Supporting Information AND `READ.me`: `PRYSM v2.0`: A Proxy System Model for Lacustrine Archives

Sylvia G. Dee[1,2,3,*], James M. Russell[1,2], Carrie Morrill[4,5], Zihan Chen[1], Ashling Neary[1]

1. Department of Earth, Environmental, and Planetary Sciences,

Brown University, Providence, RI 02912

2. Institute at Brown for Environment and Society,

Brown University, 85 Waterman Street, Providence, RI, 02903, USA

3. Department of Earth, Environmental, and Planetary Sciences,

Rice University, Houston, TX, 77005

4. Cooperative Institute for Research in Environmental Sciences,

University of Colorado, Boulder, CO 80309

5. NOAA's National Centers for Environmental Information, Boulder, CO 80305

*Corresponding Author: Sylvia G. Dee, `sylvia.dee@rice.edu`

September 26, 2018, 3:32pm

READ.me for PRYSM v2.0: LakePSM*

* Contributing Authors: Sylvia Dee, Zihan Chen

1. INSTALLATION

1.1. Fortran and Python Installation Guide

To install `LakePSM`, the user must first download or have a working FORTRAN compiler. We recommend `gfortran`, `https://directory.fsf.org/wiki/Gfortran`. The user must also have Python and iPython installed. This model was built using Python version 2.7, but given its few dependencies (see below) should be compatible with Python 3. (Optional): To simplify the data analysis process for the model outputs, we recommend `Jupyter Notebook`, `http://jupyter.org/`, for interactive data visualization. The example plotting scripts are available in both python file(.py) and `Jupyter Notebook`(.ipynb).

Python Dependencies: We suggest that users install `Anaconda`, `https://www.anaconda.com/distribution/`, for python package/dependency management.

- NumPy
- Matplotlib
- f2py: interfacing Fortran and Python languages
- RPy2: interfacing R and Python languages.

When using `Jupyter Notebook` for data-analysis in the conda environment, users may need to install extra software `nb_conda` to switch kennels from different environment.

1.2. MODEL SCRIPTS

- `lake_driver.py`: driver script to execute full PSM
- `lake_setup.inc`: common block/include file for the Environment sub-model
- `env_sub.f90`: environment-sub model
- `sensor_gdgt.py`: sensor-sub model for GDGTs
- `sensor_carbonate.py`: sensor-sub model for carbonates
- `sensor_leafwax.py`: sensor-sub model for Leaf Wax δD
- `lake_archive_compact.py`: sedimentation and compaction function
- `lake_archive_bioturb.py`: bioturbation function
- `lake_obs_bchron2.py`: application of age uncertainties with `Bchron`
- `lake_obs_analytical_error.py`: adds analytical error using random Gaussian noise

The following scripts are also included on the GitHub page to assist users in plotting the Lake PSM's output, generating input files, and other auxiliary functions.

- `ERA_Climatology.py`: create Lake-PSM formatted input climatology file from ERA-Interim Reanalysis files
- `ERA_Interim.py`: create Lake-PSM formatted input monthly time series file from ERA-Interim Reanalysis files
- `plot_lake_env_input.py`: example plotting script for the lake climatology input.

- `plot_lake_env.py`: plotting script for outputs from the environment sub-model. Options include seasonal averages of surface temperature, average mixing depth, evaporation and dO^{18} and δD of lake surface waters
- `plot_lake_temp_profile.py`: plotting script to generate lake temperature profile from the environment-sub model

2. RUNNING THE MODEL

Before running `LakePSM`, users must first set up the lake-specific environment parameters and initial status in the common block script `lake_setup.inc`. The example setup uses the data from Lake Tanganyika in East Africa. To run `LakePSM`, open the script `lake_driver.py` and follow the detailed comment instructions to run each sub-model in the interactive shell of IPython. Special note for the environment-sub model: users need to generate a Fortran executable with the `f2py` interface.

2.1. ENVIRONMENT:

Hydrological, Isotopic & Energy Balance Sub-Model

The lake model requires seven input variables from either meteorological observations, reanalysis products, or GCMs: near-surface air temperature, near-surface specific humidity, downward shortwave radiation, downward longwave radiation, near-surface wind speed, surface pressure, and precipitation. For these variables, the current model configuration uses monthly mean input data, but previous versions have been run with daily or 6-hourly inputs, and the model can be adapted depending on the user's research needs. Runoff amounts and water isotope ratios are needed if the user wishes to simulate water balance, lake level, or water isotope balance. A full description with successful applications of the water balance module is given in Benson and White (1994); Hostetler (2009); Hostetler and Benson (1994). With the exception of the water isotope ratios and catchment data, the climatological input variables are readily available in reanalysis data (e.g. NCEP or ERA-Interim) and for nearly all of the climate and paleoclimate model intercomparison project (CMIP5 and PMIP3) simulations at monthly or finer resolution. A full list of inputs and outputs for the *Environment* sub-model is given in Table 1, and Figure 1 gives a summary schematic of the sub-model functions.

Only one input file is needed, the nomenclature and type of which the user can specify e.g. `Lake_Input.txt`. This input file is read in the environment sub-model (`env_sub.f90`) in subroutine `DATA_IN`, and contains the meteorological data that forces the model in a simple text file. The current configuration of the model expects monthly inputs, so each row in this file corresponds to a one month time slice. Values in each row from are used as model input (see rows A-N in Table 1). For the environment sub-model, input rows A through I are required, and presently included. Rows I, J are required to calculate the water balance, and rows K through N are necessary to model stable water isotopes. If water balance and/or isotopes will not be modeled, these rows can either be left blank in the input file or filled with some sort of missing value.

The environment model then calculates standard outputs from the surface energy balance, wind-driven turbulent mixing using an eddy diffusion scheme, and density-driven convective mixing. Two output files are generated from the model, `surface.dat, profile.dat`. The first file (`surface.dat`) contains variables including julian day, lake surface temperature, mixing depth, evaporation, water isotopes, ice height and fraction, and lake level and discharge, all daily averaged values arranged in the column sequence listed in the output column in Table 1. The model may be additionally configured to return the outputs labeled (∗) via a change to the subroutine `SHUFFLE`. The second file (`profile.dat`) returns the temperature profile with depth at 10-m increments, and this output can be adapted in `SHUFFLE` depending on lake depth.

2.2. SENSOR MODELS

2.2.1. Leaf Wax δD

The leaf wax sensor model requires inputs of the isotope ratios in precipitation (δD_{PRECIP}, specifically). These can be obtained either through the use of an isotope-enabled GCM, extracted from the grid point or region of the lake site, or from observations such as the Global Network of Isotopes in Preciptiation (GNIP). Gridded products such as the OIPC also can be used as inputs to the leaf wax sensor model if needed.

The leaf wax sensor model is run in `lake_driver.py` via the calls to functions `wax_sensor` and `wax_uncertainty`. The output is a monthly time series of simulated δD_{WAX} for the lake site, along with uncertainties. Note that the user must specify default values for the ϵ apparent fractionation term, when possible. The sensor model has an error term based on the values reported in Sachse et al. (2012) and Konecky, Russell, and Bijaksana (2016), as noted in the main text. The sensor model code accounts for this uncertainty by sub-sampling the value of ϵ 1000 times in a Monto-Carlo fashion within the range of plausible errors. The model then provides a 95% confidence interval of simulated δD_{WAX} with the standard output.

With respect to seasonality of leaf production, the leaf wax module would only account for leaf wax seasonality if using a vegetation module that includes it (e.g. see reference in main text to *Konecky & Dee et al., in revision*). Alternatively, the user could drive the leaf wax sensor model with the δD of precipitation during the season of leaf wax production, if this is known.

2.2.2. Carbonate $\delta^{18}O$

To run the carbonate $\delta^{18}O$ sensor model, the isotopic water balance scheme in the environment model must be turned on via a series of logical statements found in the include file. To facilitate this, the lake environment model must be configured to run with additional inputs of $\delta^{18}O$ and δD of precipitation and runoff. If these input files are available, e.g. from an isotope enabled model or using local observations, the water isotope ratios of lake surface water will be computed at each time step and included in the output file (surface.dat) as separate columns.

The sensor model pulls this result ($\delta^{18}O$ of lake surface water) to drive the carbonate forward model (see Fig. 2). Users may specify which carbonate-temperature equation they prefer (see main text) by changing the 'model' input key in the function call to `carbonate_sensor`. The various calibrations can also be used to perform sensitivity test or quantify uncertainties in a hypothetical carbonate reconstruction.

If no water isotope data is available for use with the environment/full water balance scheme in the lake model, one may specific a temporal average $\delta^{18}O$ of lake water and change the lake surface temperature alone to model $\delta^{18}O$ of carbonates. However, this simplification assumes that no major hydrological changes, i.e. source water or precipitation/evaporation changes, have occurred over time.

2.3. GDGTs

The GDGT sensor model is run using lake surface temperatures or air temperatures as the input driver. Users must specify which calibration to use in the model or supply their own (see main text) and provide this information in the function call to `gdgt_sensor.py`. Users must also specify calibration uncertainties such that reasonable error space is returned with the simulated GDGT time series.

2.4. ARCHIVE MODEL

The archive model consists of two functions, one which simulates sedimentation and subsequent compaction of the sediment, and one which simulates bioturbation. The sediment compaction function requires the user to specify a sedimentation rate and a sediment depth vs. age relationship, if known. The input variables include: porosity (set to 0.95 by default) and the sedimentation rate (S). It returns the porosity profile with depth, z, and provides code for plotting the original sediment vs. depth relationship against the compacted sediment-depth relationship.

The bioturbation model requires the following inputs that the user must specify: age of the oldest layer, mixed layer thickness (cm), proxy type, and species abundance (`abu`). The species abundance term is set to 200 by default, but this parameter may be varied to generate uncertainty bounds or conduct sensitivity tests using a reasonable range of values for `abu`. Species abundance is most easily understood using carbonates as an example.

The full description of these input values is given in the original publication of TURBO2, Trauth (2013):

> The input variables for TURBO2 are the record of the abundance abu of a signal carrier (such as a foraminifera species), its isotope signature iso, the mixed layer thickness mxl through time (or down core), and a single value numb defining the number of foraminifera tests that are to be picked and have their isotope values measured. While there is only one foraminifera species in the input of turbo2 (called Species 1 and identified by the number 1 in the algorithm), the program creates a second foraminifera species (called Species 2) which has an abundance variation complementary to that of Species 1, such that the sum of both abundances is constant.

While this model was specifically designed for foramniferal species, the species abundance term has a minor impact on the final bioturbated time series, which can be isotopic, GDGT-based, etc. Users should simply use the sensor model output as the input for 'iso.'

2.5. OBSERVATION MODEL

The observation sub-model simulates an ensemble of plausible chronologies using `Bchron`. The user must specify calibrated ages and standard deviations of those ages, as well as the total depth of the sediment core and the depths where the ages occur. The output is stored in the variable `chrons` and can be analyzed alongside the 'true' pseudo-proxy time series to evaluate the impacts of dating uncertainties on the climate signals embedded in the record. For example, one can imagine computing the correlation between the data assuming 1000 different chronologies and the climate variable of interest in the climate model (e.g. precipitation amount) to see how much the dating uncertainties alone can impact the retrieval of this information.

Secondly, the observation sub-model contains a function `obs_analytical.py`.

3. INPUT/OUTPUT FILES

1. INPUT

- `ERA_INTERIM_1979_2016_Tanganyika.txt`

- PMIP3: Example Input File Folder contains historical, pre-industrial, mid-Holocene, and LGM inputs formatted for use with the to the lake environment model
- `CAM_PR_1850_2014.txt`: precipitation amount (if using lake level and water balance module)
- `CAM_d18OP_1850_2014.txt`: precipitation isotopes (if using water isotope balance module)
- `CAM_dDP_1850_2014.txt`: precipitation isotopes (if using water isotope balance module)
- `TEX86_cal.csv`: example file with calibrated age model information, used to run `BChron`in the Observation sub-model

2. OUTPUT

- `ERA-HIST-Tlake_surf.dat`: Environment sub-model output, surface water variables (see Table 1)
- `ERA-HIST-Tlake_prof.dat`: Environment sub-model output, water column profile for temperature
- `Environment_Output_LST.npy`: lake surface temperature time series vector
- `Environment_Output_MIX.npy`: lake mixing depth time series vector
- `leafwax_sensor.npy`: time series of δD_{WAX} from leaf wax sensor model
- `carb_sensor.npy`: time series of $\delta^{18}O_{CARBONATE}$ from lacustrine carbonate sensor model. Note that different model (O'Neil, Bemis, etc.) is appended to output filename.

- `gdgt_sensor.npy`: time series of GDGT values (e.g. MBT or TEX_{86}) from GDGT sensor model, with calibration errors. Note that different calibration choice β is appended to output filename.
- `chrons.npy`: output from Bchron with age model ensemble (n=1000)

4. MODEL TUNING

The lake energy balance model additionally requires a number of user-defined parameters and initial conditions. The file `lake.inc` includes parameter definitions, some of which are lake-specific. In this file, the user should specify the Earth's obliquity (degrees), the lake's latitude, maximum lake depth (i.e., the depth of the lake when it is at the sill elevation, in meters), the elevation of the basin bottom (meters), the area of the drainage basin when lake depth equals zero (hectares or $10^4 m^2$), the neutral drag coefficient CDRN (unitless, and varies from 1.0e-0.3 to 2.5e-0.3 depending on lake area and fetch), shortwave extinction coefficient (η,1/meters), albedo of melting and non-melting snow, prescribed or initial lake depth (meters, typically represents mean lake depth), prescribed or initial lake salinity (parts per thousand), $\delta^{18}O$ and δD of air above the lake (‰ SMOW), and the basin hypsometry (hectares or $10^4 m^2$; defined as the surface area the lake would attain at one meter depth increments from sill elevation to basin bottom). Logical statements in the `lake.inc` file are used to turn on/off the following modules: water balance calculations, lake ice, salinity profile, lake water $\delta^{18}O$and δD, and explicit boundary layer computations for sigma-coordinate meteorological inputs from climate models.

Finally, initial temperature, salinity, and isotopic profiles are specified within the lake model code itself, in the subroutine `init_lake`. Other initial conditions including lake ice fraction and height, and height of snow present on top of lake ice are initialized at zero in this subroutine. In our implementation, we additionally allow the user to specify depth-varying salinity, and albedos for melting and non-melting snow to provide more flexibility in lake-specific applications.

In general, lake-specific tuning of the environment model parameters is necessary. First, η, or the shortwave extinction coefficient (η), defines the attenuation of shortwave radiation with depth in the lake as:

$$I_Z/I_0 = e^{-\eta z} \tag{1}$$

where z is depth, I_Z is net downward shortwave radiation at depth z, and I_0 is shortwave radiation penetrating the surface of the lake. The variable η can be set using this relationship with $I_z/I_0 = 0.01$, or 1%, or using a simple Secchi Disk conversion, $(1.57/Z_{SD} = \eta)$ Descy et al. (2006). Controlling transparency in lake water column, the shortwave extinction coefficient η has a large impact on lake thermal stratification and seasonal mixing. By comparing average mixing depth from our sensitivity experiment on *eta* and from observational record, we estimated the best-fit value of η to be 0.065 for Lake Tanganyika (see Fig. 3).

Secondly, reflecting the surface friction of water, the neutral drag coefficient (model variable `CDRN`) depends on lake area and the height above the lake at which meteorological inputs are measured. The latter is set as z_2m in `lake.inc` (and see Garratt (1977);

Strub and Powell (1987) for more information). `CDRN` impacts the vertical transfer of heat and water vapor between the lake and overlying atmosphere, but also mixing depth and turbulent kinetic energy. To calculate this transfer, the model uses empirical bulk formulae, which rely on gradients of temperature and humidity between the lake and atmosphere along with exchange coefficients. Studies measuring this value at the boundary layer (10m) over lakes have reported values of 1.3e-03, but closer to the surface (e.g., at a level of 2m above the surface) this value is higher. We use values appropriate for the 2m level since the climatological inputs used here are predominantly from reanalysis and GCM outputs at the 2m level. Thus we initialized the model with a value of 2.0e-03 (see Fig. 4). The user may specify this value within a reasonable range (1.0e-0.3 to 2.5e-0.3) based on the observation height, wind speed, surface roughness, and size of the lake. Our sensitivity experiment suggests a strong correlation between the neutral drag coefficient and lake surface temperature.

In general, values for lake-specific parameters (e.g., depth, salinity, shortwave extinction coefficient) should be set using observations for modern simulations. Finally, given the necessity of de-biasing GCM output, we employ the commonly-used delta method to generate climate inputs for paleo PSM simulations, which involves scaling modern observations or reanalysis data by the paleo to modern change (δ) simulated by the GCM Lorenz, Nieto-Lugilde, Blois, Fitzpatrick, and Williams (2016).

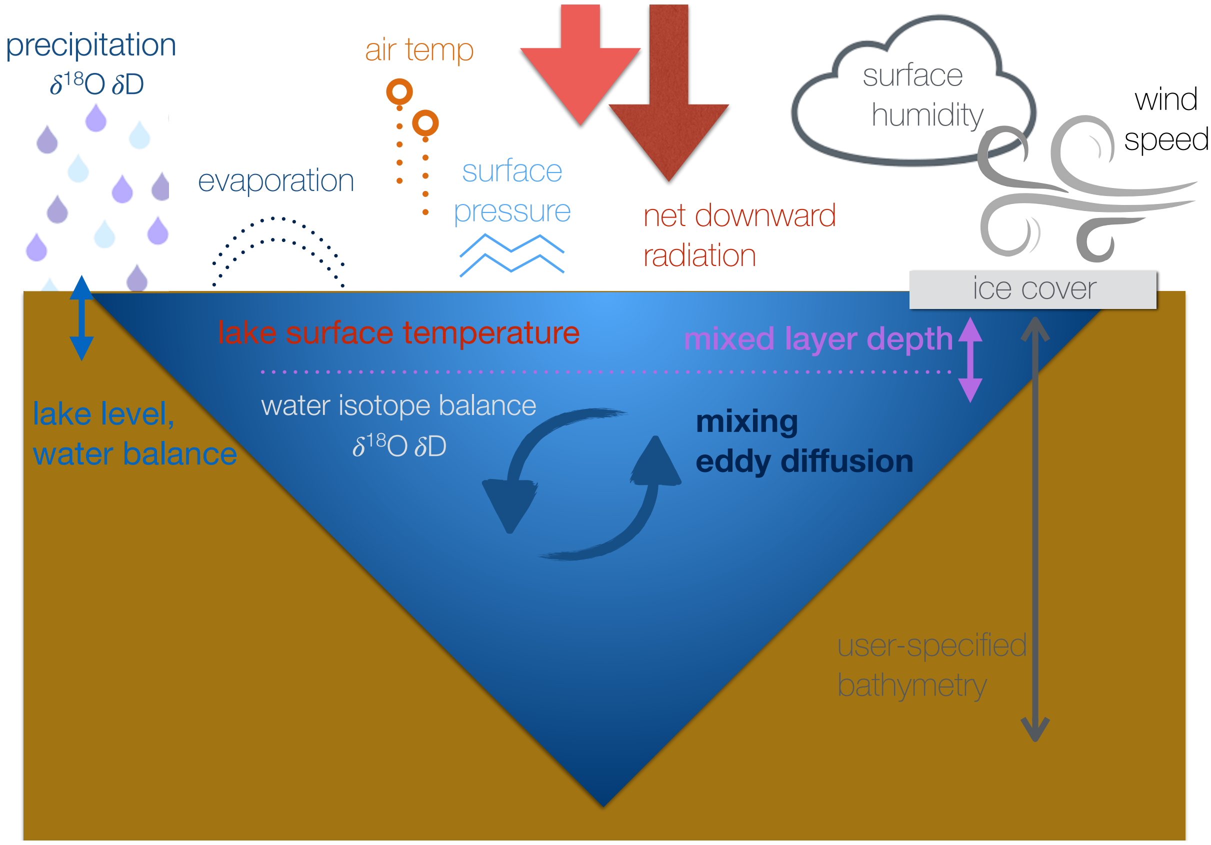

Figure 1: Schematic of the Lake Energy Balance Model (Environment Sub-Model in full Lake PSM. The model computes vertical profiles of temperature and the mixing profile with depth, and can be configured for ice or ice-free conditions. Required inputs are shown above the lake surface. Outputs are shown below the lake surface. See Table 1 for further reference.

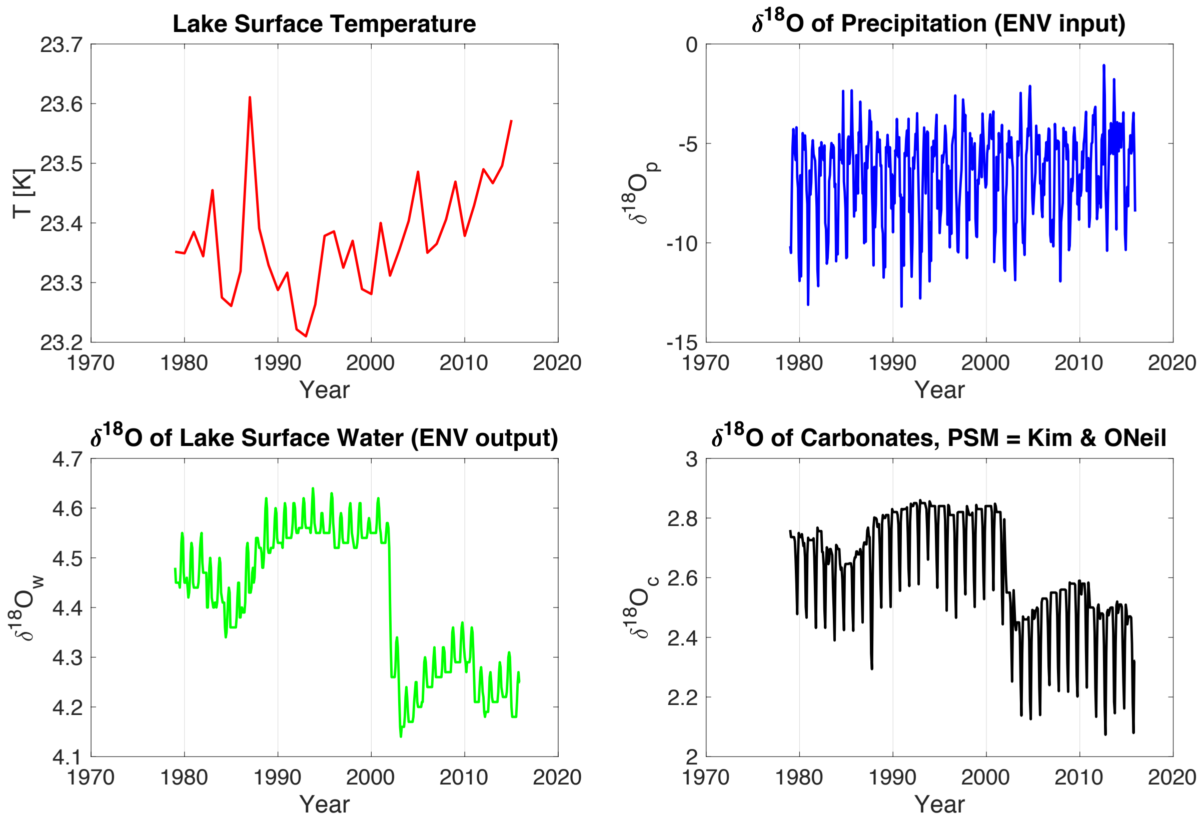

Figure 2: Lake Water Isotope Balance and Carbonate Sensor Model input/output. Top left: lake surface water temperatures over the ERA-interim reanalysis, output from the Lake Environment sub-model. Top right: $\delta^{18}O$ of precipitation from an isotope enabled GCM over Tanganyika (iCAM5, Nusbaumer et al. (2017)). Bottom left: $\delta^{18}O$ of lake water simulated by isotope mass balance scheme in Environment sub-model. Bottom right: simulated $\delta^{18}O$ of carbonates in lake water from the carbonate sensor sub-model.

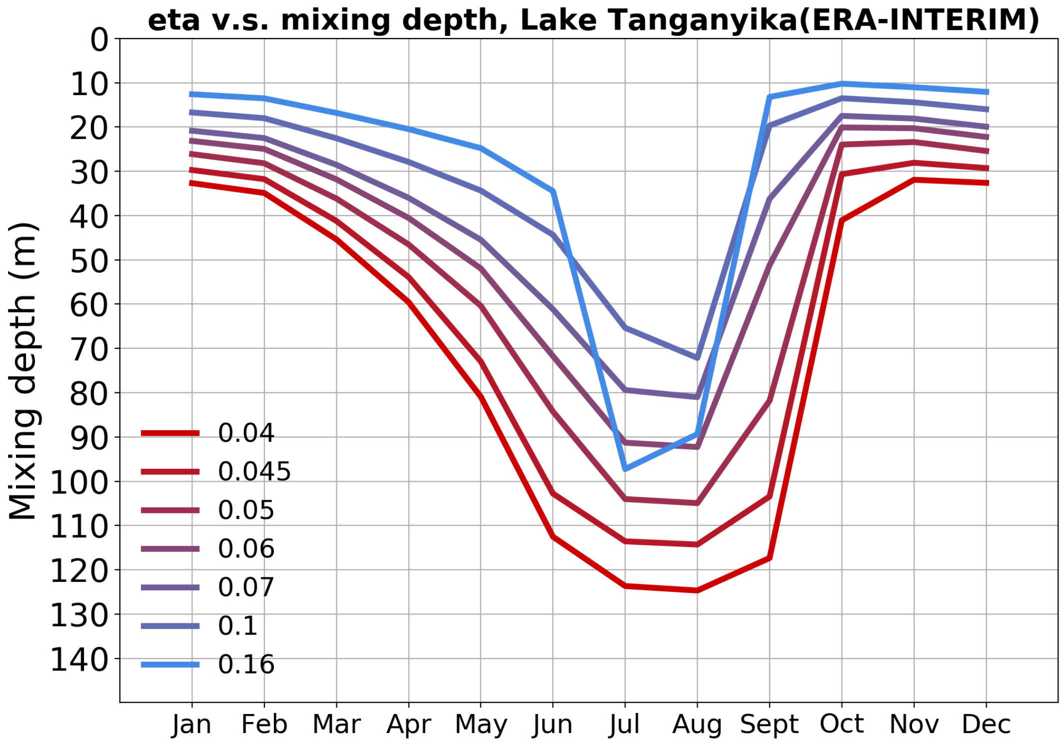

Figure 3: Shortwave Extinction Coefficient (η) indicates water transparency and influences energy distribution in the water column: Different values of η results in changes of lake mixing depth. Based on in situ temperature measurements by Plisnier et al. (1999), we estimate the average mixing depth for Tanganyika to be approximately 50m during the wet season and 80m in the dry season, which best corresponds to the sensitivity test results with $eta = 0.065$.

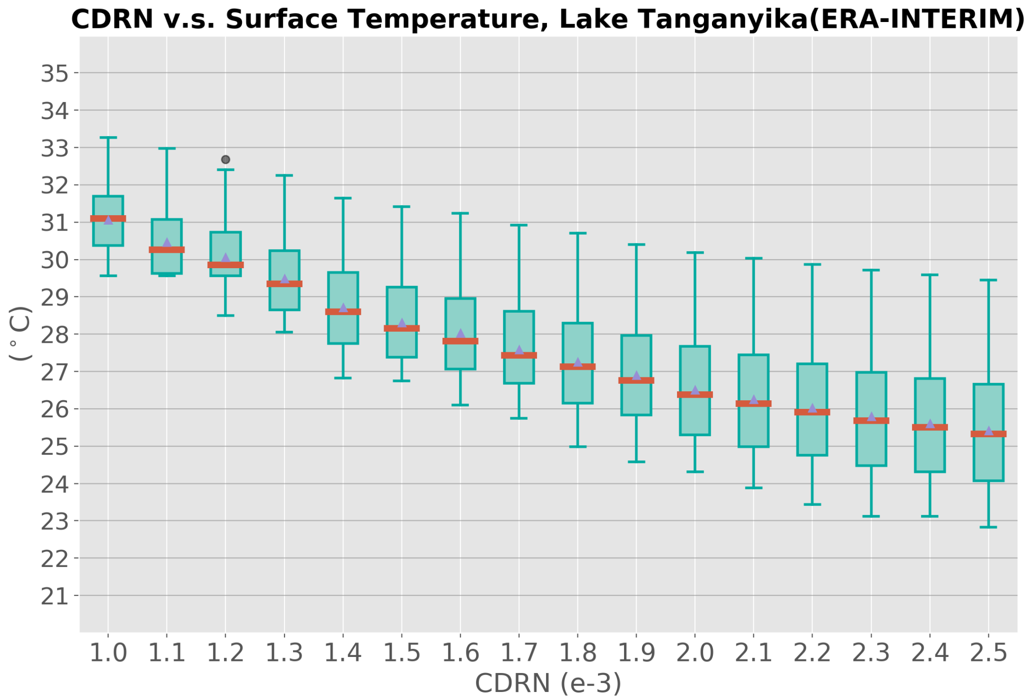

Figure 4: Neutral Drag Coefficient (CDRN) reflects the water surface friction. Our sensitivity experiment suggests its strong correlation between lake surface temperature.The in situ temperature observation from Plisnier et al. (1999) shows the lake surface temperature ranges from $25.8 \pm 0.9°$ during the dry season to $27.8 \pm 0.7°$ during the wet season. Verburg and Hecky (2003) calculates the annual average surface temperature for Lake Tanganyika to be 26.27° in 1994, which best corresponding to the sensitivity tests result as CDRN $= 2.0e - 3$.

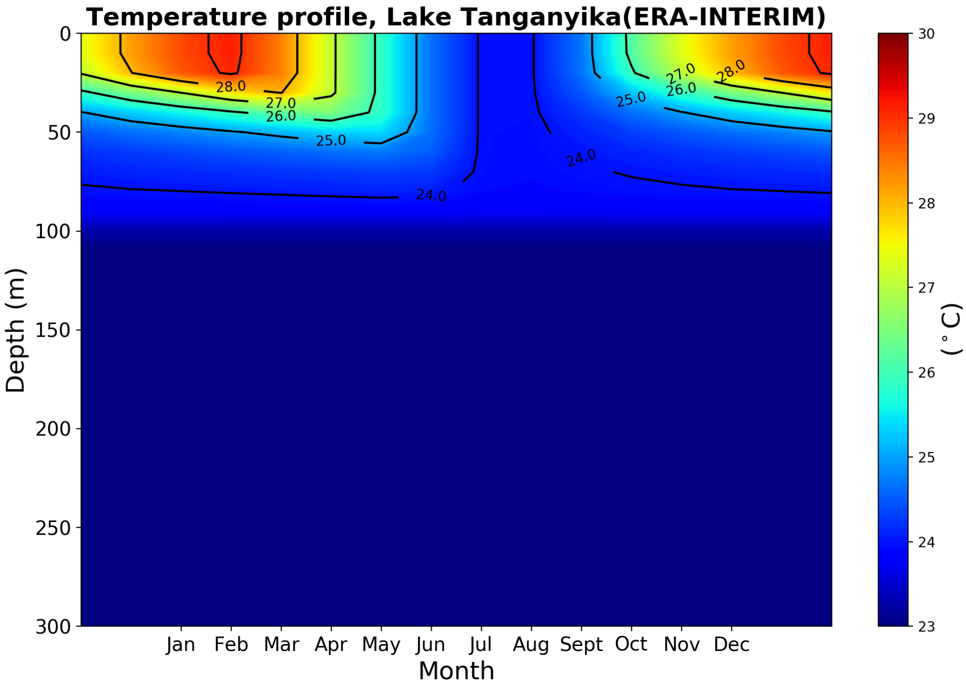

Figure 5: Final tuning result for Lake Tanganyika: Monthly average temperature profile with ERA-INTERIM input, CDRN $= 2.0e-3$, $\eta = 0.065$.

Environment Inputs	**Environment Outputs**
A. Year, Latitude, Longitude	
B. Day of year: ranges from 1 to 366	Julian day (from 1-365)
C. Air temperature at 2 meters (degrees Celsius)	Lake surface temperature (degrees Celsius, averaged over top 1 meter)
D. Relative or Specific humidity at 2 meters (% or g/kg)	Lake evaporation (mm/day)
E. Wind speed at 2 meters (m/s)	Average mixing depth (m)
F. Surface incident shortwave radiation (W/m2)	Maximum mixed layer depth (m)
G. Downward longwave radiation (W/m^2)	Lake depth (m)
H. Surface pressure (mb)	∗Ice fraction (ranges from 0 to 1)
I. Precipitation (mm)	∗Ice height (m), Snow height (m)
J. Basin runoff (mm per unit area of the drainage basin)	∗Lake discharge (m per lake area)
K. $\delta^{18}O$ of precipitation (‰ VSMOW)	∗Average lake $\delta^{18}O$ of upper 1 meter (‰ VSMOW)
L. δD of precipitation (‰ VSMOW	∗Average lake δD of upper 1 meter (‰VSMOW)
M. $\delta^{18}O$ of basin runoff (per mil VSMOW)	∗Actual lake level above/below current 1-meter lake layer (fraction of meters)
N. δD of basin runoff (‰ VSMOW	
Sensor Inputs	**Sensor Outputs**
δD_{precip}, C3:C4 vegetation fraction, ϵ: apparent fractionation	Leaf Wax δD_{wax}
LST, $\delta^{18}O_w$	$\delta^{18}O_{carb}$
LST, calibration β	brGDGT / TEX_{86}
Archive Inputs	**Archive Outputs**
Sed. rate (S), porosity (ϕ)	smoothed sensor time series, revised age-depth model
abu, iso, mxl, numb	bioturbated time series
Observation Inputs	**Observation Outputs**
Bchron-θ, σ_a	Bchron age model ensemble, proxy time series with analytical error ensemble

Table 1: Inputs and Outputs for each PSM submodel in `PRYSM v2.0`. The ∗ for Environment model fields indicate output fields that the model is not currently configured to return, but they can be added to the output via an adjustment of the subroutine `SHUFFLE`. Archive: ABU = series of abundances of carrier type 1 down core, ISO = series of isotopes (or any other sensor) measured on carrier, MXL = series of mixed layer thicknesses down core, NUMB = number of carriers to be measured. Observation: θ: age model uncertainty, positions, calibration curves, and ages (for each tie-point date); σ_a: analytical uncertainty.

References

Benson, L., & White, J. (1994). Stable isotopes of oxygen and hydrogen in the truckee river-pyramid lake surface-water system. 3. source of water vapor overlying pyramid lake. *Limnology and Oceanography*, *39*(8), 1945–1958.

Descy, J., Plisnier, P., Leporcq, B., Sténuite, S., Pirlot, S., Stimart, J., ... others (2006). Climate variability as recorded in lake tanganyika (climlake). *Brussels: Belgian science Policy*.

Garratt, J. (1977). Review of drag coefficients over oceans and continents. *Monthly weather review*, *105*(7), 915–929.

Hostetler, S. (2009). Use of models and observations to assess trends in the 1950–2005 water balance and climate of upper klamath lake, oregon. *Water resources research*, *45*(12).

Hostetler, S., & Benson, L. (1994). Stable isotopes of oxygen and hydrogen in the truckee river–pyramid lake surface-water system. 2. a predictive model of δ18o and 182h in pyramid lake. *Limnology and Oceanography*, *39*(2), 356–364.

Konecky, B., Russell, J., & Bijaksana, S. (2016). Glacial aridity in central indonesia coeval with intensified monsoon circulation. *Earth and Planetary Science Letters*, *437*, 15–24.

Lorenz, D. J., Nieto-Lugilde, D., Blois, J. L., Fitzpatrick, M. C., & Williams, J. W. (2016). Downscaled and debiased climate simulations for north america from 21,000 years ago to 2100ad. *Scientific data*, *3*, 160048.

Nusbaumer, J., Wong, T. E., Bardeen, C., & Noone, D. (2017). Evaluating hydrological processes in the community atmosphere model version 5 (cam5) using stable isotope ratios of water. *Journal of Advances in Modeling Earth Systems*, *9*(2), 949–977.

Plisnier, P.-D., Chitamwebwa, D., Mwape, L., Tshibangu, K., Langenberg, V., & Coenen, E. (1999). Limnological annual cycle inferred from physical-chemical fluctuations at three stations of lake tanganyika. In *From limnology to fisheries: Lake tanganyika and other large lakes* (pp. 45–58). Springer.

Sachse, D., Billault, I., Bowen, G. J., Chikaraishi, Y., Dawson, T. E., Feakins, S. J., ... others (2012). Molecular paleohydrology: interpreting the hydrogen-isotopic composition of lipid biomarkers from photosynthesizing organisms. *Annual Review of Earth and Planetary Sciences*, *40*.

Strub, P., & Powell, T. M. (1987). The exchange coefficients for latent and sensible heat flux over lakes: dependence upon atmospheric stability. *Boundary-Layer Meteorology*, *40*(4), 349–362.

Trauth, M. H. (2013). Turbo2: A matlab simulation to study the effects of bioturbation on paleoceanographic time series. *Computers & geosciences*, *61*, 1–10.

Verburg, P., & Hecky, R. E. (2003). Wind patterns, evaporation, and related physical variables in lake tanganyika, east africa. *Journal of Great Lakes Research*, *29*, 48–61.