently no such open-source tools have been developed to assist calibration for the widely applied lake model, DYRESM-CAEDYM.

Visualisation is fundamental to studying complex subject matter and supporting the whole information pipeline, from acquiring and exploring data and analysing models, to visual analytics, through to storytelling to communicate background information, results, and conclusions (McInerny et al., 2014). Simulation results from lake models are often multilayered across different depths, multifaceted across different modelling variables, and dynamic over a simulation period, posing difficulties in effective presentation for many end users. Although some lake models have a default Graphical User Interface (GUI) to visualise modelling results for different layers and variables, the built-in plotting functions in these GUIs are often limited and inflexible. 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.

To bridge this gap, we demonstrate in this study the development and application of an R package **dycdtools** for assisting calibration of DYRESM-CAEDYM and post-processing of simulation outputs. A calibration assistant tool has previously been developed for DYRESM-CAEDYM (Luo et al., 2018), but was coded in Fortran language and was not stored on an open platform, such as GitHub, to support further development. By contrast, the **dycdtools** package designed to support additional development can be freely downloaded and used by DYRESM-CAEDYM users. The package also includes advanced tools to support visual assessments and enable effective communication of modelling outcomes. This package not only works with DYRESM-CAEDYM but can be adjusted for other aquatic ecological models with a similar structure, such as the General Lake Model (Hipsey et al., 2019).

In the following, we first describe briefly the structure of DYRESM-CAEDYM and the shortcomings in its application that were the primary driver for development of **dycdtools**. We also present in detail the structure of the R package, including a demonstration of the package through application to a monomictic lake. We conclude with the development and application of calibration assistant and visualisation tools for aquatic ecosystem models.

DYRESM-CAEDYM

DYRESM-CAEDYM consists of a hydrodynamic model DYRESM coupled with an ecological model CAEDYM. DYRESM is one-dimensional and resolves vertical distributions of temperature, salinity and density in lakes and reservoirs based on a dynamic Lagrangian layer structure, which simulates the lake as horizontally uniform layers that expand and contract in response to heat, mass and momentum exchanges (Gal et al., 2003). CAEDYM is an aquatic ecological model designed to simulate processes affecting carbon, nitrogen, phosphorus, silicon and dissolved oxygen cycles, including several size classes of inorganic suspended solids, and phytoplankton dynamics.

DYRESM-CAEDYM takes a wide variety of forcing and morphometry data and configuration files as input to represent the major aquatic physical and biogeochemical processes. The required forcing and morphometry data include meteorology, morphometry, inflows and outflows. Configuration files allow users to adjust sensitive parameter values for a model simulation. It is the configuration files that the auto-calibration function in **dycdtools** deals with, assuming that the forcing (e.g., meteorology, inflows and outflows) and fixed data (morphometry) have been obtained from external sources (Figure 1). The scientific foundations of DYRESM-CAEDYM and information on data input preparation can be found in its scientific manual (Imerito, 2007) and in Robson and Hamilton (2004), Romero et al. (2004), Luo et al. (2018).

DYRESM-CAEDYM is supported by a proprietary GUI, but the GUI provides limited means to visualise simulation results. So far, only time series and contour plots are built in the GUI, and they require users to take complex steps to plot the observation and simulation in the same figure. The dycdtools package provides four different types of visualisation plots that could be used to complement the default GUI and to examine model simulation results in more detail (Figure 1).

Structure of the dycdtools package

There are two main function categories in the **dycdtools** package: calibration assistant and post-processing (Table 1). After the forcing and morphometry input data to DYRESM-CAEDYM have been prepared, the calibration assistant function can be used to 1) choose the optimal set of parameter

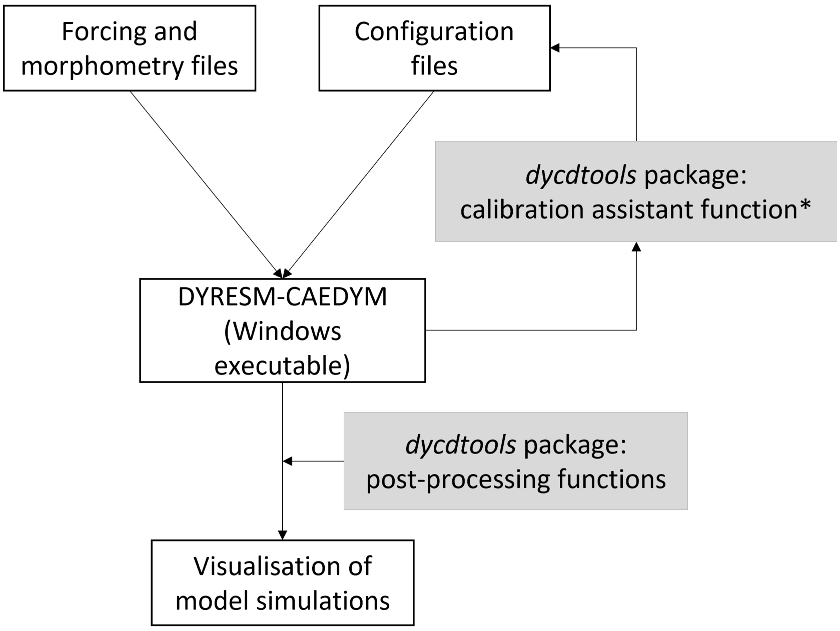
Forcing and morphometry files
Configuration files
dycdtools package: calibration assistant function*
DYRESM-CAEDYM (Windows executable)
dycdtools package: post-processing functions
Visualisation of model simulations

Figure 1: DYRESM-CAEDYM simulation process. The calibration assistant function in the **dycdtools** package deals with the model configuration files (the top shading rectangle), while the post-processing functions can be used to visualise model outputs (the bottom shading rectangle). *The calibration assistant function works only on the Windows platform, as it needs to call the Windows executable of DYRESM-CARDYM to run model simulations, while the post-processing functions are cross-platform.

values for a model simulation application to a lake, or 2) conduct sensitivity analysis of parameters of interest including using a Monte-Carlo parameter perturbation function. Post-processing includes a suite of R functions to visualise DYRESM-CAEDYM output in multiple ways, such as contours, profiles, time series, and scatterplots, for different simulation and observation depths (Table 1).

Calibration assistant function

The calibration assistant function carries out simulations with a large number of possible combinations of parameter values that users regard as potentially suitable for their model calibration, and calculates the values of nominated objective functions (i.e. statistical measures of goodness-of-fit) for each combination. Based on the calculated objective function values, users can determine the optimal set(s) of parameter values or narrow the ranges of possible parameter values. Note that the calibration assistant function does not have the source codes of DYRESM-CAEDYM, but calls the Windows executable of the lake model to run separate simulations with various combinations of parameter values. For this reason, the calibration assistant function can only work on the Windows platform.

The calibration assistant function first requires that users prepare a list of sensitive parameters for their simulation (Figure 2). The selection of sensitive parameters can be made by referring to previous literature (Bruce et al., 2006; Robson and Hamilton, 2004; Schladow and Hamilton, 1997) or through expert opinion (Lehmann and Hamilton, 2018) as well as by using a parameter sensitivity analysis, which is a part of the calibration assistant function (Ofir et al., 2017). The list of selected sensitive parameters should be prepared as comma-separated values (.csv) in a file input to the function. The potential value range and the preferred number of values within the range for each parameter also need to be defined by users in the csv file. To facilitate this process, an example csv file of many common parameters has been included in the package data set, so that users can follow the given format and adjust values to their specific case study.

Based on the user-defined potential value range and the preferred number of values, the calibration assistant function forms all possible combinations of parameter values (Figure 2). For example, for three parameters that have three different values, there are 3^3=27 possible combinations. Users can choose to make the function try every possible combination for model calibration or only a subset of randomly selected combinations. To help users deal with complicated situations where much more parameter combinations are involved, the calibration assistant function is able to parallelise DYRESM-CAEDYM simulations with multiple cores, which could substantially reduce model running time.

Table 1: List of functions in the dycdtools package for calibration assistant and post-processing and their descriptions.

Function name	Description	Categories
calib.assist	Apply different combinations of selected parameter values and output corresponding values of goodness-of-fit by calculating objective functions. Users can choose the optimal set of parameter values or refine potential parameter value ranges based on the calculated values of goodness-of-fit.	calibration assistant
plot_cont	Contour plot (heat map) showing a depth- and time-resolved matrix of biogeochemical variables.	post-processing
plot_prof	Profile graph of simulations over time (only for dates when observations are available), with variable values shown in the x axis and depth in the y axis.	post-processing
plot_ts	Time series plot of simulations for one or multiple specific depths, with time as the x axis and variable values as the y axis.	post-processing
plot_scatter	Scatter plot with observations as the x axis and simulations as the y axis. Points in the scatter plot are colour coded by depth.	post-processing

For each of the combinations, the calibration assistant function makes changes to configuration files and then calls the DYRESM-CAEDYM model to run simulations (Figure 2). After each model run, the function extracts model outputs to compare against the observations and calculates objective function values. Five commonly used objective functions are currently available in the package: Nash-Sutcliffe Efficiency coefficient (NSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Relative Absolute Error (RAE), and Pearson's r (Pearson), with more to be added in an update of the package by referring to the inventory of different statistical measures in Bennett et al. (2013). The output of the calibration assistant function is a table that outlines all combinations of parameters that were used and the corresponding objective function values. Based on the output, users can determine the optimal set(s) of parameter values (e.g., the set that gives the highest NSE value or highest weighted average values for NSE and RMSE) or narrow the range of suitable parameter values.

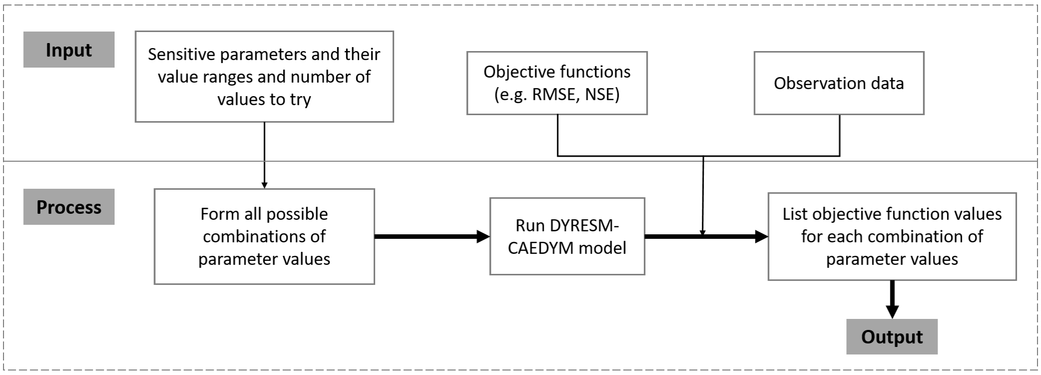
Input
Sensitive parameters and their value ranges and number of values to try
Objective functions (e.g. RMSE, NSE)
Observation data
Process
Form all possible combinations of parameter values
Run DYRESM-CAEDYM model
List objective function values for each combination of parameter values
Output

Figure 2: Structure of the calibration assistant function.

The calibration assistant function can also be used for parameter sensitivity analysis by automatically modifying parameter values. Two commonly used methods for sensitivity analysis, One At a Time (OAT) and All At a Time (AAT), can be carried out through this function by setting the "combination" argument as "all" and preparing specifically designed parameter combinations. An OAT analysis changes a single parameter at a time and is suitable for local sensitivity analysis, while ATT modifies multiple parameters simultaneously and is suitable for global sensitivity analysis, notably when there are nonlinear relationships between parameter and model outputs. Increasing numbers of environmental modelling studies report model uncertainty for a variety of purposes, including assessment of model errors, model calibration communication and diagnostic evaluation (Couture et al., 2018; Dietzel and Reichert, 2014; Pianosi et al., 2016). This function is also flexible, allowing users to try a subset of randomly selected parameter combinations (e.g. Monte-Carlo parameter perturbation) or a carefully selected combination by setting the "combination" argument as "random".

Post-processing functions

Post-processing functions provide multiple ways to visualise DYRESM-CAEDYM outputs, as follows:

- Function "plot_cont" displays a heat map of variable values with depth within the water column and over time. This visualisation is particularly suitable for displaying temporal and depth dynamics of a variable at one lake site.
- Function "plot_prof" shows vertical profiles of the simulation and corresponding observations, for all dates where observations are available.
- Function "plot_ts" plots simulated values and observations for a specified variable and depth over time. It can be used to compare temporal changes of a variable for simulations and observations at specific depths.
- Function "plot_scatter" shows observations against simulated values for corresponding time and depth, with a colour scale representing measured depth. It can be used to demonstrate visually the goodness of fit for a variable across the water column.

All four types of graphs can be used to compare simulations and observations in the same figure. Examples of each type of graph are shown in the following section.

Minimal case study example

Lake Okareka is a medium size lake in the Bay of Plenty region of North Island, New Zealand. It has a surface area of 3.46 km^2, a land catchment area of 16.7 km^2 and a maximum depth of 33.5 meters. Water drains from the lake via an outlet canal towards a large downstream lake, Tarawera. The **dycdtools** package was used to calibrate three parameters for the lake temperature simulation and visualise the temperature simulation against observed data with the four aforementioned plotting functions. The three calibrated parameters were wind stirring efficiency, vertical mixing coefficient, and light extinction coefficient, as they have been found to be most sensitive in temperature simulations in other lake systems (Weinberger and Vetter, 2012).

The simulations were conducted with forcing inputs (e.g., meteorology, inflows and outflow) at a daily time step over the period of 2002-01-23 to 2016-12-31. The meteorology and outflow data were respectively collected from the New Zealand National Climate Database (`https://cliflo.niwa.co.nz`) and an environmental monitoring website (`https://www.boprc.govt.nz/environment/maps-and-data/environmental-data`). Discharge output from a SWAT model application (unpublished) was used as catchment surface inflow to the lake. All example data are provided in a public data repository (`https://github.com/SongyanYu/ExampleData_dycdtools`) for users to familiarise themselves with model runs, calibration and visualisation.

The calibration assistant function deals with the configuration files. 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. We assigned a range and a preferred number of values to try for each of the three parameters to form a sequence of possible values (Table 2).

Table 2: Three parameters that were calibrated with assigned value ranges and preferred number of values to try. The columns of "Input file" and "Line No." describe the line of the input file where the parameter is located.

Parameter	Unit	Min	Max	N_values	Input file	Line No.
Wind stirring efficiency	-	0.2	0.8	4	Okareka.par	12
Vertical mixing coefficient	-	100	700	4	Okareka.par	15
Light extinction coefficient	m^{-1}	0.1	0.25	4	Okareka.cfg	7

The calibration assistant function tried all combinations of the three parameter values and calculated the objective function NSE for each model run. We chose the combination of parameter values that gave the highest NSE values as the optimal set and then used the post-processing tools to plot simulations against observations in four different ways. The R code for using the calibration assistant function in this example is as follows:

```
# Assisting calibration of DYRESM-CAEDYM using the dycdtools package
 library(dycdtools)
 calib.assist(cal.para = "calibration_data/Calibration_parameters.csv",
              combination = "all",
              model.var = "TEMP",
              obs.data = "calibration_data/Obs_data_template.csv",
              objective.function = "NSE",
              start.date = "2002-01-23",
              end.date = "2016-12-31",
              dycd.wd = "calibration_data/DYRESM_CAEDYM_Lake-Okareka/",
              dycd.output = "calibration_data/DYRESM_CAEDYM_Lake-Okareka/DYsim.nc",
              file.name = "calibration_data/Calibration_outputs.csv",
              write.out = TRUE,
              parallel = TRUE,
              verbose = TRUE)
```

Given each parameter had four assigned values, the calibration assistant function called DYRESM-CAEDYM to run a total of $4^3 = 64$ combinations for the three parameters. For each model run, the objective function NSE was calculated for temperature simulations. A heat map is a good way to visualise the variations in these NSE values under different combinations of parameter values. An example of R code for producing a heat map of the auto-calibration outcome is as follows:

```
# Read in model calibration results
calibration <- read.csv("calibration_data/Calibration_outputs.csv")

# Heat map
library(ggplot2)
ggplot(calibration, aes(x = wse,y = vmc,fill = NSE.TEMP)) +
       geom_tile() +
       scale_fill_distiller(palette = "PuBu", direction = 1) +
       facet_grid(~lec, scales = "free") +
       xlab("Wind stirring efficiency") +
       ylab("Vertical mixing coefficient") +
       labs(title = "Light extinction coefficient", fill = "NSE") +
       theme_bw()  +
       theme(plot.title = element_text(size = 11, hjust = 0.5))

ggsave(filename = "Figure_03.png", width = 8, height = 4)
```

The calculated NSE values varied significantly among the combinations, ranging from 0.03 to 0.95. The combination of wind stirring efficiency = 0.6, vertical mixing coefficient = 500, and light extinction coefficient = 0.25 m^{-1} achieved the highest NSE value of 0.95 (i.e., considered to be the optimal set of parameter values) (Figure 3).

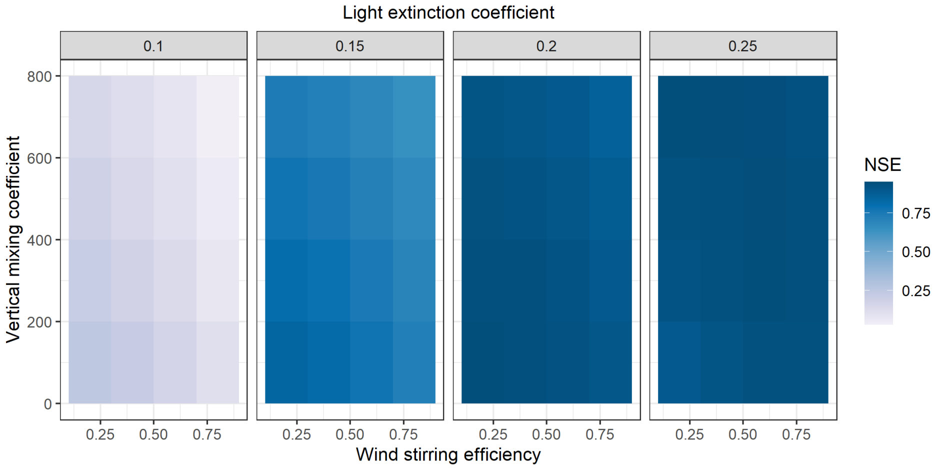

Figure 3: Heat map of Nash-Sutcliffe Efficiency (NSE) coefficient values for temperature simulations in Lake Okareka estimated for the combinations of three parameters given in Table 2 (wind stirring efficiency, vertical mixing coefficient, and light extinction coefficient [m^{-1}]).

A few other combinations could also achieve similar outcomes. For example, when the light extinction coefficient = 0.25 m^{-1}, the NSE values ranged between 0.90 and 0.95, regardless of the wind stirring efficiency and vertical mixing coefficient values shown in Table 2. Figure 3 also revealed that temperature simulations were generally poor when light extinction coefficient = 0.1 and 0.15 m^{-1}, suggesting that these values of light extinction coefficient are too low to achieve a suitably accurate temperature simulation for Lake Okareka.

Under the optimal set of parameter values identified from the calibration assistant function (Figure 3), we re-ran the DYRESM-CAEDYM model and visualised the temperature simulations with the four post-processing functions. An example R code for visualisation functions is shown preceding the corresponding figure.

The temperature simulations in Lake Okareka displayed a clear and consistent seasonal pattern. Temperature was up to 24 °C in the summer season and was below 10 °C in the winter (Figure 4). The simulation contour plot showed that the thermocline of Lake Okareka (the layer of most rapid temperature change) was around 15 m in summer, but the lake always mixed in winter (Figure 4), corresponding to isothermal conditions over the water depth, suggesting a monomictic lake mixing regime. Observations are shown as dots in Figure 4 and match well with simulations both spatially (i.e., vertically) and temporally (i.e., horizontally).

```
# Extract temperature simulations
var.values <- ext_output(dycd.output = "DYCD_Okareka/DYsim.nc",
                         var.extract = c("TEMP"))

# Interpolation of temperature across water column at an interval of 0.5 m
temp.interpolated < -interpol(layerHeights = var.values$dyresmLAYER_HTS_Var,
                              var = var.values$dyresmTEMPTURE_Var,
                              min.dept = 0, max.dept = 33, by.value = 0.5)

# Read in observed water quality data
obs.okareka <- read.csv("plotting_data/Obs_data_template.csv")
obs.okareka$Date <- as.Date(obs.okareka$Date,format="%d/%m/%Y")
# subset observed data to remain temperature observations
obs.temp <- obs.okareka[, c('Date','Depth','TEMP')]

# Contour plot
png(filename = 'Figure_04.png', width = 1200, height = 700)
plot_cont_com(sim = temp.interpolated,
              obs = obs.temp,
              plot.start = "2002-01-23",
              plot.end = "2006-12-31",
              sim.start = "2002-01-23",
              sim.end = "2016-12-31"
              legend.title = "T\n(\u00B0C)",
              min.depth = 0,
              max.depth = 33,
              by.value = 0.5,
              nlevels = 20)
dev.off()
```

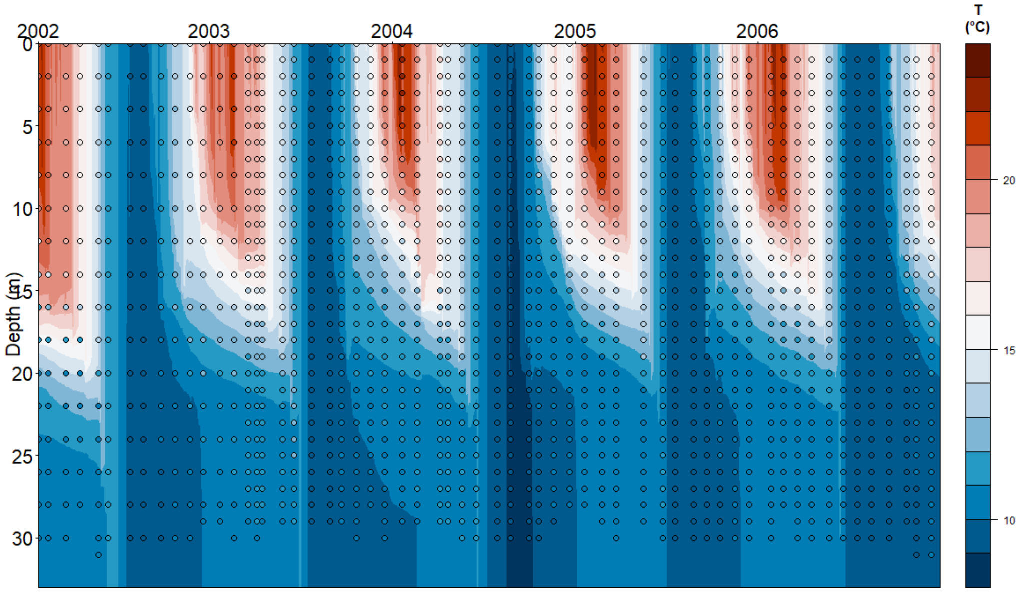

Figure 4: Example contour plot of temperature simulations and observations for Lake Okareka from 2002-01-23 to 2006-12-31. Each dot represents an observation at a specific depth on a specific date. Dots are colour coded on the same scale (left-hand side) as the simulation.

Simulations and observations of temperature with depth are shown in Figure 5 for the dates when observations were available. The vertical profiles provide a more discerning view of details such as the thermocline depth in Lake Okareka. The simulated temperature profile mostly aligned well with the observed profile but had a tendency to overestimate the depth of the thermocline in the first summer (early 2002), whilst capturing the timing of mixing (little or no temperature gradient) in winter.

```
 # Profile plot
 plot_prof(sim = temp.interpolated,
           obs = obs.temp,
           sim.start = "2002-01-23",
           sim.end = "2016-12-31",
           plot.start = "2002-01-23",
           plot.end = "2002-12-31",
           min.depth = 0,
           max.depth = 33,
           by.value = 0.5,
           xlabel = "Temperature \u00B0C")

ggsave(filename = "Figrue_05.png", height = 4, width = 7)
```

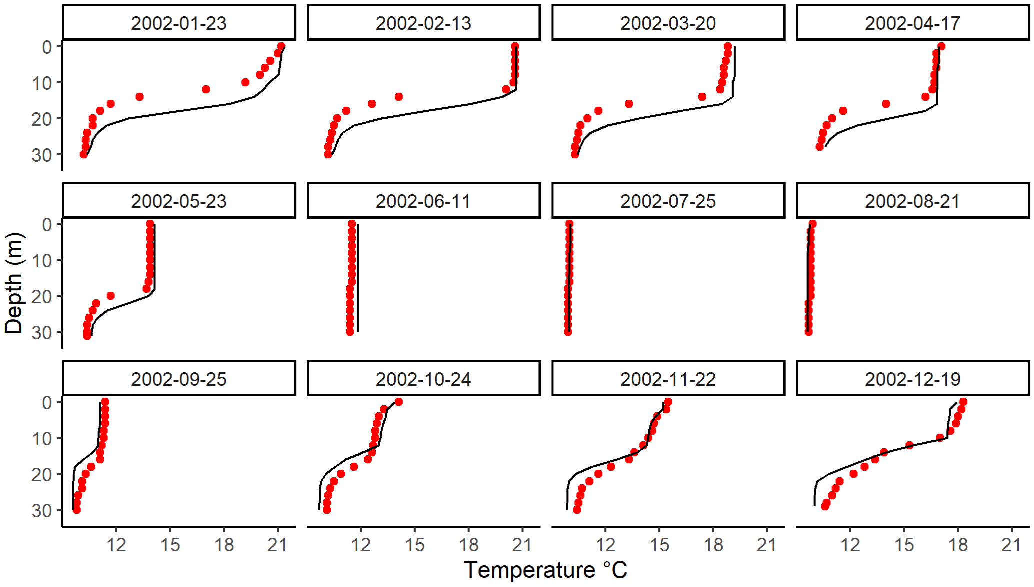

Figure 5: Example profile plot of temperature simulations (black line) and observations (red dots) for Lake Okareka from 2002-01-23 – 2002-12-19.

The time series figure displays temperature simulations as a continuous line for a specific depth over the entire simulation period, with observations shown as dots (Figure 6). The temperature in the surface of the lake (1 m deep) varied strongly over the years, with a clear seasonal pattern, while at mid-depth (14 m deep) it varied to a lesser degree but showed a similar pattern, and in the near-bottom (30 m deep) layer it remained relatively unchanged.

```
# Time series plot
p <- plot_ts(sim = temp.interpolated,
             obs = obs.temp,
             target.depth = c(1, 14, 30),
             sim.start = "2002-01-23",
             sim.end = "2016-12-31",
             plot.start = "2002-01-23",
             plot.end = "2012-12-31",
             min.depth = 0,
             max.depth = 33,
             by.value = 0.5,
             ylabel = "Temperature \u00B0C")

rmse_dpt01 <- objective_fun(sim = temp.interpolated[2:4,],
                            obs = obs.temp[obs.temp$Depth == 1, ],
                            fun = c('RMSE'),
                            start.date = '2002-01-23',
                            end.date = '2016-12-31',
                            min.depth = 0.5,
                            max.depth = 1.5,
                            by.value = 0.5)

rmse_dpt14 <- objective_fun(sim = temp.interpolated[28:30,],
                           obs = obs.temp[obs.temp$Depth == 14, ],
                           fun = c('RMSE'),
                           start.date = '2002-01-23',
                           end.date = '2016-12-31',
                           min.depth = 13.5,
                           max.depth = 14.5,
                           by.value = 0.5)
```

```
rmse_dpt30 <- objective_fun(sim = temp.interpolated[60:62,],
                            obs = obs.temp[obs.temp$Depth == 30, ],
                            fun = c('RMSE'),
                            start.date = '2002-01-23',
                            end.date = '2016-12-31',
                            min.depth = 29.5,
                            max.depth = 30.5,
                            by.value = 0.5)

rmse_text <- data.frame(x = as.Date('2007-06-01'),
                         y = 25,
                         Depth = c(1, 14, 30),
                         label = c(paste0('RMSE = ', round(rmse_dpt01$RMSE,2), ' \u00B0C'),
                                   paste0('RMSE = ', round(rmse_dpt14$RMSE,2), ' \u00B0C'),
                                   paste0('RMSE = ', round(rmse_dpt30$RMSE,2), ' \u00B0C')))

p + ylim(8,26) +
    geom_text(data = rmse_text,
              mapping = aes(x = x, y = y, label = label))

ggsave(filename = 'Figure_06.png', height = 4, width = 7)
```

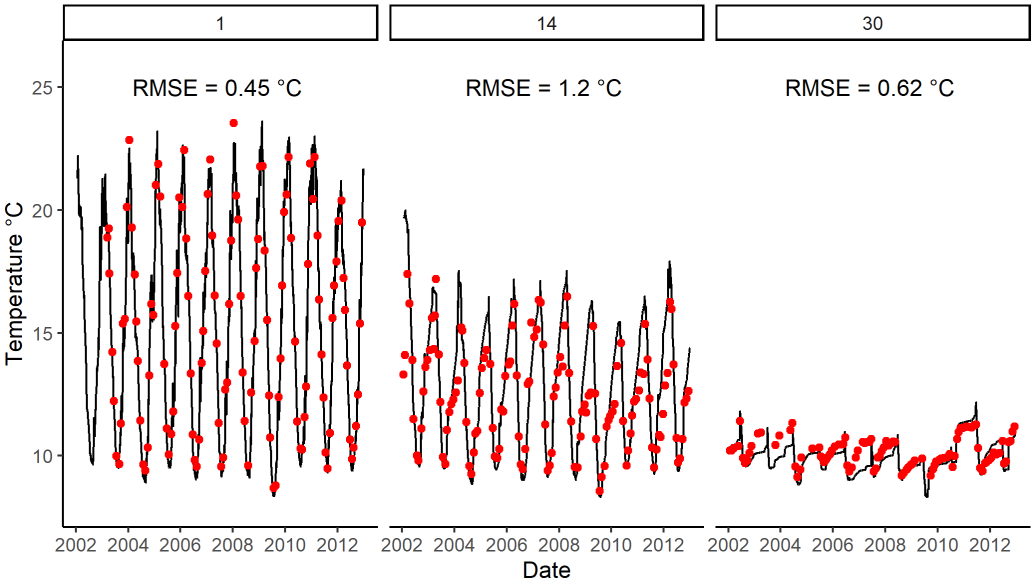

Figure 6: Example time series plot of temperature simulations (black line) and measurements (red dots) for three depths (1 m – near-surface layer, 14 m – middle layer close to the thermocline, and 30 m – near-bottom layer) of Lake Okareka from 2002-01-23 to 2012-12-31. The RMSE values calculated from the 'objective_fun' function in the package are also shown for each depth profile.

The scatter plot compares all temperature simulation values against observations denoted as dots in a Cartesian coordinate system, with a reference line of y=x to indicate the ideal fit (1:1) of simulations to observations. Each point is colour coded by depth. The majority of dots were close to the reference line, indicating good performance of DYRESM-CAEDYM in the temperature simulation (Figure 7).

```
# Scatter plot
plot_scatter(sim=temp.interpolated,
             obs=obs.temp,
             sim.start="2002-01-23",
             sim.end="2016-12-31",
             plot.start = "2002-01-23",
             plot.end="2012-12-31",
             min.depth = 0,
             max.depth = 33,
             by.value = 0.5)
```

```
ggsave(filename = 'Figure_07.png', height = 4, width = 7)
```

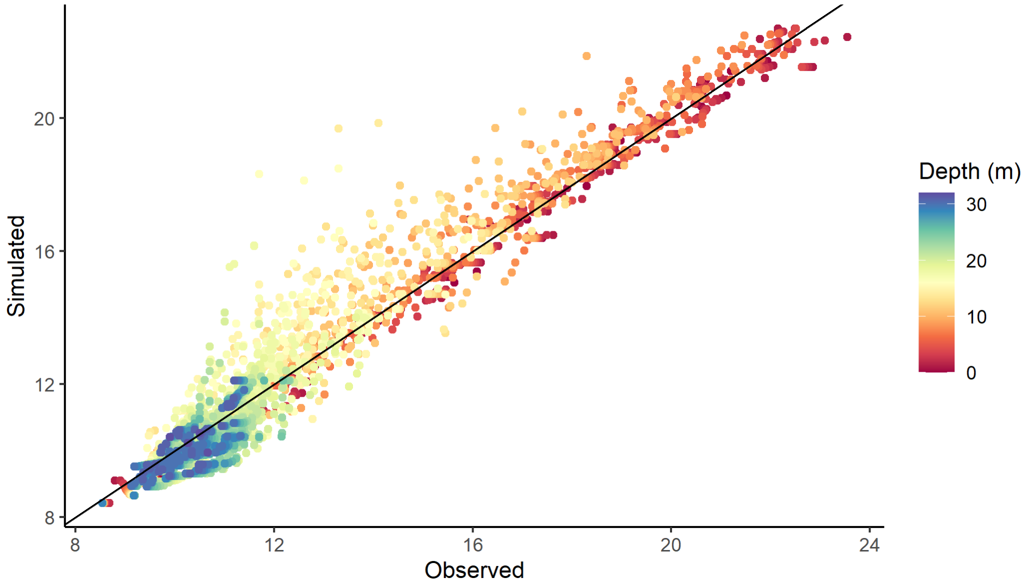

Figure 7: Example scatterplot of temperature simulations and observations for Lake Okareka from 2002-01-23 to 2012-12-31. Note that the dots are colour coded by depth. The red line is a 1:1 slope, i.e., simulated temperature = observed temperature.

Discussion

Aquatic ecosystem models have increasingly been used to shed light on lake ecosystem function (Scheffer et al., 2001) and inform policy and management decisions to improve water quality and control eutrophication (Wang et al., 2012). With progressive increases in the number and complexity of processes represented, and increasing computational power, these models have increased in complexity and have a large number of parameters. These types of models generally require extensive calibration to be suitable for lake-specific simulations, which is quite challenging, despite the development of parameter priors libraries (Robson et al., 2018) that can help with understanding possible ranges of parameters and synthesise values used in other studies. In addition, multifaceted and multilayered modelling results need to be visualised in various ways to express model uncertainty and communicate effectively with a broader community that may be engaged in understanding and interpreting the implications of model scenarios. Here, we have presented an open-source R package **dycdtools** to help address these challenges in calibrating a widely used aquatic ecosystem model DYRESM-CAEDYM and visualising its simulation results. Importantly, this package has the potential to be adjusted for other aquatic ecological models with a similar structure.

Calibration assistant function

The output of the calibration assistant function in the **dycdtools** package is a list of combinations of parameter values with their objective function values, rather than a single optimal set of parameter values provided by many auto-calibration algorithms, such as the auto-calibration module in the RAPID model developed in David et al. (2011), the auto-calibration algorithm proposed in Luo et al. (2018), and the Genetic Algorithm developed in Goldberg and Holland (1988). We chose not to provide a single parameter set partly due to the potential for parameter equifinality (Beven and Binley, 1992), a classical problem of optimisation where different sets of parameter values may provide similar simulation fit to observations. This problem is particularly pertinent for complex aquatic ecosystem models (e.g., DYRESM-CAEDYM) that include a large number of parameters. Many studies illustrated the difficulties of finding a single optimal set of parameter values in the high-dimensional parameter space, due to inter-correlation between parameters and autocorrelation of the residuals (Fernández Slezak et al., 2010; Gupta et al., 2006). Therefore, instead of using an auto-calibration algorithm to decide the optimal set of parameter values, we suggest users compare those combinations of parameter values that result in high objective values and manually choose the combination that they believe is the most physically meaningful. Such a process aligns with the concept of development of user expertise and knowledge that are critical for successful model applications (Hipsey et al., 2015).

It is worth noting that an objective function is a means to characterise the performance of environmental models, and is useful and often necessary (Bennett et al., 2013). A substantial body of work has been done to propose methods and criteria to judge the performance of environmental models, often for hydrological models (Jakeman et al., 2006; Krause et al., 2005) and less frequently for ecological models (Risbey et al., 1996). Many of the proposed objective functions have been summarised in Bennett et al. (2013) and can be applied to characterise performance of lake models. Five of the most commonly used objective functions have been included in the current package, and we plan to add functions to give users more choices of characterising model performance and to better inform parameter calibration.

Given that it is not realistic to expect any one set of parameter values to be truly representative of the actual parameter set (Beven and Binley, 1992), it is also important to convey the accuracy of the simulations and the associated uncertainties. In lake modelling, multi-parameter perturbation has been conducted using Monte-Carlo techniques to inform parameter uncertainty effects (Luo et al., 2018; Muraoka, 2019). The calibration assistant function developed here can be used to carry out such a sensitivity analysis by perturbing multiple parameters within a certain range when the “combination” argument is set to “random”. User expert knowledge (Lehmann and Hamilton, 2018) and parameter priors libraries (Robson et al., 2018) are also valuable in narrowing parameter ranges which can be important in reducing computation times of calibration and sensitivity analysis, as well as reducing the incidence of common issues such as equifinality.

Suggestions on the use of calibration assistant function

The calibration assistant function developed in this study is not intended to replace manual calibration, rather, it is aimed to save time assigned to calibration and sensitivity analysis by automatically trialling a number of parameter values so as to bring a focus to the range of sensitive parameter values. We recommend manual calibration as a means to develop expert knowledge about model parameter values, in an iterative procedure that includes assistant calibration and sensitivity analysis, as well as literature review of parameters. We also recommend a combination of OAT and AAT calibration, with the former used to better understand individual responses of parameters and the latter used with care because exponentially large numbers of possible parameter value combinations are possible and computational times could become untenable even with parallel processing enabled. For example, in the case of 10 parameters where each parameter has three possible values, the total number of possible combinations is 3^{10} = 59,049. If each model run takes 2 minutes, then it would take 59,049 × 2 = 118,098 minutes (> 32 days) to run these combinations.

Manual calibration has also been used as a way to work around the difficulties with weighting and prioritising state variables using an auto-calibration procedure (Lehmann and Hamilton, 2018). Therefore, while acknowledging the benefits of assistant calibration in trialling a number of parameter values without the heavy time burden of manual adjustment, we encourage users to actively engage in the calibration process. More specifically, users may first define suitable value ranges of various parameters from the literature (including other modelling studies) and lab or field experiments (Robson et al., 2018), and then conduct both manual and automated sensitivity analysis to understand which parameters are sensitive and should be more strategically evaluated. Finally, users can apply both the calibration assistant function and manual calibration to optimise parameter values. This process can be repeated multiple times until users are satisfied with accuracy of the simulations, assessed using quantitative output statistics and qualitative comparisons with observed data.

Visualisation of model outputs

Visualisation of model outputs is an important tool for effectively engaging academic, the public and environmental managers in expressing model uncertainty, and as a prerequisite for improving confidence in scenario simulations. This is particularly true for multidimensional model outputs that are intangible, and visualisation has a fundamental role in exploring information and generating understanding (McInerny et al., 2014). The **dycdtools** package expands on existing lake model visualisation tools, such as contours, profiles and time series used by the **glmtools** and **LakeEnsemblR** packages, by providing additional functionality including scatterplots. Each of the plotting functions is designed to best suit different presentation purposes, as outlines in the "Post-processing functions" Section. We anticipate that the developed visualisation tools in **dycdtools** package can not only complement the existing associated GUI by providing more flexible ways of visualising DYRESM-CAEDYM model outputs, but can also be adapted for other lake ecosystem models, such as the General Lake Model (Hipsey et al., 2019) and the Fresh-water Lake Model for LM (Mironov et al., 2010).

In summary, dycdtools is a modular, flexible, and open-source tool that is designed to make the calibration process of the DYRESM-CAEDYM model substantially less time-consuming and to visualise complex modelling results in multiple ways for effective communication. It is suitable for users with various levels of expertise. This study places great emphasis on tools to evaluate the quality of model calibration as many of these models need to provide robust simulations that form the basis of scenario simulations, often designed to evaluate different management and environmental (e.g., climate change, land use change) scenarios.

Acknowledgements

We thank Gebiaw Ayele from Australian Rivers Institute, Griffith University for providing the catchment inflow simulations as part of the example data for the DYRESM-CAEDYM model application. We acknowledge Bay of Plenty Regional Council for provision of data for model runs. Author contributions: S.Y., M.F., and C.M. developed the source codes for the R package; S.Y. prepared the application example; and S.Y., D.H., C.M. and M.F. wrote the paper. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data and software availability

The published dycdtools package can be downloaded from CRAN and the developer version is available from `https://github.com/SongyanYu/dycdtools`. Issues regarding the package can be submitted via `https://github.com/SongyanYu/dycdtools/issues`. The executables of DYRESM-CAEDYM and the example data supporting the case study can be found from `https://github.com/SongyanYu/ExampleData_dycdtools`. The source codes of DYRESM-CAEDYM are accessible from `https://github.com/AquaticEcoDynamics/dy-cd`

Bibliography

N. D. Bennett, B. F. Croke, G. Guariso, J. H. Guillaume, S. H. Hamilton, A. J. Jakeman, S. Marsili-Libelli, L. T. Newham, J. P. Norton, C. Perrin, et al. Characterising performance of environmental models. *Environmental Modelling & Software*, 40:1–20, 2013. URL `https://doi.org/10.1016/j.envsoft.2012.09.011`. [p4, 12]

K. Beven and A. Binley. The future of distributed models: model calibration and uncertainty prediction. *Hydrological processes*, 6(3):279–298, 1992. URL `https://doi.org/10.1002/hyp.3360060305`. [p11, 12]

L. C. Bruce, D. Hamilton, J. Imberger, G. Gal, M. Gophen, T. Zohary, and K. D. Hambright. A numerical simulation of the role of zooplankton in c, n and p cycling in lake kinneret, israel. *Ecological Modelling*, 193(3-4):412–436, 2006. URL `https://doi.org/10.1016/j.ecolmodel.2005.09.008`. [p3]

D. F. Burger, D. P. Hamilton, and C. A. Pilditch. Modelling the relative importance of internal and external nutrient loads on water column nutrient concentrations and phytoplankton biomass in a shallow polymictic lake. *Ecological Modelling*, 211(3-4):411–423, 2008. URL `https://doi.org/10.1016/j.ecolmodel.2007.09.028`. [p1]

S. Christensen and J. Doherty. Predictive error dependencies when using pilot points and singular value decomposition in groundwater model calibration. *Advances in Water Resources*, 31(4):674–700, 2008. URL `https://doi.org/10.1016/j.advwatres.2008.01.003`. [p1]

R.-M. Couture, S. J. Moe, Y. Lin, Ø. Kaste, S. Haande, and A. L. Solheim. Simulating water quality and ecological status of lake vansjø, norway, under land-use and climate change by linking process-oriented models with a bayesian network. *Science of the total environment*, 621:713–724, 2018. URL `https://doi.org/10.1016/j.scitotenv.2017.11.303`. [p5]

Y. Cui, G. Zhu, H. Li, L. Luo, X. Cheng, Y. Jin, and D. Trolle. Modeling the response of phytoplankton to reduced external nutrient load in a subtropical chinese reservoir using dyresm-caedym. *Lake and Reservoir Management*, 32(2):146–157, 2016. URL `https://doi.org/10.1080/10402381.2015.1136365`. [p1]

C. H. David, D. R. Maidment, G.-Y. Niu, Z.-L. Yang, F. Habets, and V. Eijkhout. River network routing on the nhdplus dataset. *Journal of Hydrometeorology*, 12(5):913–934, 2011. URL https://doi.org/10.1175/2011JHM1345.1. [p11]

A. Dietzel and P. Reichert. Bayesian inference of a lake water quality model by emulating its posterior density. *Water Resources Research*, 50(10):7626–7647, 2014. URL https://doi.org/10.1002/2012WR013086. [p5]

J. Doherty et al. Pest: a unique computer program for model-independent parameter optimisation. In *NATIONAL CONFERENCE PUBLICATION-INSTITUTION OF ENGINEERS AUSTRALIA NCP*, pages 551–554. Institution of Engineers, Australia, 1994. [p1]

J. Dohorty. Model-independent parameter estimation user manual part i: Pest, sensan and global optimisers. *Watermark Numerical Computing. Haettu*, 17(03):2019, 2018. [p1]

J. A. Elliott. Is the future blue-green? a review of the current model predictions of how climate change could affect pelagic freshwater cyanobacteria. *Water research*, 46(5):1364–1371, 2012. URL https://doi.org/10.1016/J.WATRES.2011.12.018. [p1]

D. Fernández Slezak, C. Suárez, G. A. Cecchi, G. Marshall, and G. Stolovitzky. When the optimal is not the best: parameter estimation in complex biological models. *PloS one*, 5(10):e13283, 2010. URL https://doi.org/10.1371/journal.pone.0013283. [p11]

M. A. Frassl, J. M. Abell, D. A. Botelho, K. Cinque, B. R. Gibbes, K. D. Jöhnk, K. Muraoka, B. J. Robson, M. Wolski, M. Xiao, et al. A short review of contemporary developments in aquatic ecosystem modelling of lakes and reservoirs. *Environmental Modelling and Software*, 117:181–187, 2019. URL https://doi.org/10.1016/j.envsoft.2019.03.024. [p1, 2]

G. Gal, J. Imberger, T. Zohary, J. Antenucci, A. Anis, and T. Rosenberg. Simulating the thermal dynamics of lake kinneret. *Ecological Modelling*, 162(1-2):69–86, 2003. URL https://doi.org/10.1016/S0304-3800(02)00380-0. [p2]

M. Gallagher and J. Doherty. Predictive error analysis for a water resource management model. *Journal of Hydrology*, 334(3-4):513–533, 2007. URL https://doi.org/10.1016/j.jhydrol.2006.10.037. [p1]

D. E. Goldberg and J. H. Holland. Genetic algorithms and machine learning. *Machine Learning*, 1988. [p11]

H. V. Gupta, K. J. Beven, and T. Wagener. Model calibration and uncertainty estimation. *Encyclopedia of hydrological sciences*, 2006. URL https://doi.org/10.1002/0470848944.hsa138. [p11]

D. P. Hamilton and S. G. Schladow. Prediction of water quality in lakes and reservoirs. part i—model description. *Ecological Modelling*, 96(1-3):91–110, 1997. URL https://doi.org/10.1016/S0304-3800(96)00062-2. [p1]

M. R. Hipsey, D. P. Hamilton, P. C. Hanson, C. C. Carey, J. Z. Coletti, J. S. Read, B. W. Ibelings, F. J. Valesini, and J. D. Brookes. Predicting the resilience and recovery of aquatic systems: A framework for model evolution within environmental observatories. *Water Resources Research*, 51(9):7023–7043, 2015. URL https://doi.org/10.1002/2015WR017175. [p12]

M. R. Hipsey, L. C. Bruce, C. Boon, B. Busch, C. C. Carey, D. P. Hamilton, P. C. Hanson, J. S. Read, E. de Sousa, M. Weber, et al. A general lake model (glm 3.0) for linking with high-frequency sensor data from the global lake ecological observatory network (gleon). *Geoscientific Model Development*, 12(1):473–523, 2019. URL https://doi.org/10.5194/gmd-12-473-2019. [p2, 12]

M. R. Hipsey, G. Gal, G. B. Arhonditsis, C. C. Carey, J. A. Elliott, M. A. Frassl, J. H. Janse, L. de Mora, and B. J. Robson. A system of metrics for the assessment and improvement of aquatic ecosystem models. *Environmental Modelling & Software*, 128:104697, 2020. URL https://doi.org/10.1016/j.envsoft.2020.104697. [p1]

A. Imerito. Dynamic reservoir simulation model dyresm v4. 0 science manual. *Centre for Water Research, University of Western Australia*, 50, 2007. [p2]

A. J. Jakeman, R. A. Letcher, and J. P. Norton. Ten iterative steps in development and evaluation of environmental models. *Environmental Modelling & Software*, 21(5):602–614, 2006. URL https://doi.org/10.1016/j.envsoft.2006.01.004. [p1, 12]

P. Krause, D. Boyle, and F. Bäse. Comparison of different efficiency criteria for hydrological model assessment. *Advances in geosciences*, 5:89–97, 2005. URL https://doi.org/10.5194/adgeo-5-89-2005. [p12]

M. K. Lehmann and D. P. Hamilton. Modelling water quality to support lake restoration. In *Lake Restoration Handbook*, pages 67–105. Springer, 2018. [p1, 3, 12]

D. M. Lewis, J. D. Brookes, and M. F. Lambert. Numerical models for management of anabaena circinalis. *Journal of applied phycology*, 16(6):457–468, 2004. URL https://doi.org/10.1007/s10811-004-5506-z. [p1]

L. Luo, D. Hamilton, J. Lan, C. McBride, and D. Trolle. Autocalibration of a one-dimensional hydrodynamic-ecological model (dyresm 4.0-caedym 3.1) using a monte carlo approach: simulations of hypoxic events in a polymictic lake. *Geoscientific Model Development*, 11(3):903–913, 2018. URL https://doi.org/10.5194/gmd-11-903-2018. [p1, 2, 11, 12]

G. J. McInerny, M. Chen, R. Freeman, D. Gavaghan, M. Meyer, F. Rowland, D. J. Spiegelhalter, M. Stefaner, G. Tessarolo, and J. Hortal. Information visualisation for science and policy: engaging users and avoiding bias. *Trends in ecology & evolution*, 29(3):148–157, 2014. URL https://doi.org/10.1016/j.tree.2014.01.003. [p2, 12]

D. Mironov, E. Heise, E. Kourzeneva, B. Ritter, N. Schneider, and A. Terzhevik. Implementation of the lake parameterisation scheme flake into the numerical weather prediction model cosmo. *Boreal Environment Research*, 15:218–230, 2010. [p12]

K. Muraoka. *Novel approaches for modelling changes in phytoplankton diversity and lake ecosystem function*. PhD thesis, The University of Waikato, 2019. [p12]

A. Nielsen, K. Bolding, F. Hu, and D. Trolle. An open source qgis-based workflow for model application and experimentation with aquatic ecosystems. *Environmental Modelling & Software*, 95:358–364, 2017. URL https://doi.org/10.1016/j.envsoft.2017.06.032. [p1]

E. Ofir, J. Heymans, J. Shapiro, M. Goren, E. Spanier, and G. Gal. Predicting the impact of lake biomanipulation based on food-web modeling—lake kinneret as a case study. *Ecological Modelling*, 348:14–24, 2017. URL https://doi.org/10.1016/j.ecolmodel.2016.12.019. [p3]

D. Özkundakci, D. P. Hamilton, and D. Trolle. Modelling the response of a highly eutrophic lake to reductions in external and internal nutrient loading. *New Zealand Journal of Marine and Freshwater Research*, 45(2):165–185, 2011. URL https://doi.org/10.1080/00288330.2010.548072. [p1]

F. Pianosi, K. Beven, J. Freer, J. W. Hall, J. Rougier, D. B. Stephenson, and T. Wagener. Sensitivity analysis of environmental models: A systematic review with practical workflow. *Environmental Modelling & Software*, 79:214–232, 2016. URL https://doi.org/10.1016/j.envsoft.2016.02.008. [p5]

J. Risbey, M. Kandlikar, and A. Patwardhan. Assessing integrated assessments. *Climatic Change*, 34(3): 369–395, 1996. URL https://doi.org/10.1007/BF00139298. [p12]

B. Robson and D. Hamilton. Three-dimensional modelling of a microcystis bloom event in the swan river estuary, western australia. *Ecological Modelling*, 174(1-2):203–222, 2004. URL https://doi.org/10.1016/j.ecolmodel.2004.01.006. [p1, 2, 3]

B. J. Robson and D. P. Hamilton. Summer flow event induces a cyanobacterial bloom in a seasonal western australian estuary. *Marine and Freshwater Research*, 54(2):139–151, 2003. URL https://doi.org/10.1071/MF02090. [p1]

B. J. Robson, G. B. Arhonditsis, M. E. Baird, J. Brebion, K. F. Edwards, L. Geoffroy, M.-P. Hébert, V. van Dongen-Vogels, E. M. Jones, C. Kruk, et al. Towards evidence-based parameter values and priors for aquatic ecosystem modelling. *Environmental modelling & software*, 100:74–81, 2018. URL https://doi.org/10.1016/j.envsoft.2017.11.018. [p11, 12]

J. Romero, J. Antenucci, and J. Imberger. One-and three-dimensional biogeochemical simulations of two differing reservoirs. *Ecological Modelling*, 174(1-2):143–160, 2004. URL https://doi.org/10.1016/j.ecolmodel.2004.01.005. [p2]

B. Z. Rousso, E. Bertone, R. Stewart, and D. P. Hamilton. A systematic literature review of forecasting and predictive models for cyanobacteria blooms in freshwater lakes. *Water Research*, 182:115959, 2020. URL https://doi.org/10.1016/j.watres.2020.115959. [p1]

M. Scheffer, S. Carpenter, J. A. Foley, C. Folke, and B. Walker. Catastrophic shifts in ecosystems. *Nature*, 413(6856):591–596, 2001. URL https://doi.org/10.1038/35098000. [p11]

S. G. Schladow and D. P. Hamilton. Prediction of water quality in lakes and reservoirs: Part ii-model calibration, sensitivity analysis and application. *Ecological Modelling*, 96(1-3):111–123, 1997. URL https://doi.org/10.1016/S0304-3800(96)00063-4. [p1, 3]

S. Takkouk and X. Casamitjana. Application of the dyresm–caedym model to the sau reservoir situated in catalonia, spain. *Desalination and Water Treatment*, 57(27):12453–12466, 2016. URL https://doi.org/10.1080/19443994.2015.1053530. [p1]

R. Wang, J. A. Dearing, P. G. Langdon, E. Zhang, X. Yang, V. Dakos, and M. Scheffer. Flickering gives early warning signals of a critical transition to a eutrophic lake state. *Nature*, 492(7429):419–422, 2012. URL https://doi.org/10.1038/nature11655. [p11]

S. Weinberger and M. Vetter. Using the hydrodynamic model dyresm based on results of a regional climate model to estimate water temperature changes at lake ammersee. *Ecological Modelling*, 244: 38–48, 2012. URL https://doi.org/10.1016/j.ecolmodel.2012.06.016. [p5]

J. T. White, J. E. Doherty, and J. D. Hughes. Quantifying the predictive consequences of model error with linear subspace analysis. *Water Resources Research*, 50(2):1152–1173, 2014. URL https://doi.org/10.1002/2013WR014767. [p1]

Songyan Yu
Australian Rivers Institute and School of Environment and Science
Griffith University
Australia
ORCID: 0000-0001-5765-7060
sunny.yu@griffith.edu.au

Christopher G. McBride
Environmental Research Institute
University of Waikato
New Zealand
ORCID: 0000-0001-5784-9767
cmcbride@waikato.ac.nz

Marieke A. Frassl
Australian Rivers Institute and School of Environment and Science
Griffith University
Australia
ORCID: 0000-0001-8843-8477
marieke.frassl@gmail.com

Matthew R. Hipsey
Aquatic Ecodynamics, UWA School of Agriculture and Environment
The University of Western Australia
Australia
ORCID: 0000-0001-8386-4354
matt.hipsey@uwa.edu.au

David P. Hamilton
Australian Rivers Institute and School of Environment and Science
Griffith University
Australia
ORCID: 0000-0002-9341-8777
david.p.hamilton@griffith.edu.au