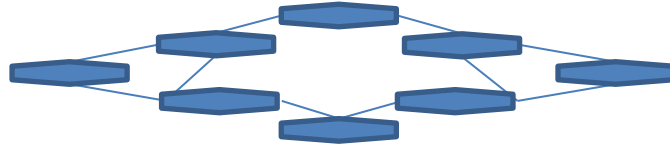


NETLAKE



NETLAKE toolbox for the analysis of high-frequency data from lakes

Working Group 2: Data analysis and modelling tools

October 2016



Suggested citation for the complete set of factsheets: Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes*. Technical report. NETLAKE COST Action ES1201. 60 pp. <http://eprints.dkit.ie/id/eprint/530>

Editors:

Biel Obrador. Department of Ecology, University of Barcelona. Av. Diagonal 645, Barcelona, Spain. obrador@ub.edu

Ian Jones. Centre for Ecology and Hydrology, Lancaster Environment Centre, Library Avenue, Bailrigg, Lancaster, United Kingdom. ianj@ceh.ac.uk

Eleanor Jennings. Centre for Freshwater and Environmental Studies, Department of Applied Science, Dundalk Institute of Technology, Dublin Road, Dundalk, Ireland.
eleanor.jennings@dkit.ie

Author affiliations:

Rosana Aguilera. Catalan Institute of Water Research, Girona, Spain.

Louise Bruce. Aquatic Ecodynamics Research Group, University of Western Australia, Perth, Australia.

Jesper Christensen. Institute of Bioscience, Aarhus University, Roskilde, Denmark.

Raoul-Marie Couture. Norwegian Institute for Water Research, Norway.

Elvira de Eyto. Burrishoole research station, Marine Institute, Ireland.

Marieke Frassl. Department of Lake Research, Helmholtz Centre for Environmental Research, UFZ, Magdeburg, Germany.

Mark Honti. Budapest University of Technology and Economics, Hungary.

Ian Jones. Centre for Ecology and Hydrology, Lancaster Environment Centre, Library Avenue, Bailrigg, Lancaster, United Kingdom.

Rafael Marcé. Catalan Institute of Water Research, Girona, Spain.

Biel Obrador. Department of Ecology, University of Barcelona, Barcelona, Spain.

Dario Omanović. Ruđer Bošković Institute, Zagreb, Croatia.

Ilia Ostrovsky. Israel Oceanographic & Limnological Research, Kinneret Limnological Laboratory, P.O.B. 447, Migdal 14950, Israel.

Don Pierson. Lake Erken field station, Uppsala University, Sweden.

Ivanka Pižeta. Ruđer Bošković Institute, Zagreb, Croatia.

Friedrich Recknagel. University of Adelaide, School of Biological Sciences, Adelaide, Australia.

Peter A. Staehr. Institute of Bioscience, Aarhus University, Roskilde, Denmark.

Koji Tominaga. Norwegian Institute for Water Research, Norway.

Michael Weber. Department of Lake Research, Helmholtz Centre for Environmental Research, UFZ, Magdeburg, Germany.

R. Iestyn Woolway. University of Reading, United Kingdom.

Table of Contents

Introduction.....	1
Factsheet 1 – Data handling: cleaning and quality control. <i>Elvira de Eyto and Don Pierson</i>	2
Factsheet 2 – Lake Heat Flux Analyzer (LHFA). <i>Ian Jones</i>	7
Factsheet 3 – The General Lake Model (GLM). <i>Marieke Frassl, Michael Weber and Louise Bruce</i>	11
Factsheet 4 – Lake Metabolizer. <i>R. Iestyn Woolway</i>	16
Factsheet 5 – High Frequency data treatment and visualization with ECDSOFT and OnLineMonitor. <i>Dario Omanović and Ivanka Pižeta</i>	23
Factsheet 6 – Lake stratification and ice phenology: modelling with MyLake. <i>Raoul-Marie Couture and Koji Tominaga</i>	28
Factsheet 7 – Knowledge Discovery in Databases - Data Mining. <i>Ivanka Pižeta</i>	35
Factsheet 8 – Bayesian calibration of mechanistic models of lake metabolism. <i>Mark Honti</i>	40
Factsheet 9 – Determination of whole-column metabolism from profiling data. <i>Biel Obrador, Jesper Christensen and Peter A. Staehr</i>	47
Factsheet 10 – Pattern detection using Dynamic Factor Analysis (DFA). <i>Rosana Aguilera and Rafael Marcé</i>	52
Factsheet 11 – Inferential modelling of time series by evolutionary computation. <i>Friedrich Recknagel and Ilia Ostrovsky</i>	57

Introduction

NETLAKE toolbox for the analysis of high-frequency data from lakes

With the advent and proliferation of high frequency *in situ* data collection from lakes has come the need to process unprecedented quantities of data in a useful and effective manner. This need has driven the development, or adoption, of a variety of techniques, programs and methodologies for working with high frequency lake data. It was, therefore, thought timely to provide an easily accessible and digestible synopsis of some of these topics. Discussions between interested members of the NETLAKE COST Action (ES1201) and a poll of Action members led to the identification of a range of such topics by the community which was felt to be of broad potential interest to those collecting high frequency data from lakes, and indeed rivers. Individual specialists were identified for each of these topics with each then writing a 'factsheet' intended as a beginner's guide to the topic. The intention was to briefly describe the objective of the method, a specific application for it, some details of the background knowledge and data requirements necessary for its use, and a broad description of how the method should be implemented. Additionally, some advice, in the form of tips, where to find further information and how to access any code was included. These factsheets were peer reviewed by experts within NETLAKE, then edited and collated. The factsheets can be downloaded individually or collectively.

The factsheets were developed within Working Group 2 (Data analysis and modelling tools) of the NETLAKE COST Action (ES1201), supported by COST (European Cooperation in Science and Technology). NETLAKE ran between 2012 and 2016.

NETLAKE toolbox for the analysis of high-frequency data from lakes



Factsheet #1

Data handling: cleaning and quality control

Elvira de Eyto and Don Pierson

Objective

The objective here is to describe some of the procedures that can be used to process high frequency monitoring (HFM) data to ensure that obvious errors have been removed and that data can be considered quality controlled. Some examples from two long running monitoring stations are discussed.

Specific application

HFM brings immediate gratification in the form of megabytes of data but without quality assurance /quality control (QA/QC) procedures, the confidence in these data will be reduced. Some variables require less care than others, but all variables need to be checked and verified, particularly if the data are being used externally and/or shared openly. Quality indicators are a useful way of informing users to what level QA/QC has taken place. Here is one example, which has been developed for the Lake Erken monitoring station:

Level 0: data straight off the logger as ASCII text files. It is critical that level 0 data is always archived for future reprocessing use.

Level 1: Checked to ensure that all expected time steps or file rows are in the file, even if they only contain missing values. Obvious outliers have been marked or removed and some maintenance log comments added. This is usually the minimum processing before sharing of data.

Level 2: Data are corrected for drift, sensor calibration, compared with neighbouring sensors and corrected accordingly. This level of analysis can be done on a by-needs basis by researchers working with the data, or can be part of a more regular QA/QC program. Typically level 2 processing requires supplementary information such as sensor calibration data.

In some instances, getting data from level 0 to level 1 can be done manually, with checks and comment additions done by experienced personnel. However, some of these processes can be automated. A great advantage to automated processing is that as QA/QC algorithms are improved the level 0 data can easily be reprocessed to an improved level 1 state.

Background

This process is quite specific to each monitoring station, and much will depend on how and where data are stored, whether there are in-built quality checks, and what the intended use of the data is. The specific steps described below are what is carried out at the Irish Marine Institute's research facility in Burrishoole, where five automatic monitoring stations are maintained. Raw data are transmitted from the stations' data loggers via GPRS every couple of minutes to a computer in the research lab. Owing to security issues with institutional firewalls (which is a common problem when transmitting data), this computer is isolated from the main institutional servers. Approximately once a month, these data are copied across to an intermediate storage home on a server which is backed up regularly.

A crucial decision to make at this stage is how you want to store the HFM data long term. In Burrishoole we have decided to store the HFM data at level 1, after a couple of fairly basic checks (described below) and additions. We also store all the level 0 files. Thus, the permanently stored files (which are saved on a SQL server) contain data which have had no data deletion or manipulation. If a data request is lodged with us, the requestor receives these data, with the caveat that the data are at level 1 (see above). We decided that this was the way for us to store and share data after observing how subjective data cleaning and manipulation was. What one person considers to be "bad" data (and may delete from the file), another person, in hindsight, may think is retrievable. This is an important consideration when the intention is to run the station for years or decades, and where it is likely that the staff in charge of the data are likely to change. Of course, where significant data cleaning has been carried out on a particular variable, these data (level 2) will also be stored (and shared if applicable), but in a separate location to the long term level 1 data storage.

Type of data and requirements

Any remotely collected data needs some QA/QC before use. Software that is useful includes MS excel, R, Python, B3 (<https://www.lernz.co.nz/tools-and-resources/b3>) or Hydras (<http://www.ott.com/products/software-solutions/ott-hydras-3-basic/>).

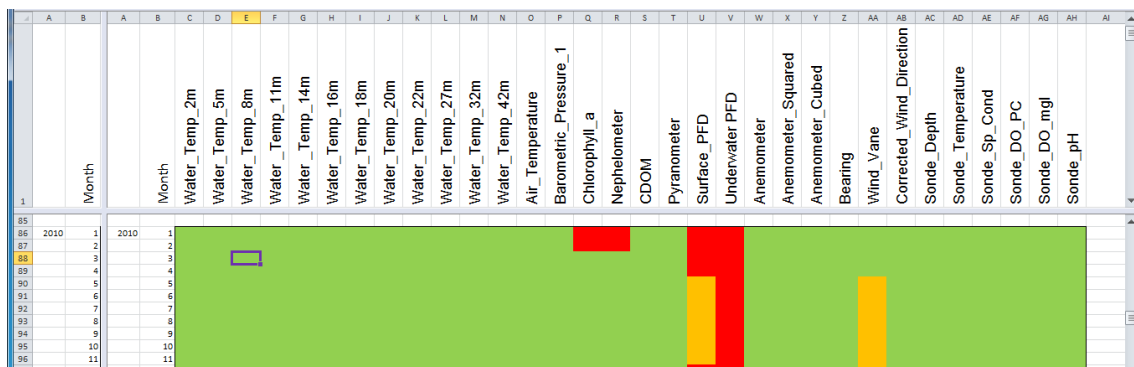
Basic procedures

Here we describe how we bring the Lough Feeagh AWQMS (automatic water quality monitoring station) data from **level 0 to level 1**. This is provided only as an example, and should be modified according to your requirements.

1. Join text files containing chunks of data together into an annual file. When data are stored every 2 minutes, an annual file is still manageable with any spreadsheet like Excel for example (262,000 rows). Once you go past this resolution (e.g. minute files, or multiple years), Excel is no longer useable, and manipulation and data viewing probably needs to take some other route.
2. Check for missing time steps and fill in where needed. This is only essential if you expect data to have a regular time step. Having a regular time step makes it easier to aggregate

data from different stations and sensors later, but is not essential. Some useful ways of doing this include:

- Use a pivot table in Excel, with day or date being the aggregating variable (720 measurements per day, 262,800 per year). A day with less than 720 values is easy to spot.
 - The Zoo package in R has a function for merging a pre-described time step with a dataset where there are gaps.
3. Fill in blanks with NA. This is for later use in R. Other programs may require blanks to be coded differently (e.g. -999, or simply blank).
 - This can be done in Excel – fill blanks or replace, but can be slow if there are a lot of them.
 4. Check for outliers
 - We use the filter in Excel. If an outlier is spotted, we will add a note to that row rather than deleting the value.
 5. We fill in comments retrospectively. An extra column is added to the data sheet, and comments are transferred from our field book to the relevant row or rows. The comments might include:
 - Sensors cleaned.
 - Sonde out for calibration.
 - Fluorometer removed for service.
 - Mooring rope replaced.
 - Anemometer looks very high. Check against the manual weather station before use.
 - Batteries flat.
 6. We normally extract a subset of data for each day (e.g. 06:00, 12:00, 18:00, 00:00)
 - Do some quick graphs of what things look like. For this purpose simple Excel templates can be prepared into which data is pasted and graphs are automatically generated.
 - Do an informal report on what the year's data look like.
 - Fill in the sensor information plot – this is a simple excel log giving a quick overview of which sensors were working at any time period.



7. Based on the summary plots and the informal report, make some additional comments to the level 1 data.

8. Upload this level 1 data to our long term storage SQL server.

The next step is to get data from **level 1** to **level 2**, which is not done routinely, but on a project basis as data are requested.

1. Make a copy of the level 1 data which can then be changed or manipulated.
2. There are a couple of programs which can be used at this stage; in Burrishoole, we use Hydras or R. Another option is B3. Excel is useful also for some things. At this stage, we would do things like:
 - fill in sensor gaps with interpolated data from another source if applicable
 - Apply temperature corrections to CDOM sensor data
 - Correct data points for sensor drift (e.g. CDOM, chl fluorometer, DO)
 - Do a more detailed analysis for outliers
 - Aggregate the data where required. A useful tool for this is the R package hydroSTM
 - Compare one sensor against another to check for drift or odd values (e.g. multiple thermistors)
 - Remove “bad” data and replace with NA

Pitfalls and tips

- Losing data. This is very common, as files multiple up very quickly. Have a structured folder system, perhaps ordered by site and year, or by sensor. File names should have some logical meaning and be consistent so that they can be sorted as required.
- Overwriting raw data. You might feel the need to correct values as you are sure they are outliers. Then you discover that actually the data were fine, and were, for example, recording an episodic event. Always keep an original version of the level 0 data. Any data manipulations should be made on a copy of these data, and only change data in the copy. The logger text files (level 0) are usually quite small, and you can always store these logger files, along with level 1 and level 2 data.
- Overzealous data cleaning....leaving you with no data!
- Thinking a sensor is working (because the values are changing), but subsequently realising that the logger is just recording some residual current.
- Mixing up your data files from different sites, or different times of data collection. We recommend creating a variable in your data logger program to identify the data logger location and the data logger program version. These variables can be outputted on a daily basis along with other diagnostic information such as logger battery voltage. Having the site and the program that created a file documented in the file itself will prevent location mix-up and will also be of use in linking data changes to program changes.

Further reading

1. <http://www.ott.com/products/software-solutions/ott-hydras-3-basic/>
2. <https://www.lernz.co.nz/tools-and-resources/b3>
3. <https://cran.r-project.org/web/packages/zoo/zoo.pdf>
4. <https://cran.r-project.org/web/packages/hydroTSM/hydroTSM.pdf>
5. <http://www.gleon.org/data/best-practices>

Contact details

Elvira de Eyto. Burrishoole research station, Marine Institute, Ireland.
elvira.deeyto@marine.ie

Don Pierson. Lake Erken field station. Uppsala University, Sweden.
don.pierson@ebc.uu.se

Suggested citation

de Eyto, E. and Pierson, D. 2016. Data handling: cleaning and quality control. In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 1). Technical report. NETLAKE COST Action ES1201. pp. 2-6.
<http://eprints.dkit.ie/id/eprint/532>.

Acknowledgement

This factsheet is based upon work from the NETLAKE COST Action ES1201, supported by COST (European Cooperation in Science and Technology).

NETLAKE toolbox for the analysis of high-frequency data from lakes



Factsheet #2

Lake Heat Flux Analyzer (LHFA)

Ian Jones

Objective

Lake thermal and mixing properties are mainly driven by fluxes of heat and wind mixing at the surface of a lake. There are several different types of heat fluxes. With the right equipment these can be measured, but such equipment can be expensive and requires expertise to deploy. As an alternative to direct measurement most of these fluxes are often calculated with established methods using the meteorological variables commonly measured by automatic lake monitoring stations. These methods can be quite detailed and require some specialist knowledge to execute. The software tool, Lake Heat Flux Analyzer (LHFA), has been written to enable the calculation of these fluxes, and related terms, from standard meteorological variables. It has been specifically written for those using data from high resolution monitoring stations on lakes. The principal fluxes calculated are Q_{sr} , the reflected short-wave radiation; Q_{in} , the incoming flux of long-wave radiation; Q_{out} the outward flux of long-wave radiation; Q_h , the sensible heat flux, driven by temperature differences between water and the overlying air; Q_e , the latent heat flux, driven by moisture differences between water and the overlying air; and Q_{tot} , the total heat flux. In addition, the software calculates transfer coefficients at the measurement height and calculates transfer coefficients, wind speed, relative humidity and air temperature at the standard reference height of 10 m, including their values for a neutral atmosphere approximation. The software tool can also be used if the meteorological data are collected over land, but the results will have some additional inaccuracies.

Specific application

Some example output for incoming and outgoing long-wave radiation and incoming and reflected short-wave radiation for 2004, calculated from data taken on a monitoring buoy at Esthwaite Water, UK, are shown in Figure 1. These data were calculated using LHFA, downloaded from the web and subsequently read into an excel file.

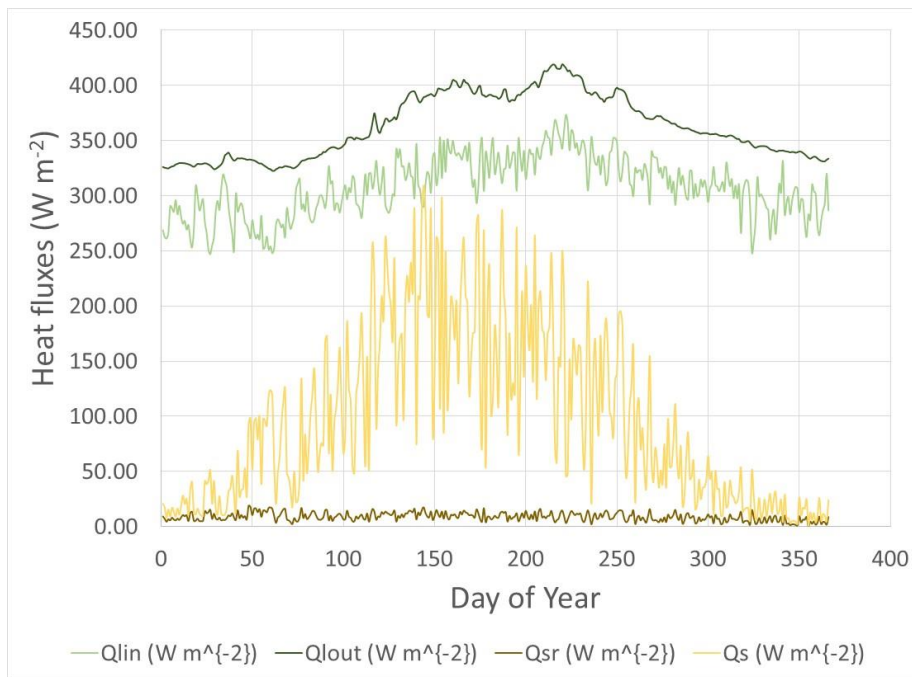


Figure 1. Radiative surface fluxes for Esthwaite Water, UK, 2004.

Background

The software can be used directly through a web-interface or the Matlab code can be freely downloaded (see link below). Some experience in setting up files and changing formats, specifically the date/time format, is required to utilise the web-version. While the tool can be used without having any prior knowledge of heat fluxes, interpretation of the results does require some understanding of the fluxes.

Type of data and requirements

The required inputs for LHFA are wind speed, air temperature, relative humidity, surface water temperature and either short-wave radiation or Photosynthetically Active Radiation (PAR), which LHFA will convert to short-wave if necessary. In addition, the measurement heights are required for wind speed, air temperature and relative humidity. Formatting is detailed in the user manual. Note that the formatting must be followed exactly. Example files are provided on the web-page (see link below).

Basic procedures

The procedure to follow is detailed in the user manual available on the web-page (link below). Only a brief synopsis is given here:

1. Collect and clean high frequency data for wind speed, air temperature, relative humidity, solar radiation and surface water temperature. Note that data must be in numeric Matlab format and missing values denoted by NA will result in an error.

2. Go to the LHFA web-site and reformat data exactly according to the instructions in the user manual and the example datasets, including collecting the data files in a single zipped folder.
3. Either download and use the Matlab version of the code, which can then be executed in Matlab, or upload the zipped data folder to the web-site.
4. Submit file. The program takes a while to run, depending on the size of the dataset and the internet connection.
5. Download the output data.

Pitfalls and tips

- Many different formulae have been developed to calculate heat fluxes. LHFA uses a set of specific established fluxes, but this is not an endorsement that these are the only or the most accurate formulae. Furthermore, the study of air-water fluxes is ongoing and new advances, not captured by this software, are likely to continue.
- Inputting data to the web interface requires very specific formatting. Whilst this formatting is described in detail in the user manual and examples are downloadable from the web-page, it can be frustrating for users to ensure their data are in the exact format required.
- One of the most common reasons for the program to fail is if the date/time is in the incorrect format or if there are strings such as 'missing' in the data. The user should instead leave that particular cell blank or include NaN, which indicates missing data.
- There is no absolute convention for describing the sign of heat fluxes. The LHFA paper describes the direction of fluxes calculated in the software, but, nevertheless, it is still easy for an inexperienced user to get confused over the direction of each of the fluxes.
- The software performs the complex calculations required for determining surface fluxes, but unless users are familiar with typical values, flux directions and meanings of fluxes, misinterpretation of results is easy.
- Any bulk formulae calculations of heat fluxes are subject to error. Results should therefore be interpreted as estimates, rather than exact values.
- Lakes modify the overlying air temperature, relative humidity and wind speed. Fluxes calculated using any land-based measurements will therefore suffer from additional inaccuracy.
- The program searches for consistent times among the different variables, wind speed, air temperature, and so on. Therefore, if the variables have slightly different times (e.g. seconds or minutes), these will not be used in the calculation. Users must ensure consistent times among the variables.

Further reading

Key References:

The reference for the paper describing the code and its uses is:

Woolway, R.I., Jones, I.D., Hamilton, D.P., Maberly, S.C., Muraoka, K., Read, J.S., Smyth, R.L., Winslow, L.A. 2015. Automated calculation of surface energy fluxes with high-frequency lake buoy data. *Environmental Modelling and Software* 70: 191–198.

Other useful references:

For an example of heat fluxes being calculated and used see:

Woolway, R.I., Jones, I.D., Maberly, S.C., Feuchtmayr, H. 2015. A comparison of the diel variability in epilimnetic temperature for five lakes in the English Lake District. *Inland Waters* 5: 139–154.

The sister paper to LHFA for calculating in-lake physics parameters is:

Read, J.S., Hamilton, D.P., Jones, I.D., Muraoka, K., Kroiss, R., Wu, C.H., Gaiser, E. 2011. “Lake Analyzer”: Derivation of lake mixing and stratification indices from high-resolution lake buoy data. *Environmental Modelling and Software* 26: 1325–1336.

For a useful background on lake physics see:

Imberger, J., Patterson J.C. 1990. Physical Limnology. *Advances in Applied Mechanics* 27: 303–475.

Code

The Matlab version of the software, the web-interface, and the user manual can all be found at: <http://heatfluxanalyzer.gleon.org/>

Contact details

Ian Jones. Centre for Ecology and Hydrology, UK.

ianj@ceh.ac.uk

Suggested citation

Jones, I.D. 2016. Lake Heat Flux Analyzer (LHFA). In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 2). Technical report. NETLAKE COST Action ES1201. pp. 7-10. <http://eprints.dkit.ie/id/eprint/533>.

Acknowledgement

This factsheet is based upon work from the NETLAKE COST Action ES1201, supported by COST (European Cooperation in Science and Technology).

NETLAKE toolbox for the analysis of high-frequency data from lakes



Factsheet #3

The General Lake Model (GLM)

Marieke Frassl, Michael Weber and Louise Bruce

Objective

Numerical modelling of lakes and reservoirs provides opportunities for addressing research questions beyond those possible with observational monitoring data alone. This is the case when the scientific or management questions are about the ecosystem state in the future, or when a higher resolution in space or time is needed than observed by monitoring. Lake models can serve as tools providing, for example, near real-time forecasting of water quality, scenario analyses of changed external drivers (e.g. climate change) or simulated data on the outcome of different management strategies.

The General Lake Model (GLM) is a one-dimensional hydrodynamics model. Hydrodynamic models describe the thermal properties and the mixing dynamics in water bodies. Based on inflow and outflow data, as well as meteorological data, GLM calculates a water and energy balance resulting in vertical profiles of temperature, salinity and density over time. As a one-dimensional model, GLM simulates the vertical profiles at one spatial point in the lake. Effects of ice cover on thermal properties and mixing of the lake can be included. GLM can also be coupled to biogeochemical models (e.g. AED, FABM), and therefore serves as the basis for models simulating the biological and chemical parameters in the water column. Data from monitoring stations are used as input data and to calibrate and validate the lake model. In combination with the observed data, GLM can be used to explore the role that stratification and vertical mixing play on the dynamics of lakes.

Specific application

The main usage of GLM is to provide simulated physical data that are coupled to an ecological model to simulate water quality. However, GLM can be used as a stand-alone tool such as using it to assess the outcome of different management strategies on the thermal structure of a reservoir. Figures 1 and 2 show the GLM output for a two-year simulation of the Grosse Dhuenn Reservoir, Germany. Note the change in simulated water level, and water temperature over time.

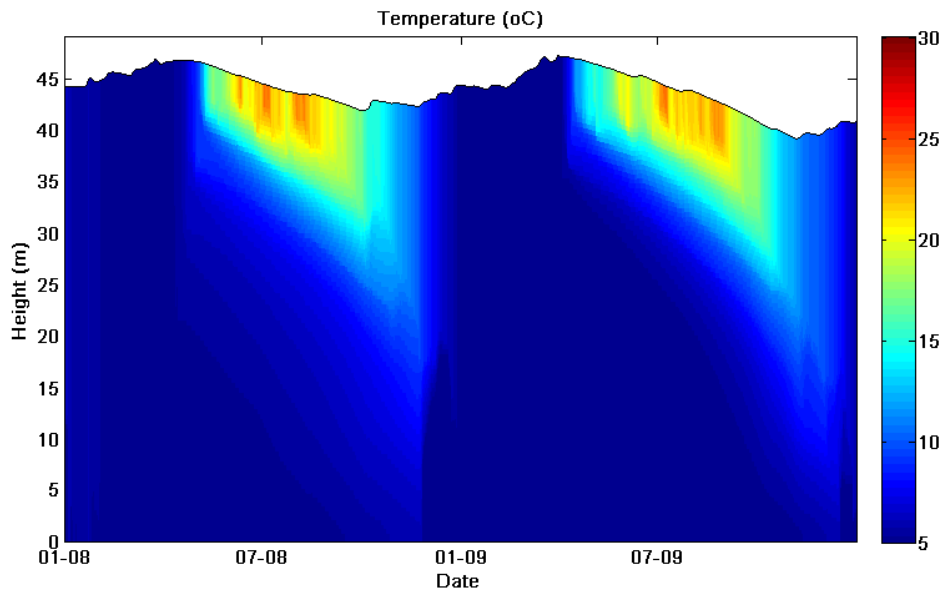


Figure 1. Simulated water temperature in the Grosse Dhuenn Reservoir, Germany (2008-2009).

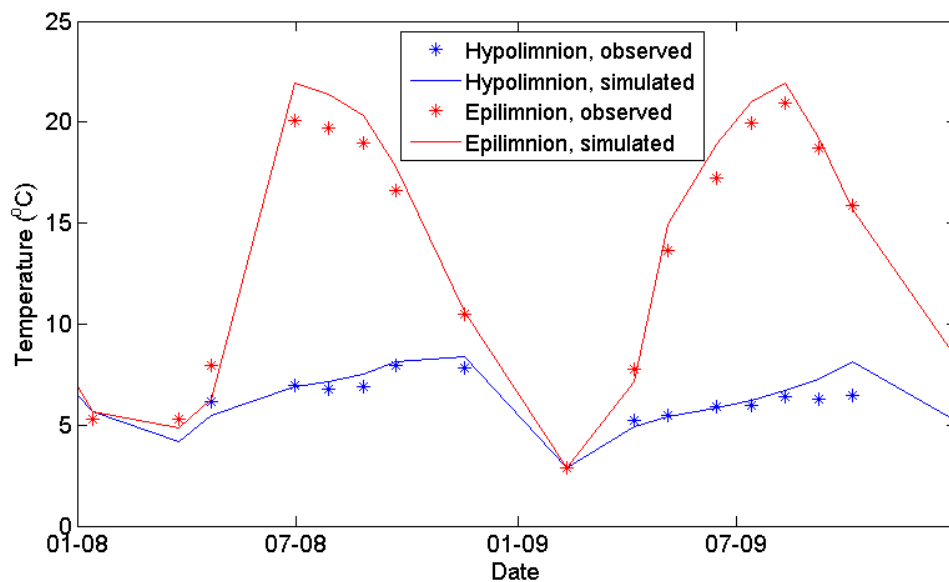


Figure 2. Comparison of simulated and observed epi- and hypolimnetic temperatures in the Grosse Dhuenn Reservoir, Germany (2008-2009).

Background

The model is written in 'C'. Compiled executables for MS Windows, Mac OS or Linux are freely available at the GLM webpage (see link below). To run a simulation, GLM can be called from the command window. Alternatively, the R-package GLMr can be used to run the model and plot the output. Some knowledge in 'R' or 'Matlab' is helpful to prepare the input files and to analyse the output. A basic knowledge of modelling techniques such as calibration and validation procedures is recommended before applying the model (Bennet et al. 2013). Calibration uses measured data to find the optimum values for selected model parameters;

validation uses different measured data to allow an assessment of model performance for an independent time period to that used to calibrate the model.

Type of data and requirements

GLM input data are specified in .txt and .csv files. The following data are required:

- Morphometric data of the lake as a hypsographic curve. These are in the form of two columns of data, one with depth (m) and the second with the area of the lake at that depth (m²).
- The extinction coefficient (1/m), averaged over the simulated time period, e.g. estimated from Secchi depth measurements.
- Meteorological data (mean air temperature (°C), mean wind speed (m/s), mean shortwave radiation (W/m²), mean longwave radiation (W/m²) or cloud cover (fraction coverage), mean relative humidity (%), total rainfall (m/day) in an hourly or daily resolution.
- Inflow and outflow data in a daily resolution: volume (m³/s), salinity (psu) and temperature (°C).
- An initial temperature profile, including the depths of each temperature measurement.

The format and file structure needed to run the model are described in depth in the user manual (see link below). Example files and simulations are provided on the webpage.

For the model validation, a time series of temperature measurements at different depths is needed. The time resolution of the calibration and validation data is not fixed. However, a higher resolution generally reduces the uncertainty in model prediction.

Basic procedures

A detailed description of how to set up and run the model is given in the manual (see link below). The basic steps are:

1. Optional: Run one of the example simulations.
2. Collect and clean-up input data and prepare the input files for your lake (.csv file with meteorological data, .csv files with inflow and outflow data, a master .nml file including lake morphometry and lake location, run time set up and initial conditions).
3. Check the format and units of your input data.
4. Split your input and monitoring data into two time frames.
5. Run the model for the first part of your available data.
6. Compare model results and observations and calibrate model parameters.
7. Use the second part of your available data to validate the model, i.e. compare observations and model results without further calibrating the model. For different metrics to quantify model fit see e.g. Bennet et al. (2013).
8. Optional: Use your model set-up to run scenarios.

Pitfalls and tips

- GLM is a one-dimensional model and is based on the assumption that variation in the vertical direction is more important than the horizontal direction. Check if this assumption applies to your lake or reservoir.
- If your simulation does not start, or shows strange results, check the file format (especially the date format) and the units of your model input (e.g. Kelvin instead of degrees Celsius; mm instead of m – see manual). Wrong units or a wrong format are the main error source and it is worth checking the model input carefully.
- To avoid frustration due to a wrong format of the input files, it is very helpful to start with the example files and to exchange those files step by step with your own data.
- GLM is an open-source community model, therefore different “sources for help” exist from which you can benefit. If you have a question, start by checking the AEMON forum, where model users actively discuss problems and offer solutions (see link below).
- To calibrate and validate the model results a range of metrics and patterns should be analysed such as the epilimnion and hypolimnion temperature, thermocline depth, stratification onset. For an example see Figure 2.
- The R-packages *glmtools* (see link below) and *Lake Analyzer* (Read et al. 2011), which are available in ‘R’ and ‘Matlab’, are useful software packages to analyse the model output.

Further reading

GLM Manual:

Hipsey, M.R., Bruce, L.C., Hamilton, D.P. 2014. GLM – General Lake Model, Model Overview and User Information (<http://aed.see.uwa.edu.au/>)

Applications of GLM:

Read, J.S., Winslow, L.A., Hansen, G.J.A., Van Den Hoek, J., Hanson, P.C., Bruce, L.C., Markfort, C.D. 2014. Simulating 2368 temperate lakes reveals weak coherence in stratification phenology. *Ecological Modelling* 291: 142-150.

Yao, H., Samal, N.R., Joehnk, K.D., Fang, X., Bruce, L.C., Pierson, D.C., Rusak, J.A. James, A. 2014. Comparing ice and temperature simulations by four dynamic lake models in Harp Lake: past performance and future predictions. *Hydrological Processes* 28: 4587-4601.

Calibration Techniques:

Bennett, N.D., Croke, B.F.W., Guariso, G., Guillaume, J.H.A., Hamilton, S.H., Jakeman, A.J., Marsili-Libelli, S., Newham, L.T.H., Norton, J.P., Perrin, C., Pierce, S.A., Robson, B., Seppelt, R., Voinov, A.A., Fath, B.D., Andreassian, V. 2013. Characterising performance of environmental models. *Environmental Modelling & Software* 40: 1-20.

Code

The model is written in 'C'. Executables are available at:

<http://aed.see.uwa.edu.au/research/models/GLM/>

The R-package *GLMr*, which can be used to run the model and analyse the output is available at: <https://github.com/GLEON>. The package *glmtools* is available at <https://github.com/USGS-R/glmtools>

Contact details

Lead developers and project owners:

Louise Bruce and Matt Hipsey, Aquatic Ecodynamics Research Group, University of Western Australia, Perth, Australia (<http://aed.see.uwa.edu.au/>)
louise.bruce@uwa.edu.au and matt.hipsey@uwa.edu.au

GLM users:

Marieke Frassl and Michael Weber, Department of Lake Research, Helmholtz Centre for Environmental Research, UFZ, Magdeburg, Germany (www.ufz.de)
marieke.frassl@ufz.de and michael.weber@ufz.de

R tools and Open Source leads:

Jordan Read and Luke Winslow, USGS Center for Integrated Data Analytics, Wisconsin, USA
lwinslow@usgs.gov and jread@usgs.gov

AEMON forum: <https://groups.google.com/forum/#!forum/aquaticmodelling>

Suggested citation

Frassl, M., Weber, M. and Bruce, L. 2016. The General Lake Model (GLM). In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 3). Technical report. NETLAKE COST Action ES1201. pp. 11-15.
<http://eprints.dkit.ie/id/eprint/534>.

Acknowledgement

This factsheet is based upon work from the NETLAKE COST Action ES1201, supported by COST (European Cooperation in Science and Technology).

NETLAKE toolbox for the analysis of high-frequency data from lakes



Factsheet #4

Lake Metabolizer

R. Iestyn Woolway

Objective

Metabolism is a fundamental ecological process that occurs at scales ranging from individual organisms to whole ecosystems. Whole ecosystem metabolism represents the balance between carbon fixation (gross primary production; GPP) and biological carbon oxidation (ecosystem respiration; R) in an ecosystem. At an ecosystem scale, metabolism estimates provide insight into the support of food webs through primary productivity, rates of carbon accumulation or loss in an ecosystem, and anticipating changes in ecosystem state. Lake metabolism can be estimated from high frequency free-water dissolved oxygen (DO) concentrations (e.g., Staehr et al. 2010). The value of quantifying lake metabolism and the availability of the necessary data has led to a rapid proliferation of computational methodologies for estimating metabolism. While technological advances in automated sensors and the expansion of cross-site collaborations have increased greatly the accessibility of high frequency DO time series, barriers are presented by the statistics, programming, and multitude of models used to convert sensor observations into estimates of lake metabolism. This analytical barrier may be overcome by the use of a new RPackage called Lake Metabolizer, which is designed to estimate lake metabolism from commonly collected sensor data.

Lake Metabolizer is an Rpackage for estimating lake metabolism and related terms from data collected by high frequency, *in situ* lake monitoring stations with relative ease. The package can be used to calculate lake metabolism using five different methods: bookkeeping, ordinary least squares, maximum likelihood, Kalman filter, and Bayesian (Table 1). For further information of the differences between the metabolism models, see Winslow et al. (*in press*) and Honti (2016). In addition, each of these five methods can be combined with one of seven models for computing the gas transfer coefficient, which influences the rate of gas exchange at the air-water interface. Lake Metabolizer also includes a number of functions that compute conversions and calculations that are commonly applied to raw data prior to estimating lake metabolism (e.g. optical conversion models). This package contains example data, example use-cases, and function documentation.

Model	Underlying statistics	Error structure	Error type
Bookkeeping	Algebra	None	None
Bayesian	Bayesian	Gaussian	Process and Observation
Kalman filter	Maximum likelihood and Kalman filter	Gaussian	Process and Observation
Maximum likelihood	Maximum likelihood	Gaussian	Process and observation
Ordinary least squares	Linear regression	Gaussian	Observation

Table 1. Table comparing the structure of the five different metabolism models included in LakeMetabolizer.

Specific application

The main application of this program is the calculation of lake metabolism using a number of different approaches published in the scientific literature. An example is the calculation of net ecosystem production (NEP), which is the difference between GPP and R, and is used to delineate heterotrophic systems (negative NEP) from autotrophic systems (positive NEP). Example output calculations for NEP and the gas transfer coefficient (k600, which estimates the amount of gas exchange at the air-water interface) for Sparkling Lake are shown in Figures 1 and 2 below. The example dataset from Sparkling Lake is included in the package and can be accessed in 'R'. In addition, the 'R' code used to generate metabolism estimates and figures for Sparkling Lake is available within the package as a demo (access using `demo(package='LakeMetabolizer')` 'R' function call).

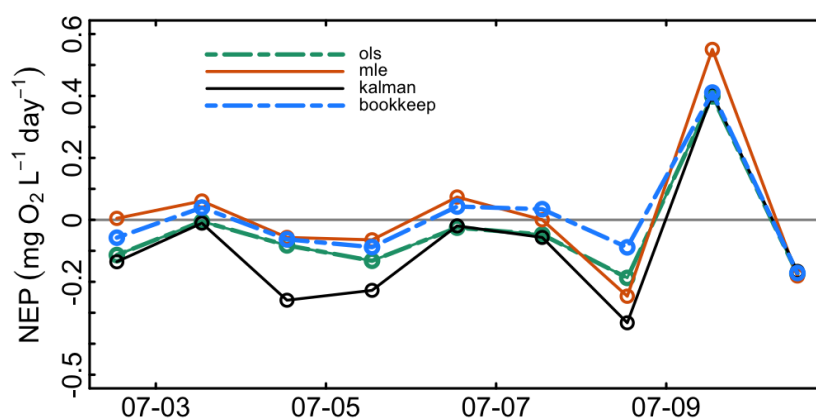


Figure 1. Comparison of four different metabolism models (OLS = ordinary least squares; MLE = maximum likelihood; Kamlan = Kalman filter; bookkeep = Bookkeeping) for estimating Net Ecosystem Production (NEP).

As all methods can be run using the same input files, Lake Metabolizer allows comparisons between methods. For example, in Figure 1 we can see that each of the methods can return different estimates, where even the sign of NEP can vary between the different methods. Furthermore, using the example dataset provided we see that the different gas transfer coefficient models can return very different estimates of k600 (Fig. 2); see Dugan et al. (*in*

press), with averages ranging from a minimum of approximately 0.5 m day^{-1} to a maximum of approximately 3 m day^{-1} . Lake Metabolizer provides a means of estimating lake metabolism and related terms using a consistent method, thereby facilitating global comparisons of high frequency data from lake buoys.

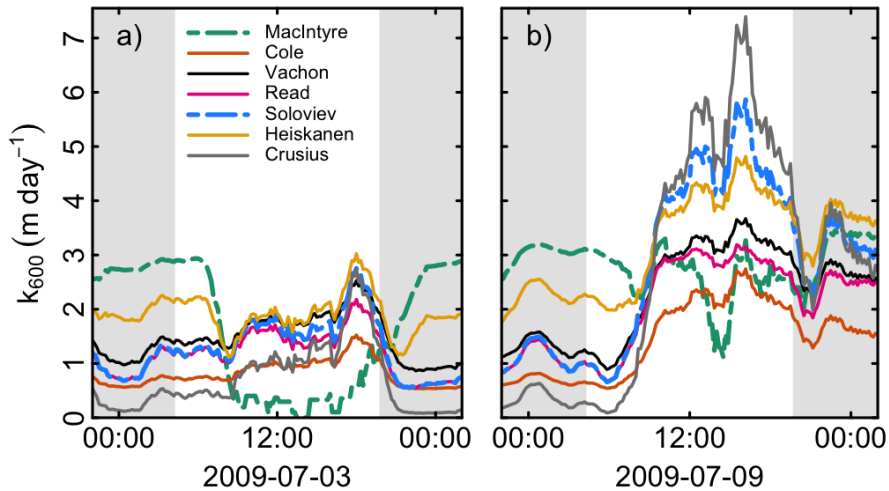


Figure 2. Comparison of the seven different gas transfer coefficient models included in the Lake Metabolizer package. Grey regions illustrate night-time, which can also be estimated by the Lake Metabolizer package (i.e. sun rise and sun set times).

Background

The package requires some experience of using 'R'. However, the user manual (see link below) does provide a number of examples for using the different functions.

Having these methods in an 'R' environment allows them to be calculated with relative ease. However, while the tool can be used without having any prior knowledge of lake metabolism, interpretation of the results does require some understanding of the principles behind aquatic metabolism.

Type of data and requirements

At a minimum, high frequency DO (at least hourly observations), irradiance (typically photosynthetically active radiation [PAR]), wind speed, and water temperature at the depth of the DO sensor are required for estimating metabolism (using the free-water oxygen technique - see Staehr et al. 2010). However, to use all of the available gas transfer coefficient models, the user will need additional data. The data required for each gas transfer coefficient model are shown in Table 2.

	k.cole (Cole and Caraco 1998)	k.crusius (Crusius and Wanninkhof 2003)	k.vachon (Vachon and Prairie 2013)	k.heiskanen (Heiskanen et al. 2014)	k.macIntyre (MacIntyre et al. 2010)	k.read (Read et al. 2012)	k.read.soloviev (Read et al. 2012; Soloviev et al. 2007)
Wind speed	✓	✓	✓	✓	✓	✓	✓
Air temperature					✓	✓	✓
Relative humidity				✓	✓	✓	✓
Short-wave radiation				✓	✓	✓	✓
Water temperature profile				✓	✓	✓	✓
Wind height					✓	✓	✓
Atmospheric pressure				✓	✓	✓	✓
Net Long-wave radiation				✓	✓	✓	✓
Latitude						✓	✓
Area			✓			✓	✓
Wind height	✓	✓	✓	✓	✓	✓	✓

Table 2. Data required for each gas transfer coefficient model included in Lake Metabolizer. References for the most relevant publication associated with each gas transfer coefficient is provided in brackets. k.read.soloviev is a new gas transfer coefficient model used by Dugan et al (*in press*) where the model of Read et al. 2012 is modified to include the influence of breaking waves, from Soloviev et al. 2007, on the gas transfer coefficient.

Formatting of the input files is detailed in the user manual. Note that the formatting of the input files is important, as the functions used by the package to load the data assumes that the user has followed the examples provided in the user manual. For example, DO data must be formatted as a tab-delimited text file as:

```
dateTime      doobs_0.5
2009-07-23 00:00      13.550
2009-07-23 00:01      13.493
2009-07-23 00:02      13.458
```

This file format is the same as that required by Lake Heat Flux Analyzer (Woolway et al. 2015, see Jones 2016) and Lake Analyzer (Read et al. 2011), thus allowing them to be used by a number of programs to provide specific details of the lake.

Basic procedures

The procedure to follow is detailed in the user manual of the 'R' package for Lake Metabolizer (see link below), and differs depending on the chosen model. Only a brief synopsis is given here:

1. Collect and clean high frequency data (see de Eyto and Pierson 2016).
2. Determine which types of data and metadata are available (e.g. wind speed, air temperature, short-wave radiation, lake latitude, lake area, etc.).

3. Compare list of data available to determine which model(s) are available for use (see user manual).
4. Choose gas transfer coefficient and metabolism methods for estimating metabolism and related variables.
5. Load necessary time series and metadata in 'R' using the helper functions provided (see user manual)
6. Run metabolism model using the helper function for that particular model.

Pitfalls and tips

- The package estimates metabolism with the most widely used modelling techniques. However, there are a number of areas where implementation differs and it is unclear if there is community consensus that point to a single model strategy.
- As defined, negative Gross Primary Production (GPP) and positive Respiration (R) are ecologically impossible. Unfortunately, unconstrained metabolism estimates using free-water oxygen can return negative GPP and positive R. There are generally two strategies for handling such model output, (i) the model can be run unconstrained and the impossible estimates can be removed, and (ii) the model can be written to constrain the parameters and force the estimation of positive GPP and negative R.
- All methods, except for the bookkeeping method, estimate GPP using a linear light dependency of primary production. Although this approach may be adequate for many lakes, there is evidence that light saturation or even inhibition may more accurately model metabolism in some lakes. Integration of non-linear primary production relationships with light may be included in later versions of the package.
- Currently, LakeMetabolizer supports estimates of metabolism from a surface DO sensor at a single location. Future versions of the package may include calculation of whole-lake metabolism across multiple DO sensors (Obrador et al. 2014, see Obrador et al. 2016).

Further reading

Key References:

The reference for the paper describing the code and its uses is:

Winslow, L. A., Zwart, J. A., Batt, R. D., Dugan, H. A., Woolway, R. I., Corman, J. R., Hanson, P. C., Read, J. S. LakeMetabolizer: An R package for estimating lake metabolism from free-water oxygen using diverse statistical models. *Inland Waters* (in press)

Lake Metabolizer Manual:

<http://cran.r-project.org/web/packages/LakeMetabolizer/LakeMetabolizer.pdf>

Other references:

For an example of lake metabolism being calculated and used see:

Dugan, H.A., Woolway, R.I., Santoso, A.B., Corman, J.R., Jaimes, A., Nodine, E.R., Patil, V.P., Zwart, J.A., Brentrup, J.A., Heatherington, A.L., Oliver, S.K., Read, J.S., Winters, K.M., Hanson, P.C., Read, E.K., Winslow, L.A., Weathers, K.C. Consequences of gas flux model choice on the interpretation of metabolic balance across 15 lakes. *Inland Waters* (in press)

Other useful references for lake metabolism and the gas transfer coefficients are:

Batt, R.D., Carpenter, S.R. 2012. Free-water lake metabolism: addressing noisy time series with a Kalman filter. *Limnology and Oceanography Methods* 10: 20-30.

Cole, J.J., Caraco, N.F. 1998. Atmospheric exchange of carbon dioxide in a low-wind oligotrophic lake measured by the addition of SF₆. *Limnology and Oceanography* 43, 647-656.

Crusius, J., Wanninkhof, R. 2003. Gas transfer velocities measured at low wind speed over a lake. *Limnology and Oceanography* 48: 1010-1017.

de Eyto, E., Pierson, D. 2016. Data handling: cleaning and quality control. In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 1). Technical report. NETLAKE COST Action ES1201. pp. 2-6.

<http://eprints.dkit.ie/id/eprint/532>.

Heiskanen, J.J., Mammarella I., Haapanala, S., Pumpanen, J., Vesala, T., MacIntyre, S., Ojala, A. 2014. Effects of cooling and internal wave motions on gas transfer coefficients in a boreal lake. *Tellus B: Chemical and Physical Meteorology* 66: 1-16.

Holtgrieve, G.W., Schindler, D.E., Branch, T.A., A'Mar, T. 2010. Simultaneous quantification of aquatic ecosystem metabolism and respiration using a Bayesian statistical model of oxygen dynamics. *Limnology and Oceanography* 55: 1047-1063.

Honti, M. 2016. Bayesian calibration of mechanistic models of lake metabolism. In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 8). Technical report. NETLAKE COST Action ES1201. pp. 40-46.

<http://eprints.dkit.ie/id/eprint/539>.

Jones, I.D. 2016. Lake Heat Flux Analyzer (LHFA). In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 3).

Technical report. NETLAKE COST Action ES1201. pp. 11-15. <http://eprints.dkit.ie/id/eprint/534>.

MacIntyre, S., Jonsson, A., Jansson, M., Aberg, J., Turney, D.E., Miller, S.D. 2010. Buoyancy flux, turbulence, and the gas transfer coefficient in a stratified lake. *Geophysical Research Letters* 37 doi: 10.1029/2010GL044164.

Odum, H. 1956. Primary production in flowing waters. *Limnology and Oceanography* 1: 102-117.

Obrador, B., Staehr, P.A., Christensen, J. 2014. Vertical patterns of metabolism in three contrasting stratified lakes. *Limnology and Oceanography* 59: 1228-1240.

Obrador, B., Christensen, J., Staehr, P.A. 2016. Determination of whole-column metabolism from profiling data. In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 9). Technical report. NETLAKE COST Action ES1201. pp. 47-51. <http://eprints.dkit.ie/id/eprint/540>.

Read J.S., Hamilton D.P., Jones I.D., Muraoka K., Kroiss R., Wu C.H., Gaiser E. 2011. Derivation of lake mixing and stratification indices from high-resolution lake buoy data. *Environmental*

Modelling and Software 26: 1325–1336.

Read, J.S., Hamilton, D.P., Desai, A.R., Rose, K.C., MacIntyre, S., Lenters, J.D., Smyth, R.L., Hanson, P.C., Cole, J.J., Staehr, P.A., Rusak, J., Pierson, D., Brookes, J., Laas, A., Wu, C. 2012. Lake-size dependency of wind-shear and convection as controls on gas exchange. *Geophysical Research Letters* 39: L09405.

Soloviev A., Donelan, M., Graber, H., Haus, B., Schlüssel, P. 2007. An approach to estimation of near-surface turbulence and CO₂ transfer velocity from remote sensing data. *Journal of Marine Systems* 66: 182–194.

Staehr, P.A., Bade, D., van de Bogert, M.C., Koch, G.R., Williamson, C., Hanson, P., Cole, J.J., Kratz, T. 2010. Lake metabolism and the diel oxygen technique: State of the science. *Limnology and Oceanography Methods* 8: 628–644

Vachon, D., Prairie, Y. 2013. The ecosystem size and shape dependence of gas transfer velocity versus wind speed relationship in lakes. *Canadian Journal of Fisheries and Aquatic Sciences* 70: 1757-1764.

Woolway, R.I., Jones, I.D., Hamilton, D.P., Maberly, S.C., Muraoka, K., Read, J.S., Smyth, R.L., Winslow, L.A. 2015. Automated calculation of surface energy fluxes with high-frequency lake buoy data. *Environmental Modelling and Software* 70: 191–198.

Code

The code for LakeMetabolizer has been released under the GPL version 2 open-source license. It is available both as an ‘R’ package on CRAN, using the command `install.packages('LakeMetabolizer')` and under the version management repository used for development (<https://github.com/GLEON/LakeMetabolizer>).

Contact details

R. Iestyn Woolway. University of Reading, UK.
riwoolway@gmail.com

Luke A. Winslow, United States Geological Survey, WI, USA.

Jake A. Zwart, University of Notre Dame, IN, USA.

Suggested citation

Woolway, R.I. 2016. Lake Metabolizer. In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 4). Technical report. NETLAKE COST Action ES1201. pp. 16-22. <http://eprints.dkit.ie/id/eprint/535>.

Acknowledgement

This factsheet is based upon work from the NETLAKE COST Action ES1201, supported by COST (European Cooperation in Science and Technology).

NETLAKE toolbox for the analysis of high-frequency data from lakes



Factsheet #5

High frequency data treatment and visualization with ECDSOFT and OnLineMonitor

Dario Omanović and Ivanka Pižeta

Objective

After repeatedly collecting a series of numerical x,y pairs of data, and before further treatment, it might be useful to visualize them, to check with “*an expert eye*” whether the data are in the expected frames and/or to decide about subsequent steps such as smoothing, background subtraction, and determination of signal heights and positions. 2D, or even better, 3D visualization with rotation feasibility could reveal useful information.

Getting rid of high-frequency noise that is superimposed on a signal of interest helps to recognize the useful parts of a signal. However, it is important to visually check that the signals are not distorted when smoothing, because peak features will be extracted from these smoothed series, and the so-called secondary curves will be constructed, and these will be checked against expected models (e.g. linear, exponential, etc.). So, visualization of smoothing effects while choosing algorithms and their parameters is necessary, and is welcome and appreciated in any step of data treatment.

The basic software tool is ElectroChemistry Data SOFTWARE (ECDSOFT) (Omanović and Branica 1998) designed for treatment of data obtained by the electrochemical method, but it is capable of accepting any set of signals that matches the required format. The software itself has been continuously upgraded and is open for further improvements on request. The subsequently developed software, sharing the main data treatment features of ECDSOFT, intended to automatically analyse sets of such data and present them in a near-real time domain is OnLineMonitor.

Specific application

ECDSOFT is designed specifically for the treatment of voltammetric signals, i.e. current vs. voltage peak-shaped curves (Figure 1). Such signals usually consist of only a few well-defined peaks (two peaks are shown in Figures 1a and 1c), but there could be a high-frequency noise superimposed on the signal and there could be a background level that is not related to the analyte expressed as a peak, and which preferably should be removed (e.g. subtracted). Still, any such set of curves originating from different methods (e.g. lake temperature vs. depth and time), could benefit from the software, as it is handy for visualization, smoothing, background

subtraction, and peak height and position determination, and can perform all this automatically once the parameters have been determined.

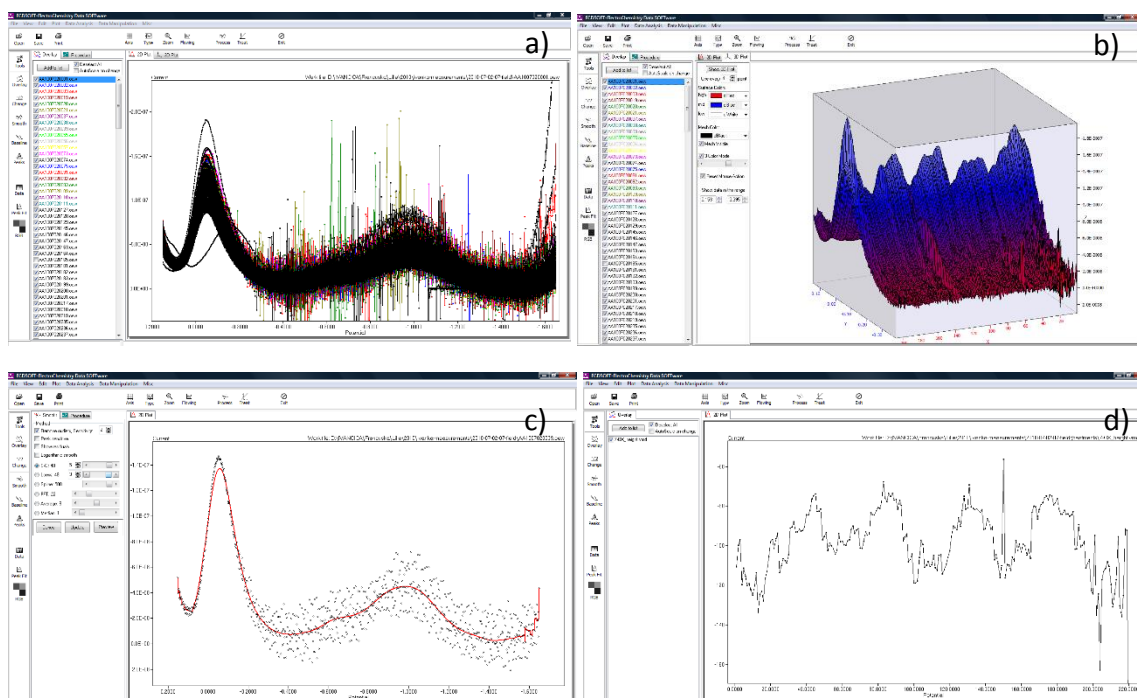


Figure 1. a) an example of raw data from voltammetric automatic measurements of dissolved oxygen in a small lake for several days at 1 hour intervals; b) 3D plot of the data from a) (only first peak on the left); c) S-G smoothing of one curve from a) – distortion of peak height is shown in red (with S-G: 40) (indicating that another parameter of S-G smoothing should be chosen, i.e. S-G: 18); d) result of automatic oxygen peak height determination for all series of signals from b) (showing day – night variations of oxygen concentration for five days).

OnLineMonitor software is designed to track specific folders in a defined time period (e.g. every 5 mins) for new files (e.g. voltammograms, spectra, etc.) and analyse them using predefined parameters for specific peak shaped signals. Up to 20 different parameters can be monitored. The software also allows simple correlation analysis of data (Figure 2).

Background

Both software packages are free of charge and could be downloaded at <https://sites.google.com/site/daromasoft/> (ECDSoft) or could be obtained on request (OnLineMonitor; the program is still in the early development stage and thus is not yet available for download over the web-page). Opening and presentation of files (of predefined format, see below) is as simple as any other document, while the use of data treatment routines needs knowledge on general signal characteristics. The user is responsible for organizing the data in repeatable sequences if a 3D presentation is required. Otherwise it can be any sequence of high-frequency data (e.g. time series of temperature, wind, pressure...).

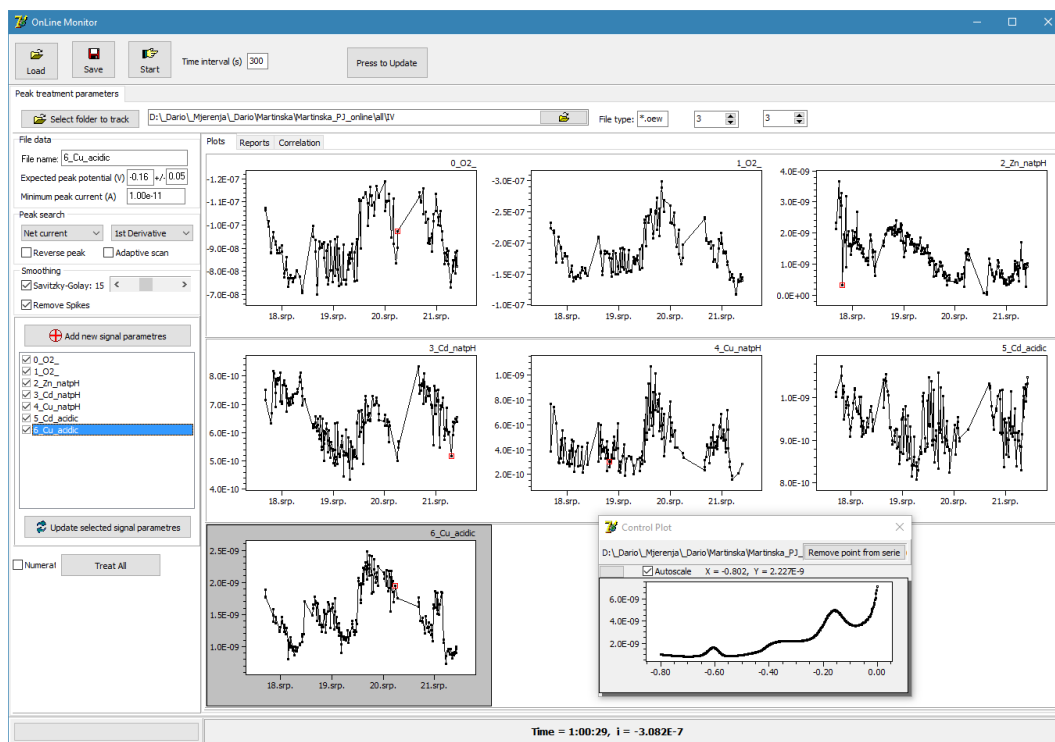


Figure 2. OnLineMonitor program showing automatic analysis and presentation of signal data (O₂, Zn, Cd, Cu) obtained during ~3.5 days of automated voltammetric measurements in the Krka River estuary, Croatia.

Type of data and requirements

Any standard laptop would do to install the software and to use it. The data should be stored in the format "x TAB y" or "x SPACE y" under its name in .vmd files (e.g., name.vmd) (Figure 3). Up to 500 files (XY datasets) could be loaded at the same time and graphically presented, as shown in Figure 1a. Any XY data-pair (e.g. selected in the Excel table) could be plotted in ECDSOFT by using "copy-paste" formalism. In this case, for further processing the dataset should be saved. Also, vice versa, all datasets that have been entered can be copied and transferred to another program (e.g., Excel) by copying all data to the clipboard.

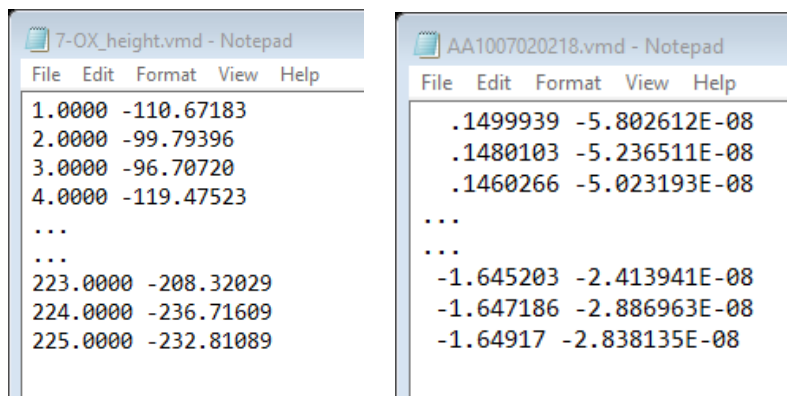


Figure 3. Examples of format files ready to be used by ECDSOFT.

Basic procedures

GENERAL

Once data are inserted they are displayed, as seen in Figure 1a. The name of each file is listed on the left side of the screen. Each file can be excluded or included by clicking the checkmark to the left of the file name. It is also possible to zoom-in on any part of the dataset. A 3D presentation is available by selecting the **3D Plot** box (in Figure 1b the 3D presentation of zoomed signals from Figure 1a is shown). The figure can easily be rotated in any direction to enable the best view by left-clicking on the plot and dragging it with the mouse.

SMOOTHING

For smoothing, select *Smooth* on the left toolbar. Choose the method of smoothing from *Savitzky-Golay (S-G)*, *Loess*, *Spline*, *FFT*, *Average* or *Median*. The choice of the method would depend on the character of the signal and noise spectrum, while its effects are automatically refreshed and easily visually checked (the user is advised to get informed about the basic principles of the methods). Each method has a scrollbar to adjust for additional parameters. In Figure 1c, the S-G method was chosen and the scrollbar was adjusted to 40. The red line is a *Preview* of the results of smoothing. After pressing the *Update* button, a new smoothed curve is ready for further treatment. It can be saved under a different name. The smoothing could be repeated with different methods by pressing *Resume* on the *Plot/Resume* menu (F5).

PEAK FEATURES

For peak height and position determination of a whole set of curves, select *Data Management/Automatic processing* and adjust the parameters, selecting from many options. Generally, it is better to consider *peak feature*, because except for peak height, the signal can also be quantified by a peak area and/or by determination of a peak 1st, 2nd or 4th derivative. Selected peak features are automatically determined and saved in a file that can be uploaded and viewed in the same program (Figure 1d) or in another program (e.g. Excel). The OnLineMonitor program (Figure 2) provides similar final analysis results as those presented in Figure 1d, but for up to 20 measuring parameters. It is advised to first decide about parameters of signal treatment in ECDSOFT, and then apply them in OnLineMonitor, where more parameters could be followed in the same time.

Pitfalls and tips

ECDSOFT and OnLineMonitor, primarily designed for voltammetric data sets, have many specific features for this type of signal, but could analyse “peak-shaped” signals of any kind. The software is full of useful details and is very handy once the user becomes familiar with it. An intuitive trial-and-error approach is advised when first approaching this software. ECDSOFT is more intuitive and intended for “In-depth” analysis and visualisation, while OnLineMonitor is more appropriate for tracking changes of selected parameters in (e.g. time) sequence.

Further reading

For examples on the use and usefulness of this software see:

Omanović, D., Branica, M. 1998. Automation of voltammetric measurements by polarographic analyser PAR 384B. *Croatica Chemica Acta* 71: 421-433.

Superville, P.J., Louis, Y., Billon, G., Prygiel, J., Omanović, D., Pižeta, I. 2011. An adaptable automatic trace metal monitoring system for on line measuring in natural waters. *Talanta* 87: 85-92.

Superville, P.J., Pižeta, I., Omanović, D., Billon, G. 2013. Identification and on-line monitoring of reduced sulphur species (RSS) by voltammetry in oxic waters. *Talanta* 112: 55-62.

Code

The ECDSOFT program can be downloaded from the following link (source code is not available):

<https://sites.google.com/site/daromasoft/home/ecdssoft>

Contact details

Ivanka Pižeta. Ruđer Bošković Institute, Zagreb, Croatia.

pizeta@irb.hr

Dario Omanović. Ruđer Bošković Institute (Zagreb, Croatia)

omanovic@irb.hr

Suggested citation

Omanović, D. and Pižeta, I. 2016. High Frequency data treatment and visualization with ECDSOFT and OnLineMonitor. In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 5). Technical report. NETLAKE COST Action ES1201. pp. 23-27. <http://eprints.dkit.ie/id/eprint/536>.

Acknowledgement

This factsheet is based upon work from the NETLAKE COST Action ES1201, supported by COST (European Cooperation in Science and Technology).

NETLAKE toolbox for the analysis of high-frequency data from lakes



Factsheet #6

Lake stratification and ice phenology: Modelling with MyLake

Raoul-Marie Couture and Koji Tominaga

Objective

Lake modelling is a thriving field of research, and many modelling tools are now available to the researchers (see Janssen et al. 2015). Prospective users of a model will make a choice based, amongst others, on the desired level of complexity and their preferred scientific programming environment. Here we describe the MyLake lake model, a simple one-dimensional (1D) daily time-step model that can be used to simulate seasonal changes in ice coverage in lakes. This model is aimed at researchers who prefer to use Matlab/Octave language for scientific computing applications. This factsheet describes briefly how to set-up the MyLake lake model in order to simulate thermal stratification and ice phenology in a lake. Such modelling efforts will allow studying the effect of various forcing factors on the lake's thermal balance and predict the evolution of the ice coverage over time. Because the presence of ice influences exchanges between the atmosphere and the surface of the lake as well as the penetration of light in the water column, ice modelling is often a first step towards simulating other seasonal phenomenon, such as algal blooms and bottom-water anoxia.

The lake model MyLake v1.2 was written by Tuomo Saloranta and Tom Andersen (Saloranta and Andersen 2007). It has been used in several studies since then, focussing on boreal lakes experiencing seasonal ice cover. New functionalities and modules have been added during the course of these studies regarding biogeochemistry (e.g., Couture et al. 2015): here we focus on v1.2 which is described in the original publication. MyLake uses a stacked layer geometry consisting of mixed horizontal layers. Its hydrodynamic module (1) calculates day- and night-time surface heat fluxes and heat sources, turbulent kinetic energy from wind (in the absence of an ice cover), and heat-fluxes between water and sediment, (2) performs convective mixing, and (3) applies a routine to calculate the vertical turbulent diffusion coefficient and the settling of solid components. If the density of the water coming into the lake (i.e., inflow) is less than or equal to the density of the surface layer, the inflow is mixed with the surface layer. Otherwise, the inflow is added on top of the first layer which is heavier than the inflow, thus displacing an equal amount of outflowing water and conserving lake volume. The model then calculates congelation and ice growth.

Specific application

The MyLake lake model is best suited for applications having the following characteristics: (1) the geometry of the lake basin lends itself to the “1D assumption”, which neglects lateral heterogeneity, or if you can accept the limitation of working with a single 1D profile (2) you favour short integration time over model complexity, (3) you do not need complex lake physics or ecological modelling (e.g., saline or groundwater intrusions, reservoir management, food-web, etc.), (4) you want access to the source code and would like to modify it yourself if necessary, and finally (5) the lake experiences seasonal ice cover. If you are satisfied with these constraints, the MyLake lake model is useful as it includes only the most significant physical (e.g., heat conservation equations for the temperature distribution in a horizontally homogenous, vertically stratified lake), chemical (e.g. P partitioning) and biological (phytoplankton growth) processes in a balanced way. From the computational point of view, the model is designed to facilitate sensitivity and uncertainty analysis and to simulate a large number of lakes over long periods of time (e.g., decades). Here, we will describe how to setup the MyLake model in a generic boreal lake with a simple bathymetry, launch a 40 year simulation, and visualize the results.

Background

The code is written in Matlab. Basic knowledge on how to retrieve code from the GitHub platform, and on using Matlab and associated computing skills (File I/O, scripting, plotting) is required.

Software requirements: MATLAB version > 2012 with Statistical toolbox, or Octave; a GitHub client (we recommend SourceTree). The model has been tested using Matlab in a Windows environment and using Octave in a Linux environment.

Type of data and requirements

Input files are tab-delimited text files, with one line per lake depth interval (initial condition file) or one line per simulation day (input file). They can be prepared in spreadsheet software and saved to tab-delimited text.

The MyLake initial condition file requires the following:

- Depth levels (in metres, positive from the surface).
- Horizontal areas of each depth layer (m²).
- Initial temperature profiles (°C).
- Initial profiles of biogeochemical variables (set to zero for this exercise).
- Initial value of total ice thickness (m).
- Initial value of snow thickness (m).

The depth levels and horizontal areas are to be calculated from available bathymetric information. This will enable MyLake to calculate lake volume, which is then assumed to remain constant.

The Mylake input file comprises columns of the following daily values:

- Global radiation ($\text{MJ m}^{-2} \text{d}^{-1}$) (optional, can be estimated based on latitude/longitude)
- Cloud cover (0 - 1)
- Air temperature ($^{\circ}\text{C}$) at 2 m
- Relative humidity (%) at 2 m
- Atmospheric pressure (hPa) at station level
- Wind speed (m s^{-1}) at 10 m height above ground
- Precipitation (mm d^{-1})
- Daily inflow ($\text{m}^3 \text{d}^{-1}$)
- Daily inflow concentrations (mg m^{-3}) of suspended matter, total phosphorus, dissolved organic phosphorus, chlorophyll; dissolved organic carbon and other biogeochemical variables (can all be excluded if looking only at temperature and ice formation)

The MyLake v.1.2 parameter file (*lake_para.txt*) comprises two sets of parameters: the lake physical parameters and the lake biological parameters. Note that Mylake v.2, not described here, contains a new set for the biogeochemical processes in the water column and in the sediment column. These parameters can be edited manually via the text files, or written at run-time via either (1) the auto-calibration module or (2) a catchment model running before MyLake. The physical parameters relating to lake thermodynamics and ice formation are shown in Table 1.

Par	Detail	Units
Albedo	Input of fn heatflux_v12.m	
dz	Grid step size	m
Kz_K1	Open water diffusion parameter	
Kz_K1_ice	Under ice diffusion parameter	
Kz_N0	Min. stability frequency	s^{-2}
C_shelter	Wind shelter parameter	
lat	Latitude	decimal degrees
lon	Longitude	decimal degrees
alb_melt_ice	Albedo of melting ice, Input of fn heatflux_v12.m	
alb_melt_snow	Albedo of melting snow, Input of fn heatflux_v12.m	
lambda_i	PAR light attenuation coef. of ice	m^{-1}
lambda_s	PAR light attenuation coef. of snow	m^{-1}
F_sed_sld	Volume fraction of solid in sediment	1-phi
I_scV	Scaling factor for inflow volume (multiplicative)	
I_scT	Scaling coef. for inflow temperature (additive)	

Table 1. Name of the user-defined parameters affecting ice formation taken from the parameter file, along with short description and units.

In addition, additional parameters are found in the script file and not read from the parameter files (although they can be changed manually), as shown in Table 2.

Par	detail	default value	Units
dt	Time step	1	
e_par	Average energy of PAR photons	240800	J mol ⁻¹
Kz_b1	Diffusion param. exponents	0.43	
Kz_b1_ice	Diffusion param. exponents	0.43	
rho_fw	Density of freshwater	1000	kg m ⁻³
rho_ice	Density of ice/snow-ice	910	kg m ⁻³
rho_new_snow	Density of new snow	250	kg m ⁻³
max_rho_snow	Maximum snow density	450	kg m ⁻³
L_ice	Latent heat of freezing	333500	J kg ⁻¹
K_ice	Ice heat conduction	2.1	W m ⁻¹ K ⁻¹
C1	Snow compaction coef. 1	7.0	
C2	Snow compaction coef. 2	21.0	
Tf	Water freezing point temperature	0	°C
K_sed	Thermal diffusivity of sed	0.035	m ² d ⁻¹
rho_sed	Bulk density of inorg. solids in sed.	2500	kg m ⁻³
rho_org	Bulk density of org. solids in sed.	1000	kg m ⁻³
cp_sed	Heat capacity of sediment	1000	J kg ⁻¹ K ⁻¹
Ksw	Sediment-p.w. mass transfer coef.	1×10 ⁻⁶	
Frail2Ice_tresh	Threshold where frazil turns into ice	0.03	m
Cw	Volumetric heat capacity of water	4.18×10 ⁶	J K ⁻¹ m ⁻³
G	Gravity acceleration	9.81	m s ⁻²

Table 2. Name of the model default parameters affecting ice formation fixed in the main code along with short description and units.

Basic procedures

The procedure to follow is detailed in the user manual available on the web-page (link below). Only a brief synopsis is given here:

1. Prepare your input and initial condition files.
2. Obtain the source code of MyLake v1.2 at the GitHub repository (see below).
3. Copy the folder v12 and all its contents to your computer.
4. Start MATLAB and modify all the path names for model code, observations, and forcing files in the example model application script (TSA_modelIVAN_v12.m) so that they point to the right folders (search for string "H:\\" in order to find these lines). Use relative file path names if preferred.
5. Run the application script. Some information, such as ice-on/off dates, is displayed on command window while the model is running. A default set of figures are plotted when the model run is finished, as shown on Figure 1 (upper panel).

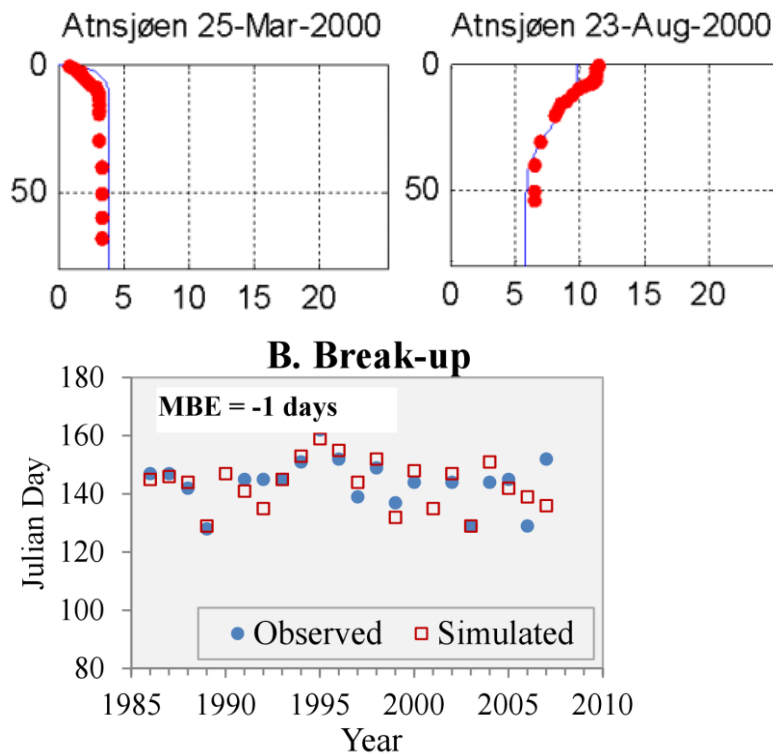


Figure 1. Upper panel: Vertical temperature profile measured (line) and simulated by the MyLake lake model (circles) in winter and spring in Lake Atnsjøen (Norway). Lower panel: Ice break-up dates observed (solid symbols) and predicted (open symbols) in the same lake based on observed meteorological forcing data; also shown are the mean bias error (MBE). Reproduced from Gebre et al. (2014).

Pitfalls and tips

- Input data that contain data gaps will be linearly interpolated by the model.
- The model is based on a daily time step (although surface heat balance is solved separately during day- and night time), therefore there might be numerical error if a process is much faster than a day. In particular, the surface layer can undergo significant temperature variations during a day. To alleviate this problem and avoid numerical instability in the temperature and ice simulations, the model grid size (dz , see Table 1) should be larger than the thickness of the surface layer.
- Model outputs are very sensitive to model daily inputs, such as inflow volumes, air temperature, and wind speed, especially so for smaller lakes.
- Changing default parameterisation for lake physics is rarely necessary outside of heat diffusion, wind mixing (sheltering) and ice albedo parameters.

Further reading

Key Reference:

Saloranta, T., Andersen, T. 2007. MyLake—A multi-year lake simulation model code suitable for uncertainty and sensitivity analysis simulations. *Ecological Modelling* 207: 45-60.

MyLake User Manual and code descriptions:

Saloranta, T., Andersen, T. 2004. MyLake (v.1.1) Technical model documentation and user's guide for version 1.1. http://brage.bibsys.no/xmlui/bitstream/handle/11250/212445/1/4838_200dpi.pdf

For a Lake ice module physical description see supplement of Gebre et al (2014). <http://www.the-cryosphere.net/8/1589/2014/tc-8-1589-2014-supplement.pdf>

For the model's wiki pages on GitHub, see:

https://github.com/biogeochemistry/MyLake_public

Other useful references and recent applications of the model:

Gebre, S., T. Boissy, Alfredsen, K. 2014. Sensitivity of lake ice regimes to climate change in the nordic region. *The Cryosphere* 7: 1589-1605.

Janssen, A.G., Arhonditsis, G.B., Beusen, A., Bolding, K., Bruce, L., Bruggeman, J., Couture, R.-M., Downing, A.S., Alex Elliott, J., Frassl, M.A., Gal, G., Gerla, D.J., Hipsey, M.R., Hu, F., Ives, S.C., Janse, J.H., Jeppesen, E., Jöhnk, K.D., Kneis, D., Kong, X., Kuiper, J.J., Lehmann, M.K., Lemmen, C., Özkundakci, D., Petzoldt, T., Rinke, K., Robson, B.J., Sachse, R., Schep, S.A., Schmid, M., Scholten, H., Teurlincx, S., Trolle, D., Troost, T.A., Van Dam, A.A., Van Gerven, L.P.A., Weijerman, M., Wells, S.A., Mooij, W.M. 2015. Exploring, exploiting and evolving diversity of aquatic ecosystem models: a community perspective. *Aquatic Ecology* 4: 1-36.

Couture, R.M., Tominaga, K., Starrfelt, J., Moe, S.J., Kaste, O., Wright, R.F. 2014. Modelling phosphorus loading and algal blooms in a Nordic agricultural catchment-lake system under changing land-use and climate. *Environmental Science: Processes and Impacts* 16: 1588-1599.

Couture, R.M., DeWit, H., Tominaga, K., Kiuru, P., Markelov, I. 2015. Oxygen dynamics in a boreal lake responds to long-term changes in climate, ice phenology and DOC inputs. *Journal of Geophysical Research: Biogeosciences*. 120: 2441-2456.

Romarheim, A.T., Tominaga, K., Riise, G., Andersen, T. 2015. Natural stochasticity vs. management effort: use of year-to-year variance for disentangling significance of two mutually confounding factors affecting water quality of a Norwegian cold dimictic lake. *Hydrology and Earth System Sciences* 19: 2649-2662.

Holmberg, M., Futter, M., Kotamäki, N., Fronzek, S., Forsius, M., Kiuru, P., Pirttioja, N., Rasmus, K., Starr, M., Vuorenmaa, J. 2014. Effects of changing climate on the hydrology of a boreal catchment and lake DOC - probabilistic assessment of a dynamic model chain. *Boreal Environment Research Suppl. A*: 66-82.

Dibike, Y., Prowse, T., Bonsal, B., Rham, L.D., Saloranta, T. 2012. Simulation of North American lake-ice cover characteristics under contemporary and future climate conditions. *International Journal of Climatology* 32: 695-709.

Saloranta, T.M., Forsius, M., Jarvinen, M., Arvola, L. 2009. Impacts of projected climate change on the thermodynamics of a shallow and a deep lake in Finland: model simulations and Bayesian uncertainty analysis. *Hydrology Research* 40: 234-248.

Code

The code for this technique was written in the Matlab language, and is available at: https://github.com/biogeochemistry/MyLake_public

Contact details

Raoul-Marie Couture and Koji Tominaga. Norwegian Institute for Water Research, Norway.
rmc@niva.no (Twitter : @MyLake_model)

Tuomo Saloranta. The Norwegian Water Resources and Energy Directorate, Norway.

Suggested citation

Couture, R.M. and Tominaga, K. 2016. Lake stratification and ice phenology: Modelling with MyLake. In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 6). Technical report. NETLAKE COST Action ES1201. pp. 28-34. <http://eprints.dkit.ie/id/eprint/537>.

Acknowledgement

This factsheet is based upon work from the NETLAKE COST Action ES1201, supported by COST (European Cooperation in Science and Technology).

NETLAKE toolbox for the analysis of high-frequency data from lakes



Factsheet #7

Knowledge Discovery in Databases - Data Mining

Ivanka Pižeta

Objective

Knowledge discovery in databases (KDD) (Fayyad et al. 1996) is the process of identifying valid, novel, potentially useful, and ultimately understandable patterns or models in data. Data mining (DM) is a step in the knowledge discovery process consisting of particular data mining algorithms that find patterns or models in data. Techniques involved in data mining represent a blend of statistics, pattern recognition and machine learning. An online application is presented where datasets are uploaded to apply KDD.

Specific application

As the objective of knowledge discovery is very broad, the outcome of KDD depends on the expert knowledge of the user who collects the data. The user must define and understand the problem, understand, prepare and model the data, evaluate the results and implement them (an example in this factsheet will try to help in understanding the principles of this work). KDD could give suggestions for further collection of data, for modification of an experiment, and for more specific statistical analyses.

Background

The background of the method is algorithms based on Boolean algebra (see e.g.: <http://www.ee.surrey.ac.uk/Projects/Labview/boolalgebra>) representing the Inductive Learning by Logic Minimization (ILLM) system. Basic understanding of data mining principles is required (see introductory chapters in Fayyad et al. 1996).

Type of data and requirements

Data should be organized in a tabular form. Each row is an example/case, and each column is an attribute/parameter. The first row contains names of the attributes. The maximal number of examples in the basic program is 250 with up to 50 attributes (except in <http://dms1.irb.hr>

where more examples are allowed). They could be **mixed, numerical and categorical data**, and it is not necessary that each case (row) has an attribute value (column). The ability to work with missing data is an added advantage of KDD.

Basic procedures

In order to approach the terminology of DM, the basic procedure will be explained through the example about smokers, given below. A parallel example would be the occurrence of lake stratification or algal blooms.

The procedure starts by defining the problem: if someone is interested in the problem of smokers (or the occurrence of lake stratification or algal blooms) and wants to find out their main characteristics and how they are different from non-smokers (or no occurrence of lake stratification or algal blooms), one has to collect data (attributes) of every case (person/lake) in the population of interest, which includes both smokers and non-smokers, their age, sex, education, profession, income and so on (lake characteristics like surface area of the lake, depth, temperature, wind, nutrients, oxygen...). The result of the data collection phase is a data table in which every object (person/lake) corresponds to one row of a table, described by a set of attributes (columns). For unknown attribute values '?' is used instead. In the 'smoker' problem the attribute containing the information if a person is a smoker or non-smoker (in the lake problem if a lake is stratified/non-stratified or an algal bloom is present/absent) presents the **target attribute** (with values "YES" or "NO"). It means that we are interested in models in which we relate the property "smoker"/"stratification or algal bloom" to other attributes of the person/lake. Every data mining task must have one, and only **one target attribute at a time**. All other attributes are **input attributes**, which are used to build the model of the smoker/a stratified lake/algal bloom present.

After we have selected the target attribute, we must select also the **target class**. In our domain, the target attribute has two classes: smokers and non-smokers. We can select any of these classes as the **target (positive) class**. The other class is the non-target or **negative class**. The result of the data mining process is one or more models (rules), which describe some of the most important subgroups of the target (positive) class. Models describe differences between the target and the non-target (negative) class. Input attributes are used in model descriptions. It must be noted that **existence of examples in both target and non-target classes is mandatory** because the object of induction is the search for differences between the classes.

The table with the collected data for N cases will have N+1 rows (the first row contains attribute names) and M+1 columns (M input attributes and one target attribute). The target attribute is marked by an exclamation mark (!) in front of the first character of its name in the first row, and the target positive class is marked by an exclamation mark (!) in front of each positive class value (see Table 1). Exclusion of an attribute from the calculation is accomplished by putting a question mark (?) in front of its name in the first row. In another session, another attribute could be assigned to be a target one. Also, another set of attributes could be included or excluded from the calculation (in lake science, a target attribute could be either occurrence

of stratification or occurrence of an algal bloom). As discussed below, the entire process of so-called model induction depends on the quality and quantity of data put into the table.

NAME	AGE	SEX	EDUCATION	PROFESSION	WEIGHT	INCOME	?SMOKER	!class_SMOKER
Jan	30	M	LOW	worker	27.3	14000	!YES	!1
John	55.5	M	MIDDLE	worker	90	20000	NO	0
Clara	?	F	HIGH	teacher	65.2	1000	NO	0
Mary	18	F	MIDDLE	student	55.1	0	NO	0
Tom	70	M	HIGH	?	60	9000	!YES	!1
Bill	35	M	MIDDLE	prof	33	16000	NO	0
Steve	42.2	M	LOW	driver	27	7500	!YES	!1
Marc	29	M	?	waiter	31	8300	!YES	!1

Table 1. A simple example of a table prepared for data mining ('smokers' example).

Data can be prepared in Excel, and then saved as a .txt file as requested by the software. The data is then uploaded by the program to the server where the calculation will be done and results in the form of models (rules) will be displayed.

It should definitely be noted that the rule obtained, accurately characterises the difference between the examples describing smokers and non-smokers (Table 2). All smokers are men and have an income below 15000. In other words, all non-smokers are either women, or men who earn more than 15000. From the standpoint of quality in the learning set and interpretability of results we can be quite satisfied with the result. We cannot, however, be satisfied with the overall result, as we know that actually there are a large number of women who smoke. For the same reason we can also expect that the predictive quality of the resultant model will be poor.

<p>Induction results:</p> <p>The result is the following model for the positive class of the target attribute class_SMOKER</p> <p>SUBGROUP A</p> <p>true positive rate (sensitivity*) 100.0%</p> <p>true negative rate (specificity**) 100.0%</p> <p>SUBGROUP A has 2 conditions which both must be satisfied:</p> <p>attribute SEX is equal M</p> <p>attribute INCOME is less than 15000.00</p>
--

Table 2. Resulting model for the positive class of the target attribute class_SMOKER.

Note that the model output includes information on its sensitivity and specificity. **Sensitivity** is a relative number representing the number of correctly predicted positive examples in respect to the total number of positive examples in the input data file. High sensitivity is a greatly appreciated property of every good model, especially if high sensitivity can be obtained together with high specificity. **Specificity** is a relative number representing the number of correctly predicted negative examples in respect to the total number of negative examples in the input data file. High specificity is a necessary property of reliable data models. Many

applications require specificity equal or very near to 100%. Only in situations which require general models or models with high sensitivity, can specificity below 80% be tolerated.

By increasing the **generalization level** (a parameter selectable during data upload) the user can try to induce models with better sensitivity, but typically, models induced in this way will have worse specificity. It is always worth trying this possibility because the decrease of the specificity may be less significant than the gain obtained in sensitivity. By decreasing the **generalization parameter** (a parameter selectable during data upload) the user can try to induce models with better specificity, but typically, models induced in this way will have lower sensitivity.

The poor quality of this example model is the result of problems in data collection. In our set of eight collected examples there is not a single woman smoker. Methods of machine learning as a source of information about the world have only the data we give them in the form of examples. In this case there was no theoretical chance to construct a proper model of women smokers. The only solution is to expand the set of examples and repeat the procedure of induction models.

The conclusion is that the quality of the induced model depends entirely on the quality of the input data. This equally applies to the choice and the amount of available examples as well as the selection and quantity of the attributes which describe examples.

Pitfalls and tips

As it is time consuming to add exclamation marks in front of each target attribute value assigned a positive class, it is more convenient to form additional double columns for each attribute that will be assigned a target one (and keep them inactive by “?” in front of their name). In this column, “!1” is put for a positive class, and “0” for negative classes. When this column is assigned to be a target attribute (by “!” in front of its name), the true one is excluded by “?” (Think/try what would happen if not!). If, for example, we want to assign a positive class to an attribute having values smaller than a certain number, then we first apply SORT BY that attribute function in Excel, then create this double column of “!1” and “0”, along with our decision of what a positive class is. In another session (run), it can be easily changed, either by choosing a new target attribute or by shifting the border of positive class in the same target attribute by rearranging “!” and “?” signs.

Further reading

Key References:

An introduction can be found on the same webpage as the program itself:
<http://dms.irb.hr/index.php> and http://dms.irb.hr/tutorial/tut_applic_ref.php

For the fundamentals of data mining see:

Fayyad U., Piatetsky-Shapiro, G., Uthurusammy, R. (Eds.) 1996. Advances in Knowledge Discovery and Data Mining. AAAI Press/ The MIT Press. Massachusetts.

Other useful references:

Pyle, D. 1999. Data Preparation for Data Mining. Morgan Kaufmann. San Francisco.

Hofsheimer, M., Siebes, A.P. 1994. Data Mining: The Search for Knowledge in Databases. Technical Report. Centre for Mathematics and Computer Science. Amsterdam.

Witten, J., Eibe, F. 2000. Data Mining: Practical Machine Learning tools and techniques wit Java implementations. Morgan Kaufmann. San Francisco.

Weiss, S., Indurkkhya, N. 1998. Predictive Data Mining - A practical guide. Morgan Kaufmann. San Francisco.

Berry J.A., Linoff, G.S. 2000. Mastering Data Mining: the art and science of customer relationship management. John Wiley & Sons. New York.

Fürnkranz, J., Gamberger, D., Lavrač, N. 2012. Foundations of Rule Learning. Springer. Heidelberg.

Code

On-line data treatment is available. A tutorial and detailed instructions on how to prepare the data in a table, how to upload it and get the results is available at: <http://dms.irb.hr>.

Contact details

Ivanka Pižeta. Ruđer Bošković Institute, Zagreb, Croatia.

pizeta@irb.hr

Also, find contact details of the authors of the program on the web page.

Suggested citation

Pižeta, I. 2016. Knowledge Discovery in Databases - Data Mining. In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 7). Technical report. NETLAKE COST Action ES1201. pp. 35-39. <http://eprints.dkit.ie/id/eprint/538>.

Acknowledgement

This factsheet is based upon work from the NETLAKE COST Action ES1201, supported by COST (European Cooperation in Science and Technology).

NETLAKE toolbox for the analysis of high-frequency data from lakes



Factsheet #8

Bayesian calibration of mechanistic models of lake metabolism

Mark Honti

Objective

Resolve the identification issue (different pairs of production and respiration rates produce very similar dissolved oxygen time-series) that arises when complex mechanistic (process-based) metabolic models are calibrated against high-frequency dissolved oxygen (DO) measurements.

Specific application

Changes in DO are primarily related to net ecosystem production (NEP), and as such the time-dynamics of DO shows an aggregated picture on lake metabolism. Physical (e.g. gas exchange with the atmosphere, transport in the water) and chemical (e.g. many redox reactions) processes also contribute to these changes. Thus, it is difficult to disentangle major metabolic processes such as gross primary production (GPP) and ecosystem respiration (R). Several combinations of GPP and R result in very similar DO dynamics resulting in uncertain estimation of photosynthetic and respiration parameters. Instead of seeking for the parameter combination best fitting the data, Bayesian calibration narrows the domain of parameter combinations that yield similarly good fit on the basis of your prior expectations about parameter values. Sampling of posterior parameter distributions yield uncertainty distributions for each parameter.

Background

- Understanding lake metabolism.
- Experience in mechanistic modelling.
- Basic experience in programming.
- Understanding basic statistics (concepts of probabilities, probability distributions).

Type of data and requirements

For the most basic metabolic model, high frequency (30 min or less) records of DO, water temperature (vertical temperature profile), Photosynthetically Active Radiation (PAR), and wind velocity are needed. In shallow lakes, the coefficient of diffuse light attenuation (\sim turbidity) is used.

Extended metabolic models may use a set of additional data:

- Phytoplankton biomass (\sim chlorophyll fluorescence)
- Eddy diffusivity in stratified lakes
- Flow velocity and direction
- Wind direction
- pH, conductivity, alkalinity, CO₂ concentration

Basic procedures

Bayesian parameter inference is an advanced calibration technique, so it is assumed that a mechanistic metabolism model is already up and running.

The first step is to formulate expectations on the parameter values that is to set up the so-called prior distributions. This is usually done by explicitly listing the expected range and expected high probability region (if any) for each parameter based on literature values, expert opinion and domain of meaningfulness (e.g. values below or above a threshold are accepted or not). This information is then compiled into a proper statistical distribution for each parameter. The types and parameterisations (like: mean, standard deviation, etc.) of prior distributions express your subjective willingness to accept a certain value for the parameter in question.

Common prior distribution types are:

- uniform (there is a strictly defined meaningful domain, but there is no preferred choice within that domain),
- beta (the domain of meaningfulness is between 0 and 1 with a peak somewhere in between),
- normal (there is a preference for the mean value, there are no limits, deviations from the preference are accepted in both directions with the same decreasing probability),
- log-normal (negatives are not accepted, a certain deviation above the preference is accepted with higher probability than below it)

Besides these typical examples, any proper unimodal (=having a single peak) statistical distribution will do, if it properly expresses your subjective scientific expectations against the parameter.

The core of the procedure can either be done by modifying your present calibration routine or by plugging your model into a Bayesian calibration framework (e.g. JAGS or BUGS). The first option is discussed below.

Bayesian parameter inference requires the goodness-of-fit measure to be a proper statistical likelihood function. Therefore, if you previously used RMSE, Nash-Sutcliffe or similar informal measures, you have to modify the evaluation module of your script. For high-frequency DO data equidistantly sampled in time the best-suited formal statistical likelihood function is the first-order autoregressive error model. This has 2 parameters: the standard deviation of error innovations (e.g. the change of error from one timestep to the other) and the one-step autocorrelation coefficient. The log-likelihood ($\log L$) of a certain parameter combination is calculated from the residual time-series (E) as follows:

$$\log L = -\frac{n}{2}\log(4\pi) - \frac{1}{2}\sum_i I_i^2$$

where n is the length of the residual series, and I_i are the scaled innovations of the residual series at each timestep except the very first one ($I_i = \frac{E_i - \rho \cdot E_{i-1}}{\sigma}$, where ρ is the autocorrelation between steps and σ is the standard deviation of error innovations). The likelihood is used in combination with the prior probability to evaluate model performance:

$$P_{\text{post}} \propto P_{\text{prior}} \cdot L$$

P_{post} is the posterior probability function that should be used as a new objective function in the calibration procedure. In practice, log posterior probability is used to prevent numerical underflows (when small numbers are accidentally rounded to 0) during computation:

$$\log P_{\text{post}} \propto \log P_{\text{prior}} + \log L$$

Using the autoregressive error model one arrives at the following equation for log posterior probability:

$$\log P_{\text{post}} = -\frac{n}{2}\log(4\pi) - \frac{1}{2}\sum_i I_i^2 + \sum_j \log P_{j,\text{prior}}$$

where j iterates over the model parameters. The log prior probability of individual parameters ($P_{j,\text{prior}}$) couldn't be expanded further in the above equation as it depends on the type of the prior distribution (e.g. normal, lognormal, uniform, etc.) assumed for the given parameter.

The optimal parameter combination will be a compromise between model fit and your subjective expectations. When parameter identification is poor, this compromise usually fits almost as well as unconstrained calibration. It is worth noting that unconstrained calibration does not deliver the objective truth, which may or may not be revealed by unrealistic parameter values.

The uncertainty of posterior parameters can be derived by producing a numerical sample from the posterior parameter distribution using Markov Chain Monte Carlo (MCMC) sampling. The core of this rejection sampling algorithm (Metropolis-Hastings sampler) is:

1. Start with any arbitrary parameter combination. For practical reasons, the combination that belongs to the maximum posterior probability is preferred, if available.
2. Create a new parameter combination from (1) by using a “jump” or “proposal” distribution: Generate a random normal number for each parameter with mean centred at the previous parameter value.
3. Evaluate the log posterior with the new parameter combination. If it is higher than the log posterior of the previous combination, accept the new parameter values and go back to 2. If the new posterior probability is lower, accept the new parameter combination with $P_{post,new} / P_{post,previous}$ probability (or $\exp\{\log P_{post,new} - \log P_{post,old}\}$ when log probabilities were used) and go back to step 2.

Repeating this cycle sufficient times (10^3 to 10^4 iterations), the set of parameter values that have been accepted at step 3 will converge to a proper sample from the posterior parameter distribution. The first portion of the sample is usually discarded because it is distorted by the stabilisation of the sample. The second part of the sample should look like thick noise bands in terms of both posterior probability and parameter values.

The posterior uncertainty of individual parameters can be visualized by extracting the posterior marginal distributions from the sample in plots of density functions of each parameter (Figure 1).

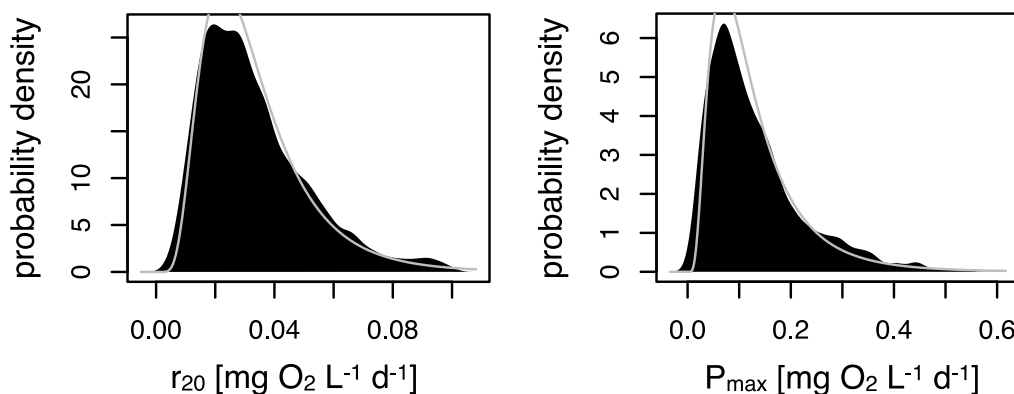


Figure 1. Selected posterior parameter marginals (black shading) from an MCMC sample. r_{20} : community respiration rate at 20°C, P_{max} : maximal rate of gross primary production. The thin grey lines show a fitted lognormal distribution.

Although the principle of MCMC is simple and any implementation following the basic algorithm will work, there are several intricate tricks to make the sampler more efficient. These include a gradual fine-tuning of the proposal distribution to reflect the size and correlation structure of the posterior, thinning the sample to reduce serial correlation, and many others. Therefore, it is generally advisable to use the many existing MCMC implementations of ‘R’ or any other statistical environment.

Pitfalls and tips

Bayesian calibration does not resolve the identification problem of metabolic parameters in an objective way. As priors are subjective, posteriors represent a subjective compromise. Different priors would lead to different posteriors.

Bayesian calibration suffers from the general issue of parameter interpretability just like any other calibration method used for any type of mechanistic model: parameters are optimised during the calibration to compensate for structural deficiencies of the model. Therefore, parameters are biased and hence should be considered as abstract quantities with limited and uncertain physical, chemical or biological meaning. This limits the confidence in analysing calibrated parameter values.

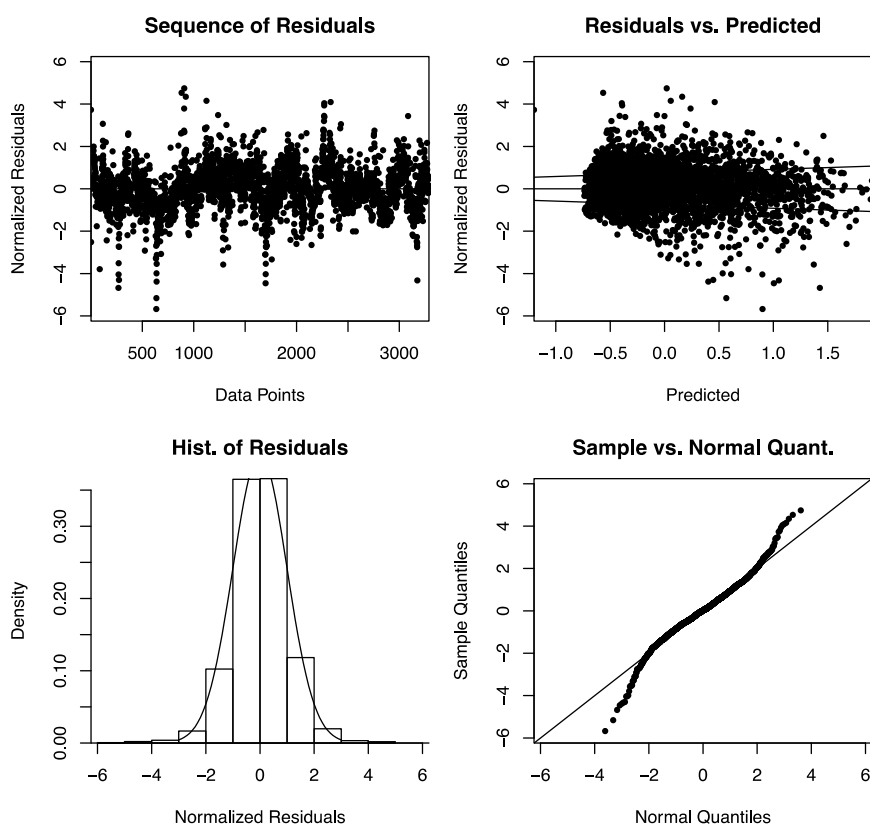


Figure 2. Residual diagnostic plots. Top left: sequence or trace plot; top right: residuals as function of the predicted (DO) value; bottom left: histogram of standardised residuals and a fitted normal distribution; bottom right: Q-Q plot.

Tips

- *Validating the error model.* In formal statistical approaches the likelihood function has to be validated against the posterior residuals to ensure that the statistical assumptions behind the error model are correct or at least not far from reality. This is usually done by testing each assumption on the residuals between observations and the maximum posterior probability solution. In the case of a metabolic model and autoregressive errors, this means testing if residuals have no significant autocorrelation beyond a 1-step lag (*acf* plot), and that innovations are normally distributed with a mean of zero (Q-Q plot). Figure 2 shows a thorough analysis for

independent, normally distributed residuals via plotting their sequence, their dependence on the predicted variable, their density function and a Q-Q plot (layout courtesy of Peter Reichert, EAWAG).

- *Checking MCMC progress.* To assure that MCMC converges successfully, it is common to launch parallel chains and observe whether they converge to the same region. Typical chain lengths are in the range of 2,000–100,000 iteration cycles. It can be shown that the proposal distribution is acceptably tuned if the mean acceptance probability is between 15 and 40 %.
- *Interpreting posteriors.* Posteriors may show two typical relations to priors. If they are very similar to priors, the calibration data did not contain any new and meaningful information about the parameters. This indicates weak identifiability. If posterior distributions are significantly narrower than prior ones, data contained useful information on parameters and hence, priors were suppressed to some degree. Nevertheless, posteriors still remain conditional on priors unless an infinitely long dataset is used for calibration.

Further reading

Key References:

As Bayesian statistics is a fully-fledged discipline within statistics, there are dozens of thick textbooks on the topic. A good example is:

Gelman A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., Rubin, D.B. 2013. *Bayesian Data Analysis*. 3rd edition. CRC Press.

Other useful references:

Application examples related to advanced calibration of metabolic models include

Use of first-order autoregressive error model in calibration:

Van de Bogert, M.C., Carpenter, S.R., Cole, J.J. Pace, M. 2007. Assessing pelagic and benthic metabolism using free water measurements. *Limnology and Oceanography: Methods* 5: 145-155.

Use of first-order autoregressive error model in calibration, parameter uncertainty assessed with bootstrapping:

Solomon, C.T., Bruesewitz, D.A., Richardson, D., Rose, K., Van de Bogert, M., Hanson, P., Kratz, T., Larget, B., Adrian, R., Babin, B.L., Chiu, C.Y., Hamilton, D.P., Gaiser, E., Hendricks, S., Istvánovics, V., Laas, A., O'Donnell, D.M., Pace, M., Ryder, E., Staehr, P.A., Torgersen, T., Vanni, M.J., Weathers, K., Zhu., G. 2013. Ecosystem respiration: Drivers of daily variability and background respiration in lakes around the globe. *Limnology and Oceanography* 58: 849-866.

Use of first-order autoregressive error model in calibration, parameter uncertainty assessed with PEST (informal likelihood procedure with Monte Carlo):

Hanson, P.C., Carpenter, S.R., Kimura, N., Wu, C., Cornelius, S.P., Kratz, T.K. 2008. Evaluation of metabolism models for free-water dissolved oxygen methods in lakes. *Limnology and Oceanography: Methods* 6: 454-465.

Use of Kalman filter (an example of linearised Bayesian updater) with independent, identically distributed error:

Batt, R.D., Carpenter, S.R. 2012. Free-water lake metabolism: Addressing noisy time series with a Kalman filter. *Limnology and Oceanography: Methods* 10: 20-30.

BaMM - Proper Bayesian inference with independent, identically distributed error and simple multi-objective calibration:

Holtgrieve, G.W., Schindler, D.E., Branch, T.A., A'mar, Z. 2010. Simultaneous quantification of aquatic ecosystem metabolism and reaeration using a Bayesian statistical model of oxygen dynamics. *Limnology and Oceanography* 55: 1047–1063.

A complex Bayesian error model demo on DO data from a Swiss river:

Reichert, P., Schuwirth, N. 2012. Linking statistical description of bias to multi-objective model calibration. *Water Resources Research* 48: W09543.

Code

Due to the task-specific requirements there aren't any ready solutions that would meet all limnological needs, but there are solid frameworks which help to carry out the basic steps of Bayesian parameter inference and uncertainty analysis. It is advised to start with the examples attached to these frameworks and develop your own likelihood function, etc.

Rpackages for Bayesian inference can be downloaded from CRAN (by the 'install.packages' command): *mcmc*, *rjags*.

The LakeMetabolizer Rpackage can help you to assemble your metabolic model.

Contact details

Mark Honti. Budapest University of Technology and Economics, Hungary.
honti@vit.bme.hu

Suggested citation

Honti, M. 2016. Bayesian calibration of mechanistic models of lake metabolism. In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 8). Technical report. NETLAKE COST Action ES1201. pp. 40-46.
<http://eprints.dkit.ie/id/eprint/539>.

Acknowledgement

This factsheet is based upon work from the NETLAKE COST Action ES1201, supported by COST (European Cooperation in Science and Technology).

NETLAKE toolbox for the analysis of high-frequency data from lakes



Factsheet #9

Determination of whole-column metabolism from profiling data

Biel Obrador, Jesper Christensen and Peter A. Staehr

Objective

Aquatic metabolism is a fundamental descriptor of ecosystem functioning in lakes. At an ecosystem scale, the metabolism represents the overall rates of production and consumption of organic matter, and is thus informative of the lake carbon balance (Staehr et al. 2012b). Rates of primary production and respiration in lakes are increasingly estimated from diel variations in free-water dissolved oxygen (DO) concentrations (Staehr et al. 2010). While most of the free-water approaches to lake metabolism rely on measurements from a single sonde placed in the epilimnion, increasingly common automated profiling systems allow the determination of metabolic rates along the whole water column (Obrador et al. 2014; Staehr et al. 2012a).

This technique allows determination of metabolic rates, gross primary production (GPP), ecosystem respiration (ER) and net ecosystem production (NEP) for different depth layers along the water column as well as areal, depth-integrated, rates (i.e. per unit area).

Specific application

We used this technique in Obrador et al. (2014) to quantify the relative contribution of the different depth layers to the total metabolism of the water column, and to assess the importance of mixing regime and light availability on the vertical patterns of metabolism in three stratified temperate lakes.

Background

This methodology requires previous knowledge on lake metabolism and on the basic procedures to determine metabolic rates from high-frequency oxygen data (Staehr et al. 2010, Hanson et al. 2008, see also Woolway 2016). Basic statistical knowledge, programming skills (R, SAS, Matlab), and some modelling experience are also required.

Type of data and requirements

- High-frequency profiling data on:
 - Dissolved oxygen (DO)
 - Temperature (T)
- High-frequency data on:
 - Light attenuation (K_d).
 - Wind speed
 - Incident Photosynthetically Active Radiation (PAR)

Data should be at least hourly frequency. The vertical resolution depends on the aim of the work, but at least one measurement in epilimnion, metalimnion and hypolimnion are required.

If high-frequency K_d values are not available, a simple light model from the optically active water components can be constructed, or K_d can be estimated from Secchi disk measurements.

Basic procedures

DATA ARRANGEMENT AND INITIAL CALCULATIONS

1. Align all input data with time so that all measurements of DO (DO_z) and T (T_z) at each depth z correspond in time. For slow profiling systems, it is possible to apply a temporal smoothing or to interpolate the data.
2. Calculate PAR for each time step and depth (PAR_z) from incident PAR and K_d values.
3. Determine epilimnion, metalimnion and hypolimnion depths for each time step, using appropriate definitions, such as the bottom of the epilimnion (Z_{mix}) being the shallowest depth at which the density gradient exceeds a certain threshold. These calculations can be easily done with Lake Analyzer (Read et al. 2011).
4. Create a table with DO_z , T_z and PAR_z values, together with epi-, meta- and hypolimnion depths as well as the wind speed at each time step.
5. Calculate physical fluxes for each depth and time step. Calculate **diffusive air-water gas exchange** (D_s , only considered above Z_{mix}) from the gas transfer velocity for oxygen, and the water-atmosphere oxygen gradient using a standard method (for example Crusius and Wanninkhof 2003, but see Bade 2009 and Woolway 2016 for more details). Calculate flux between adjacent depth layers due to **mixed-layer deepening** (D_z) using changes in Z_{mix} . Calculate **vertical diffusive flux** (D_v) using an estimated vertical diffusivity (for example, Hondzo and Stefan 1993).

METABOLIC CALCULATIONS

7. Calculate the rates of change in DO ($\frac{\Delta O_2(z)}{\Delta t}$). These are the rates of change in DO between two consecutive time steps for each depth layer z .
8. Calculate NEP_z for each time step and depth. Obtain high-frequency NEP rates at each depth from the model describing the dynamics in DO

$$\frac{\Delta O_2(z)}{\Delta t} = NEP_z + Dz_z - Dv_z - Ds_z \quad \text{Eq. (1)}$$

9. Estimate daily physiological parameters for each depth layer. Use a light-dependent photosynthesis model combined with a temperature-dependent respiration model (Hanson *et al.* 2008). NEP_z equals photosynthesis (GPP_z) minus respiration (ER_z) at each depth layer z .

$$NEP_z = GPP_z - ER_z \quad \text{Eq. (2)}$$

Using, for example, the Jassby and Platt (1976) light saturating model of photosynthesis and a Q_{10} of 2 for respiration, the equation describing NEP_z from depth-specific light and temperature is:

$$NEP_z = P_{\max,z} \tanh\left(\frac{\alpha_z PAR_z}{P_{\max,z}}\right) - R_{20,z} 1.07^{(T_z - 20)} \quad \text{Eq. (3)}$$

where $P_{\max,z}$, α_z , and $R_{20,z}$ are the maximum photosynthetic rate, the light use efficiency and the respiration rate at 20°C, respectively, at each depth z .

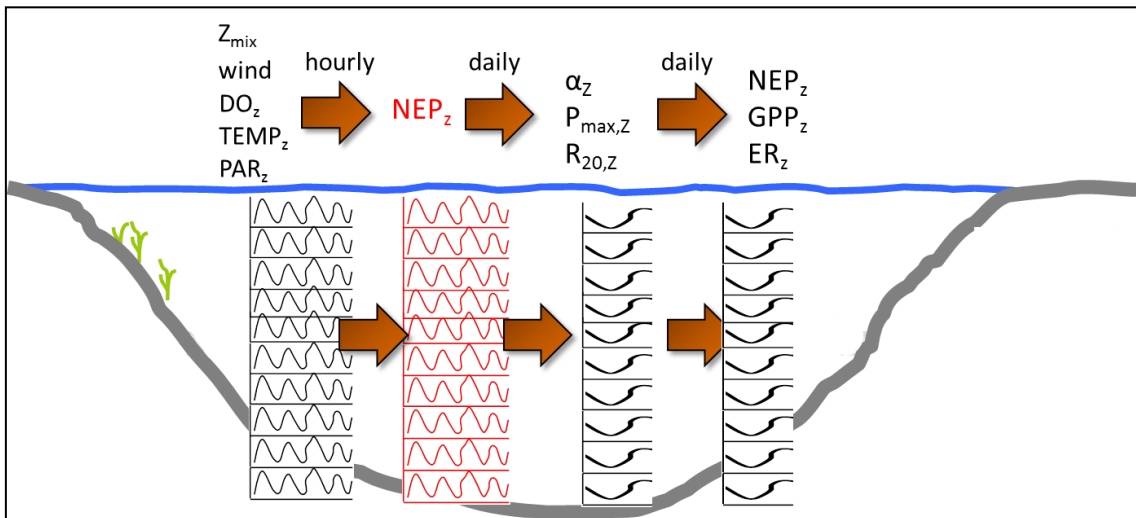


Figure 1. General approach to obtain daily metabolic rates by fitting mechanistic metabolic models to high-frequency profiling data.

10. Apply non-linear fitting between modelled and observed DO data on a fitting window of 24 hours to estimate the parameters P_{\max} , α , and R_{20} at each depth for each day. The parameter estimation can be done by traditional least squares or maximum likelihood fitting methods or by implementing the model in a Bayesian framework (see Honti 2016 and Woolway 2016 in this booklet).
11. Calculate hourly GPP_z , R_z and NEP_z rates from Equations 2 and 3.
12. Integrate the data over 24 hours to obtain the daily depth-specific GPP_z , R_z and NEP_z rates.
13. Integrate the depth-specific rates over the whole water column to obtain the daily areal rates.

Pitfalls and tips

- This methodology is an improved version of the methods used in Staehr et al. (2012a) mainly in the use of a mechanistic modelling approach rather than the traditional book-keeping approach.
- A strict control of units is fundamental, particularly regarding the sign convention of fluxes.
- Changing the size of the smoothing window, or using variance control methods like Kalman filtering can help increase the signal-to-noise ratio in very noisy datasets (Batt and Carpenter 2012).
- The estimate of vertical diffusion can be improved with high-resolution profiles and more advanced mechanistic calculation methods (Imberger 1985).
- The technique can be modified to work with discrete depth data (i.e. from sondes placed at fixed depths along the water column).

Further reading

Key References:

Obrador, B., Staehr, P.A., Christensen, J. 2014. Vertical patterns of metabolism in three contrasting stratified lakes. *Limnology and Oceanography* 59: 1228-1240.

Staehr, P.A., Christensen, J., Batt, R., Read, J. 2012a. Ecosystem metabolism in a stratified lake. *Limnology and Oceanography* 57: 1317-1330.

Other useful references:

Batt, R., Carpenter, S. 2012. Free-water lake metabolism: addressing noisy time series with a Kalman filter. *Limnology and Oceanography-Methods* 10: 23-30.

Bade, D. L. 2009. Gas exchange across the air-water interface. In *Encyclopedia of Inland Waters*. Academic Press. Oxford. pp 70–78.

Crusius, J., Wanninkhof, R. 2003. Gas transfer velocities measured at low wind speed over a lake. *Limnology and Oceanography* 48: 1010-1017.

Hanson, P.C., Carpenter, S.R., Kimura, N., Wu, C., Cornelius, S.P., Kratz, T. 2008. Evaluation of metabolism models for free-water dissolved oxygen methods in lakes. *Limnology and Oceanography: Methods* 6: 454-465.

Hondzo, M., Stefan, H. G. 1993. Lake water temperature simulation model. *Journal of Hydraulic Engineering* 119: 1251-1273.

Honti, M. 2016. Bayesian calibration of mechanistic models of lake metabolism. In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 8). Technical report. NETLAKE COST Action ES1201. pp. 40-46.

<http://eprints.dkit.ie/id/eprint/539>.

Imberger, J. 1985. Thermal characteristics of standing waters: an illustration of dynamic processes. *Hydrobiologia* 125: 7-29

Read, J.S., Hamilton, D.P., Jones, I.D., Muraoka, K., Kroiss, R., Wu, C.H., Gaiser, E. 2011. Derivation of lake mixing and stratification indices from high-resolution lake buoy data. *Environmental Modelling and Software* 26: 1325–1336.

Staeher, P.A., Bade, D., van de Bogert, M.C., Koch, G.R., Williamson, C., Hanson, P., Cole, J.J., Kratz, T. 2010. Lake metabolism and the diel oxygen technique: State of the science. *Limnology and Oceanography Methods* 8: 628–644.

Staeher, P.A., Testa, J., Kemp, M., Cole, J.J., Sand-Jensen, K., Smith, S. 2012b. The metabolism of aquatic ecosystems: History, applications, and future challenges. *Aquatic Sciences* 74: 15–29

Woolway, R.I. 2016. Lake Metabolizer. In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 4). Technical report. NETLAKE COST Action ES1201. pp. 16-22. <http://eprints.dkit.ie/id/eprint/535>.

Code

The code for this technique was written in Statistical Analysis System (SAS) and is available upon request.

Contact details

Biel Obrador. University of Barcelona, Spain.
obrador@ub.edu

Suggested citation

Obrador, B., Christensen, J. and Staeher, P.A. 2016. Determination of whole-column metabolism from profiling data. In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 9). Technical report. NETLAKE COST Action ES1201. pp. 47-51. <http://eprints.dkit.ie/id/eprint/540>.

Acknowledgement

This factsheet is based upon work from the NETLAKE COST Action ES1201, supported by COST (European Cooperation in Science and Technology).

NETLAKE toolbox for the analysis of high-frequency data from lakes



Factsheet #10

Pattern detection using Dynamic Factor Analysis (DFA)

Rosana Aguilera and Rafael Marcé

Objective

One of the main applications of time-series analysis is the identification of trends and cyclic patterns in the data. Many trend detection and frequency decomposition analyses already exist for those purposes, particularly to address single time-series. However, classical methodologies are not particularly well suited to cope with multivariate problems. Dynamic Factor Analysis (DFA) decomposes a collection of time-series into common patterns and associated error terms (Zuur *et al.* 2003a). Broadly speaking, this method resembles performing Principal Component Analysis (PCA) but it is specifically designed for time-series. The end-product is a collection of patterns shared by all time-series, the relative relevance of each pattern across time-series, and error terms.

DFA is a dimension-reduction method that estimates underlying common patterns in a set of time-series (Zuur *et al.* 2003a). An attractive feature of DFA is its ability to treat time-series that have been recorded irregularly over time, or have short duration. Moreover, DFA allows time-series to be short and thus the lack of sufficiently long records does not represent a problem (Zuur and Pierce 2004). The extracted patterns (e.g., cycles and/or trends) are associated to factor loadings, which indicate the weight that each pattern has for each monitoring point. These two end products, i.e., patterns and factor loadings, can be then analyzed in order to characterize the temporal and spatial variability of the extracted water quality signals. The resulting description of the extracted patterns thus facilitates the interpretation and the identification of potential drivers of change in the system.

Specific application

The main application is the **detection of hidden patterns** that are shared by sets of time-series. If the time-series belong to a network of monitoring points, the spatial dimension can also be considered by looking at the relevance of each extracted pattern at each particular point, based on the magnitude and sign of the associated so-called factor loadings.

The greatest advantage of this method is its **ability to cope with missing observations and uneven sampling resolution** in time-series.

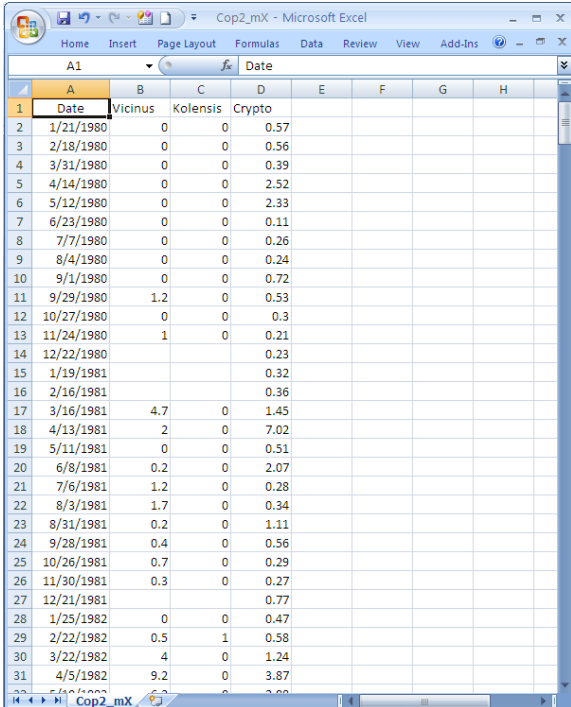
Background

The main tool is the **MARSS** (Multivariate Auto-regressive Space-State Model; Holmes *et al.* 2012) **R-Package**. A basic knowledge of 'R' would therefore be beneficial. Nevertheless, the MARSS manual and associated documents at the CRAN repository (<http://cran.r-project.org/web/packages/MARSS/index.html>) provide detailed information about setting up the DFA model.

It is also recommended to acquire some basic knowledge about time-series analysis (time-frequency domains, aliasing, autocorrelation function, etc.) before using DFA.

Type of data and requirements

The input files for DFA using the MARSS R-Package are .csv (Comma Separated Value) files with columns for each variable (an example is presented below). The data need not be standardized since the DFA script includes a previous data standardization step.



The screenshot shows a Microsoft Excel spreadsheet titled 'Cop2_mX'. The data is organized into columns: 'Date', 'Vicinus', 'Kolensis', and 'Crypto'. The rows represent time points from 1/21/1980 to 4/5/1982. The 'Date' column contains dates, while the other three columns contain numerical values representing abundances. The values for 'Vicinus' range from 0 to 4.7, 'Kolensis' from 0 to 1, and 'Crypto' from 0.11 to 3.87.

Date	Vicinus	Kolensis	Crypto
1/21/1980	0	0	0.57
2/18/1980	0	0	0.56
3/31/1980	0	0	0.39
4/14/1980	0	0	2.52
5/12/1980	0	0	2.33
6/23/1980	0	0	0.11
7/7/1980	0	0	0.26
8/4/1980	0	0	0.24
9/1/1980	0	0	0.72
9/29/1980	1.2	0	0.53
10/27/1980	0	0	0.3
11/24/1980	1	0	0.21
12/22/1980			0.23
1/19/1981			0.32
2/16/1981			0.36
3/16/1981	4.7	0	1.45
4/13/1981	2	0	7.02
5/11/1981	0	0	0.51
6/8/1981	0.2	0	2.07
7/6/1981	1.2	0	0.28
8/3/1981	1.7	0	0.34
8/31/1981	0.2	0	1.11
9/28/1981	0.4	0	0.56
10/26/1981	0.7	0	0.29
11/30/1981	0.3	0	0.27
12/21/1981			0.77
1/25/1982	0	0	0.47
2/22/1982	0.5	1	0.58
3/22/1982	4	0	1.24
4/5/1982	9.2	0	3.87

Figure 1. Example of input data for DFA analysis

In this case (Figure 1), three time-series are being considered in the analysis: the abundances of two copepods species (*Cyclops vicinus* and *Cyclops kolensis*) as well as the abundance of cryptophytes. The Date column is included here for posterior reference but it is omitted in the analysis. Of course, the columns may represent other arrangements (a variable in different locations, systems, etc).

The data need to be **evenly spaced**; i.e., the user must decide on a time-step if the data are unevenly sampled and adjust the observations to a specific resolution (e.g., monthly, daily, etc.). However, the analysis accepts missing values.

Basic procedures

1. Preparation of input file as indicated above.
2. Read MARSS documentation to understand the basic procedures and the different options of the analysis. Key model parameters are:
 - Number of common patterns (m) to be tested. The analysis does not automatically find the most efficient model in terms of the number of patterns to be extracted. So a trial-and-error procedure may be useful at this point, using a model selection criteria like Akaike Information Criterion (AIC), already included in the MARSS package.
 - Structure of error variance-covariance matrix (R). This tries to account for measurement errors and their covariance structure. Trial-and-error may also be needed here.

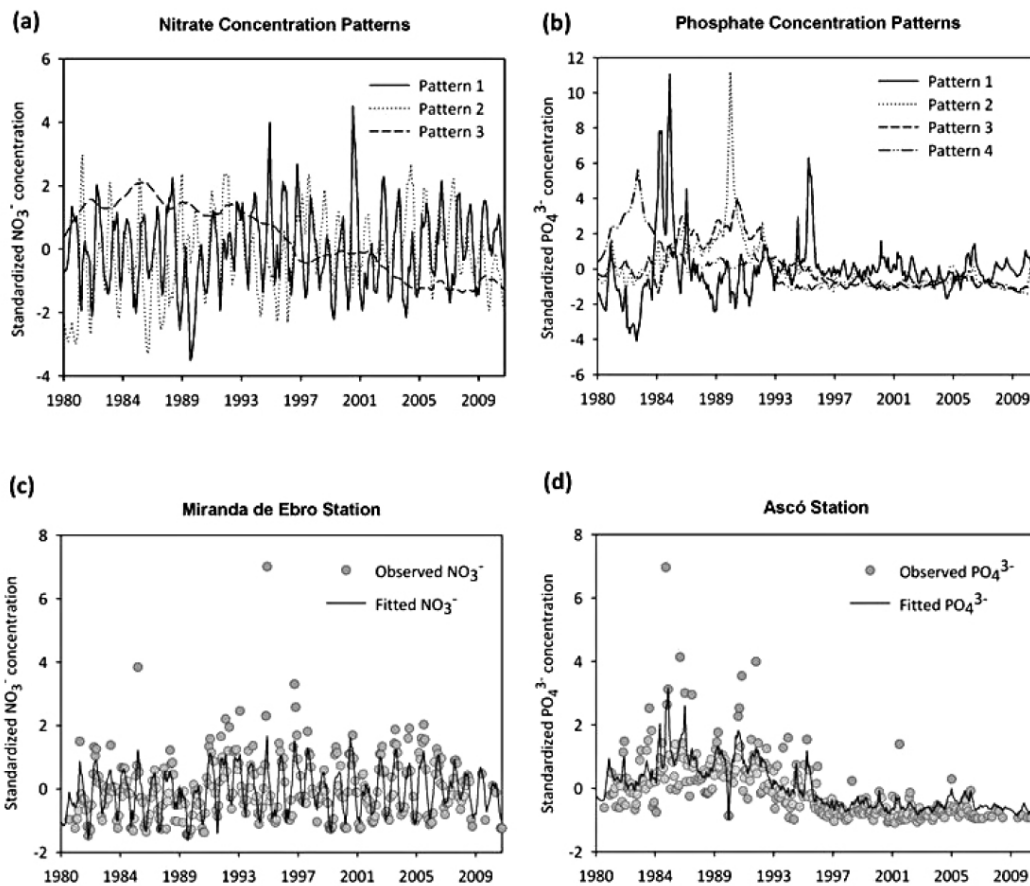


Figure 2. Patterns extracted from 50 time-series of nitrate (left) and phosphate (right) in different locations along the Ebro River basin, and the fit for two selected sampling points. The modelled lines in C and D are a linear combination of the patterns in A and B.

Pitfalls and tips

A visual assessment of the fit of the DFA model against the observed time-series may be useful to decide if the DFA analysis performed well (Figure 2). However, the absence of fit for a

variable or sampling location does not mean that the analysis is not working in that case, it may simply imply that there are no common patterns detected for that particular site or variable.

Depending on the number of time-series involved, as well as the length of data series, DFA can be time consuming and computationally demanding. Consider the use of a High Performance Computer if you are working with long series from many sites. At least you can run in parallel all trial-and-error runs related to parameter selection, which are independent.

Further reading

Key References:

Holmes, E.E., Ward, E., Wills, K. 2012. MARSS: Multivariate Autoregressive State-space Models for analyzing Time-series Data. *The R Journal* 4: 11-19.

Zuur, A.F., Fryer, R.J., Jolliffe, I.T., Dekker, R., Beukema, J.J. 2003a. Estimating common trends in multivariate time series using dynamic factor analysis. *Environmetrics* 14: 665-685.

Other useful references:

Aguilera, R., Marcé, R., Sabater, S. 2015. Detection and attribution of global change effects on river nutrient dynamics in a large Mediterranean basin. *Biogeosciences* 12: 4085–4098.

Holmes, E.E. 2013. Derivation of the EM algorithm for constrained and unconstrained multivariate autoregressive state-space (MARSS) models. Technical Report. *arXiv preprint arXiv: 1302.3919*.

Zuur, A.F., Ieno, E.N., Smith, G.M. 2007. *Analysing ecological data*. Springer. New York.

Zuur, A.F., Pierce, G.J. 2004. Common trends in Northeast Atlantic Squid time series. *Journal of Sea Research* 52: 57-72.

Zuur, A.F., Tuck, I.D., Bailey, N. 2003b. Dynamic factor analysis to estimate common trends in fisheries time series. *Canadian Journal of Fisheries and Aquatic Sciences* 60: 542-552.

Code

The code for this technique was written in the 'R' language and is available in the MARSS package. For a complete application of the technique including several sampling locations you may use the 'R' code by Rosana Aguilera and included in her PhD project (contact her for a copy of the R-codes).

Contact details

Rosana Aguilera. Catalan Institute of Water Research, Girona, Spain.
r.aguilera@icra.cat

Rafael Marcé. Catalan Institute of Water Research, Girona, Spain.
rmarce@icra.cat

Suggested citation

Aguilera, R. and Marcé, R. 2016. Pattern detection using Dynamic Factor Analysis (DFA). In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 10). Technical report. NETLAKE COST Action ES1201. pp. 52-56. <http://eprints.dkit.ie/id/eprint/541>.

Acknowledgement

This factsheet is based upon work from the NETLAKE COST Action ES1201, supported by COST (European Cooperation in Science and Technology).

NETLAKE toolbox for the analysis of high-frequency data from lakes



Factsheet #11

Inferential modelling of time series by evolutionary computation

Friedrich Recknagel and Ilia Ostrovsky

Objective

The hybrid evolutionary algorithm (HEA) has been designed: 1) to represent and forecast multivariate relationships between environmental conditions and population densities by inferential (IF-THEN-ELSE) models, and 2) to quantify ‘tipping points’ for population outbreaks by IF-conditions (Figure 1). During the course of hundreds of iterations, HEA discovers the ‘best-fitting’ model after optimising model structures by genetic programming and model parameters by differential evolution towards the lowest RMSE and highest R^2 (Cao et al. 2013).

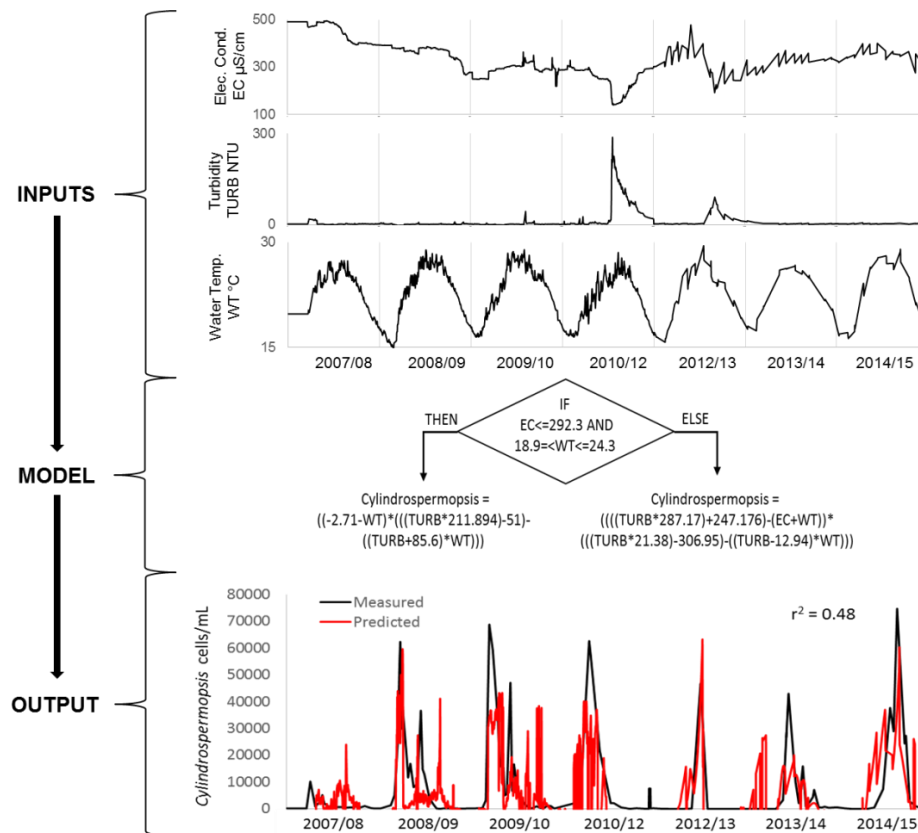


Figure 1. 20-day-ahead forecasting of *Cylindrospermopsis raciborskii* in Lake Wivenhoe (Australia) by means of inferential modelling based on HEA. The IF-condition suggests that fast population growth of *C. raciborskii* in Lake Wivenhoe may occur within the temperature range of 18.9 to 24.3 °C and at conductivity levels lower than 292 μS/cm.

The forecasting accuracy of inferential models by HEA suits early warning of population outbreaks. Ensembles of inferential models allow scenario analysis of how shifts in physical-chemical boundaries impact on aquatic communities. Meta-analysis of ‘tipping points’ and ecological relationships across lakes with the same stratification regime and trophic state allows the generalisation of knowledge inherent in complex ecological data.

Specific application

Quantifying ecological tipping points and relationships has been demonstrated successfully by case studies for Lakes Müggelsee (Germany), Kinneret (Israel), Taihu (China) and Lajes (Brazil) (Recknagel et al. 2016; Recknagel et al. 2015; Recknagel et al. 2014; Recknagel et al. 2013). **Short-term forecasting and early warning** of cyanobacteria blooms as well as meta-analysis of tipping points have been demonstrated successfully by case studies for Lakes Wivenhoe, Somerset and Samsonvale (Australia) (Recknagel et al. 2014). **Spatially-explicit short-term forecasting** of cyanobacteria blooms has been demonstrated successfully by case studies for Lakes Lajes (Brazil), Taihu (China) and Wivenhoe (Australia) (Recknagel et al. 2015; Zhang et al. 2015; Cao et al. 2016).

Background

The tool is available as user-friendly software written in C++. To use the tool requires basic programming skills. To execute evolutionary computations by HEA can be very time-consuming. It is therefore recommended to run HEA on supercomputers in cloud mode.

Type of data and requirements

Ecological time series are recorded in .xls spreadsheets where rows contain input- and output parameters of interest (e.g. physical, chemical and biological data) for consecutive equidistant time steps. Since the HEA software learns from patterns, modelling of seasonal and inter-annual dynamics requires at least 3 years of data, but it generalises best with decades of data containing a wealth of patterns. If data are missing or have been measured at non-equidistant time steps, interpolation of data to the smallest measured time step is required (HEA licence includes a software tool for flexible linear data interpolation of time series). Whilst ‘day’ is the recommended time step for ‘several-day-ahead’ predictive modelling, there is no restriction to the choice of the smallest time step. Data for spatially-explicit modelling of same ecological attribute measured simultaneously at multiple sites has the same requirements as for modelling single-site data (HEA licence includes detailed manual and data examples for single- and multi-site modelling experiments).

The .xls spreadsheets need to be completed by specifying HEA control parameters such as numbers of inputs, outputs, generations, boot-strap loops etc. before being saved as Text (Tab

delimited) files. To run HEA, the HEA *exe*-file together with the Text file need to be submitted to a supercomputer.

Basic procedures

1. Prepare equidistant input and output data as well as HEA control parameters in .xls files before saving them as Text (Tab delimited) files.
2. Submit HEA *exe*-file together with Text file to supercomputer.
3. Review the modelling protocol documenting 10 'best fitting' models by: IF-THEN-ELSE rules, graphical validation, root mean squared error (RMSE), R^2 , ranking inputs by sensitivity, input sensitivity functions.

Pitfalls and tips

- Since HEA ranks inputs by sensitivity after each run, noise from the least sensitive inputs can be removed for consecutive runs that may improve model validity.
- To avoid bias by relying on a single model, averages and Min-Max envelopes of an ensemble of 3 to 5 best-fitting models can be utilised for validation.
- Since HEA infers IF-THEN-ELSE rules for the underlying research question, the IF conditions reveal quantitative thresholds that explain causes for high and low output magnitudes.

Further reading

Key References:

Cao, H., Recknagel, F., Orr, P. 2014. Parameter optimisation algorithms for evolving rule models applied to freshwater ecosystem. *IEEE Transactions on Evolutionary Computation* 18: 793-806.

Cao, H., Recknagel, F., Bartkow, M. 2016. Spatially-explicit forecasting of cyanobacteria assemblages in freshwater lakes by multi-objective hybrid evolutionary algorithms. *Ecological Modelling*, 342, 97-112.

Recknagel, F., Adrian, R., Köhler, J., Cao, H. 2016. Threshold quantification and short-term forecasting of *Anabaena*, *Aphanizomenon* and *Microcystis* in the polymictic eutrophic Lake Müggelsee (Germany) by inferential modelling using the hybrid evolutionary algorithm HEA. *Hydrobiologia* 778: 61-74.

Other useful references:

Recknagel, F., Branco, C.W., Cao, H., Huszar, V.L., Sousa-Filho, I.F. 2015. Modelling and forecasting the heterogeneous distribution of picocyanobacteria in the tropical Lajes Reservoir (Brazil) by evolutionary computation. *Hydrobiologia* 749: 53-67.

Recknagel, F., Orr, P., Cao, H. 2014. Inductive reasoning and forecasting of population dynamics of *Cylindrospermopsis raciborskii* in three sub-tropical reservoirs by evolutionary computation. *Harmful Algae* 31: 26–34.

Recknagel, F., Ostrovsky, I., Cao, H. 2014. Model ensemble for the simulation of plankton community dynamics of Lake Kinneret (Israel) induced from in situ predictor variables by evolutionary computation. *Environmental Modelling & Software* 61: 380-392.

Recknagel, F., Ostrovsky, I., Cao, H., Chen, Q. 2014. Hybrid evolutionary computation quantifies environmental thresholds for recurrent outbreaks of population density. *Ecological Informatics* 24: 85–89.

Recknagel, F., Ostrovsky, I., Cao, H., Zohary, T., Zhang, X. 2013. Ecological relationships, thresholds and time-lags determining phytoplankton community dynamics of Lake Kinneret, Israel elucidated by evolutionary computation and wavelets. *Ecological Modelling* 255: 70-86.

Zhang, X., Recknagel, F., Chen, Q., Cao, H., Li, R. 2015. Spatially-explicit modelling and forecasting of cyanobacteria growth in Lake Taihu by evolutionary computation. *Ecological Modelling* 306: 216-225.

Code

HEA has been coded in C++ language and is not yet freely available. The authors offer short courses on inferential and process-based modelling, and welcome collaboration on data processing and modelling (for more details please contact friedrich.recknagel@adelaide.edu.au).

Contact details

Friedrich Recknagel. University of Adelaide, School of Biological Sciences, Adelaide, Australia. friedrich.recknagel@adelaide.edu.au

Iliia Ostrovsky. Israel Oceanographic & Limnological Research, Kinneret Limnological Laboratory, P.O.B. 447, Migdal 14950, Israel. ostrovsky@ocean.org.il

Suggested citation

Recknagel, F. and Ostrovsky, I. 2016. Inferential modelling of time series by evolutionary computation. In Obrador, B., Jones, I.D. and Jennings, E. (Eds.) *NETLAKE toolbox for the analysis of high-frequency data from lakes* (Factsheet 11). Technical report. NETLAKE COST Action ES1201. pp. 57-60. <http://eprints.dkit.ie/id/eprint/542>.

Acknowledgement

This factsheet is based upon work from the NETLAKE COST Action ES1201, supported by COST (European Cooperation in Science and Technology).