**WATExR tool suggested workflow**

*LJB. Last update: 28/10/2019*

**Brief overview:**

Aim to predict lake concentration of total phosphorus (TP), chlorophyll-a (chl-a) and biovolume of cyanobacteria. Predictions will be made using a Bayesian Belief Network (BBN), driven using observed water chemistry and ecology from the previous summer, and observed or forecasted meteorological variables. Predictions will be made for the 3 months after the current month. For Vansjø, the WFD class is decided on based on the condition of the lake during the months May - October. So preliminary splitting of the year into seasons will be:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Season** | **Lake forecast produced in month** | **Forecast produced for months** | **Met data to parameterise & drive the BN model** | **Variables included in forecast** |
| Early summer | April | May, June, July | Previous winter (Oct-March): ERA-interim reanalysis\*  May - July: seasonal forecast\* | Met (rain, pptn, wind)  Lake chem + ecol |
| Late summer | July | Aug, Sep, Oct | [May-June: ERA-interim reanalysis\*, July: seasonal data\*]\*\*  Aug - Oct: seasonal data\* | Met (rain, pptn, wind)  Lake chem + ecol |
| Winter | Oct | Nov, Dec, Jan | Nov - Jan: seasonal forecast\* | Met (rain, pptn, wind) |
| Spring | Jan | Feb, March, April | Feb - Apr: seasonal forecast\* | Met (rain, pptn, wind) |

**\*** All bias corrected. Data used for bias correction depends on whether the models are being run in hindcast or operational mode. Hindcast: EWEMBI, operational: met.no.

\*\* May not be required, depending on final BN structure decided on.

Water quality/ecology predictions will only be produced for Early and Late summer. Then, we are interested in variables (to start with):

* Total P concentration
* Chl-a concentration
* Cyanobacteria biovolume and maximum value for the season
* Overall WFD status (calculated as a function of the other variables)
* (Other possibles: PTI, colour,…)

Predictions will only be produced for Vanemfjorden, the western basin in Lake Vansjø (often referred to as Van2).

Note: GOTM has also been set up for the lake, and there is the possibility of including GOTM predictions too (e.g. ice cover duration).

**Coding aims:**

1. **Automated workflow for us to be able to run several hindcast experiments**

Hindcast experiments are described in a document in the WATExR Dropbox folder (…\Dropbox\WATExR\WP4\_Assessment\Draft\_Protocol\_WATExR\_hindcast\_runs\_v3.docx). They can be broken into the following 3 tasks:

* **Task 1 - assess lake model forecasting skill**: parameterise and run model driven by best possible observed met data, with cross-validation to get predictive uncertainty.

Met data: met.no data for temp and precipitation, Rygge station data for wind speed. Data has been downloaded: GitHub\WATExR\Norway\_Morsa\Data\Meteorological\Obs\_metno

* **Task 2 - generate pseudo observations for the period 1981-2010**: parameterise and then drive model using bias-corrected reanalysis data (ERA-interim). Model output from this will be assumed to be ‘true’ (to provide longer time series of “observed” water quality than observations alone)

Met data: ERA-interim reanalysis data, bias corrected using EWEMBI

* **Task 3 - assess seasonal met forecasting skill in the context of lake forecasting**: run model with ERA-interim for warmup and seasonal forecast S4 for target season (both bias-corrected), and compare output to pseudo-observations from Task 2.

Met data: ERA-interim reanalysis data and System4 seasonal forecast data, both bias corrected using EWEMBI

For each task, involves running model for each year/season in the hindcast period.

Date of the hindcast period (incl.):

* Task 1: 1989-2018
* Task 2 & 3: 1981-2010

1. **Operational workflow to predict water quality in the coming season**

For the two summer seasons, we will gather historic met and chem data from the previous year, as well as seasonal forecast data for the coming season, to be able to predict lake chemistry and ecology in the coming season. This will probably involve the following data:

* Historic observed met: ERA-interim data (perhaps ERA5?), bias corrected using met.no or local observed data
* Seasonal met data: possibly S5, bias corrected using met.no/local observed data

**Priorities, suggested workflow and environment**

Our first priority is the hindcast experiment to assess the skill of the models and seasonal forecasts, but we should try to make any code we produce for this as clean as possible so it can be incorporated within the operational workflow.

Development environment:

* ICRA WATExR GitHub repository, Norway\_Morsa folder: <https://github.com/icra/WATExR/tree/master/Norway_Morsa>
* NIVA’s Jupyter Hub

I would suggest we start by making a set of Jupyter notebooks which independently do the required steps, so we can easily visualise output. Later, we can convert the contents of these notebooks into python functions, which can be called from a master script (either a notebook which we could ‘GUI-ise’ using e.g. ipyWidgets, or make e.g. an RShiny app)

Suggested starting point, ~8 notebooks (4 for the hindcast experiment, 4 similar ones for the operational forecasting):

**Hindcast experiment**

**NB1a\_hindcast: Run the Climate4R scripts to download met data for the hindcast experiment**

* The Climate4R package is on R. The WATExR-specific scripts are in the Github repository: <https://github.com/icra/WATExR/tree/master/R>. See the readme there for a description of the different files.
* @sixtoherrera and @miturbide on Mattermost are very helpful and quick at troubleshooting, ask them if there are any issues. Flick back up through the discussions too for problems people have had and potential solutions.
* Jose got the data download and bias correction scripts working for Vansjø, and has put them into Jupyter Notebooks. Notebooks are here: <https://github.com/icra/WATExR/tree/master/Norway_Morsa/MetData_Processing/notebooks>

(To do: double-check bias correction settings are right. Leah asked via Mattermost 27/10/19)

* Leah got observations.R and seasonalForecast.R working for Vansjø last November in Magdeburg. Files are in <https://github.com/icra/WATExR/tree/master/Norway_Morsa/Climate4R_Scripts>, in the ‘Magdeburg’ folder. Changes compared to master scripts:
* Login details (not provided in the GitHub scripts): loginUDG("WATExR", "1234567890")
* Vansjø-specific stuff: changing the lat and long variables (domain extent and location of the lake), plus a few other bits and pieces
* Some more comments to explain what bits are doing

Scripts will probably have changed a bit now, and we want more variables than just in these example scripts, but they should be useful anyway.

Notebook aims:

* Download historic observed meteorological data to parameterise/warm up models or run a baseline period:
  + For Task 1: Download local observed met data

(Done to mid 2018, data saved here: <https://github.com/icra/WATExR/tree/master/Norway_Morsa/Data/Meteorological/Obs_metno>. Jose to update to include all of 2018. Jose did a notebook which does at least some of this, in <https://github.com/icra/WATExR/tree/master/Norway_Morsa/MetData_Processing/notebooks>)

* + - met.no’s 1km x 1km gridded data for the Morsa catchment
    - Combine with wind data from Rygge airport (downloaded to Dec 2019)
  + For Task 2: Download global observed/reanalysis met data. Think this is done (JLG).
    - EWEMBI and ERA-interim data. ERA-interim data is here: <http://meteo.unican.es/tds5/dodsC/interim/interim075_WATExR.ncml>

EWEMBI is also on the unican server.

* + - Bias-correct ERA-interim data using EWEMBI (using a function in the Climate4R package.  [See: https://github.com/SantanderMetGroup/downscaleR/wiki/bias-correction-of-seasonal-forecasts’](https://github.com/SantanderMetGroup/downscaleR/wiki/bias-correction-of-seasonal-forecasts))
  + Rearrange into nice Python structure for accessing later
  + Out of interest, compare bias in EWEMBI and ERA-interim data with met.no data. Done (JLG), but results only posted up on Slack. Jose to make a little word doc.
* Download seasonal forecast met data for the hindcast period and bias correct:
  + **Data**: System4. Has 15’ish members, which are different initializations
  + **Variables of interest**
    - For BN: precipitation, mean air temp, wind speed
    - For the future, probably also worth downloading as a one-off (for SimplyP/GOTM): min and max temp (to calculate PET), humidity, radiation, atmospheric pressure. Climate4R scripts calculate PET, so run those bits of the code.
  + **Download** seasonal data.

We need different seasons for the two models that we’re going to run the hindcast experiment for (GOTM and the Bayesian Network). For both models, for each season in the hindcast period:

Select lead time 0. Download for four seasons, for period 1981-2010:

* + - GOTM (Jose doing this):
      * Spring (March-May): download Feb, March, Apr, May
      * Summer (Jun-Aug): download May, June, July, Aug
      * Autumn (Sep-Nov): download August, Sep, Oct, Nov
      * Winter (Dec-Feb): download Nov, Dec, Jan, Feb
    - Bayesian network:
      * Early summer: May-Jul
      * Late summer: Aug-Oct
      * Winter: Nov-Jan
      * Spring: Feb-Apr
    - SimplyP+MyLake: same as BN, but with one extra month at start for warm-up:
      * Early summer: April, May-Jul
      * Late summer: Jul, Aug-Oct
      * Winter: Oct, Nov-Jan
      * Spring: Jan, Feb-Apr
  + **Bias correct** seasonal forecast data using EWEMBI data and calculate PET if not already done (or bias-correct PET too?)
  + Rearrange data into some kind of nice python structure that we can work with. E.g. something like a daily\_met\_dict with key [season, year, variable], which returns a pandadas dataframe with datetime index and one column for each ensemble member.
* QC: Time series plots and summary stats to check all looks ok.
* Tercile plot comparing observed and seasonal seasonal forecast data for temp, precipitation, wind speed (Climate4R package has a function for this in visualizeR).
* Save into a nice structure for access by other notebooks. E.g. a dictionary with one key per dataset (observed, reanalysis, seasonal)? **See NB2a point 1** for what’s coming next for the met data. Good to have this in mind when saving it.

**NB2a\_hindcast: Generate input data for BN**

All these features are worked out in notebooks in folder <https://github.com/icra/WATExR/tree/master/Norway_Morsa/Model_Development/Notebooks/01_MakeDataMatrices>, so code can be reused from there.

1. Post-processing of met data. This list is preliminary (exact features will be decided on by late October 2019, once the final BBN structure is decided on)
   1. Input met data for Tasks 1 and 2:

For both observed met data and bias-corrected ERA-Interim data:

For each season/year:

* + 1. Sum of winter precipitation for the winter before the current season
    2. Mean seasonal air temp
    3. Seasonal precipitation sum
    4. Something wind-related. e.g. count of number of days per season when daily mean wind speed was less than 3 m/s (LJB to specify).
  1. [Input to forecast late summer lake status (Task 3):

Create a realistic patched met dataset, which represents what would be available to us when producing forecasts in July of lake status in Aug-Oct:

* + - Patch together bias-corrected ERA-interim data from May and June with seasonal forecast data from July, to create one continuous data series (May-July)
    - Calculate a(ii) – a(iv) on this early summer met season dataset (which will be used to predict the lake in late summer).

Current BN structure doesn’t require this, so leave for now.]

* 1. For the seasonal forecast data: For all seasons/years/ensemble members, convert to seasonal frequency and calculate a(ii)-a(iv)

1. Water chemistry and ecology data:
   * Read in observed chemistry and ecology data, stored here (one file for cyanobacteria, everything else is in the other file: GitHub\WATExR\Norway\_Morsa\Data\Observed\_Chem\_Ecol

We’re interested in variables: **TP and chl-a concentration, cyanobacterial biovolume, lake colour**. Everything else can be dropped.

* + Join data, drop anything outside the hindcast period
  + Calculate mean seasonal water chemistry and ecology (to compare to simulations), so have a dataframe with an index something like (year, season)
  + Calculate means for the previous summer (explanatory variable)

1. Store all target and explanatory variables in a nice big dataframe

N.B. we are also planning on running the hindcast experiments with SimplyQ-GOTM, and hopefully also with SimplyP-GOTM-FABM. Therefore will need to generate input data for these models. This will likely involve patching together ERA-interim and S4 data (bias corrected with EWEMBI) for the warm-up period, with S4 (bc’d) for the target seasons. Note the seasonal split of the experiments is different for GOTM, which will be comparable with all the other WATExR lake physics models, than for SimplyP-GOTM-FABM (which will be the same as the BN). Magnus and FCL responsible for GOTM and can probably take responsibility for SimplyP-MyLake too.

**NB3a\_hindcast: Set up and run the Bayesian Belief Network (BBN)**

1. Define the BBN structure (read in bn.fit object? Or structure and data used to create fit)
2. For each kind of met data (met.no, ERA-Interim, S4/S5):

For each season, year (and for S4/5 each member in the seasonal forecast ensemble):

* Using the features generated in NB2, make predictions for the target season for the vars of interest
* Save in nice format for export to elsewhere (probabilities of being in different WFD classes, and predicted class per variable and overall). Something like a dictionary with key (season, year, result variable), returns dataframe with columns for each ensemble member and 3 rows (‘High’, ‘Good’, ‘Moderate’). The values in the cells are a probability of being in that row (sum of probabilities across rows adds up to 1).

If seasonal data, for each season and year:

* 1. Aggregate results over ensemble members? Or do that in next notebook?

**NB4a\_hindcast: Calculate model performance and visualise results (probably easier to integrate with NB3a?)**

These steps are much less certain than the previous ones, needs more thinking through. Contributions welcome! Probably also a case of just getting started and solving problems as they arise

1. Observed data processing:
   1. Read in historic seasonal observed water quality/ecology data from NB2
   2. Reclassify into categories output by the BBN. E.g. see discretization boundaries in <https://github.com/icra/WATExR/blob/master/Norway_Morsa/Model_Development/Notebooks/01_MakeDataMatrices/C_Data_for_BN.ipynb> for example code and boundaries for the variables as a starting point. Here are WFD class boundaries too:

WFD\_class\_dict = {'TP\_lake': {'G-M':20., 'M-P':39.},

'chl-a\_lake': {'G-M':10.5, 'M-P':20.},

'CyanoBiovol': {'G-M':1., 'M-P':2.}

}

1. Read in forecasted water quality/ecology from NB3. As this data isn’t available yet, create dummy data.
2. For each season:
   1. For each month in the season:
      1. Generate tercile plots comparing observed and simulated data (using the Climate4R visualise script)
      2. Calculate some kind of goodness-of-fit statistic – see Climate4R visualise again for ideas?

Key questions: Better/worse performance for different variables? Better for certain seasons?

* 1. …?

1. Tercile plots & stats comparing seasonal met forecast and met.no observed precipitation, temperature and wind speed, plus GoF stats.

--------------------------------------------------------------------------------------------------------------------------------

**Forecasting**

Similar to the previous 4 notebooks, but this time the aim is to produce operational seasonal water quality forecasts:

**NB1b\_forecast: Run the Climate4R scripts to download met data for the operational forecast**

4 times a year:

1. Start date is the next month. Set the target season (early summer, late summer, winter, spring)
2. Download historic met data to update the model:
   1. Will probably be using ERA5 data, but discussions still ongoing as to the source of this. Hopefully Copernicus.
   2. Download daily precipitation, average temperature and wind speed for the previous ~year (to cover the previous winter anyway), to as recent as possible.
3. Download seasonal forecast met data for the (current month and?) coming season. Not sure what system we’ll be using, maybe System5?
4. Bias correct ERA5 and seasonal forecast data using met.no/local observed data (assuming historic bias applies)

**NB2b\_forecast: Source any chem/ecol data, post-process the data and generate features:**

Just a couple of differences to NB2a (mostly simplifications):

1. Water chemistry and ecology data:
   1. Read in observed lake chemistry and ecology data from the previous summer. From NIVABase (see James’ nivapy python package for convenience functions to do this if working on the network), Aquamonitor or Vannmiljø. Don’t spend long on this for now though - it wouldn’t take long to once a year (e.g. in February) extract relevant data and store it in a csv next to the scripts:
      1. From Station ID? Don’t know what the Van1 code is in Aquamonitor. I got data from Vannmiljø, Vannlokalitet\_kode: 003-30776 (Vannsjø, vanemfjorden (VAN2)
      2. Variables: (again, these are Vannmiljø- codes) P-TOT (Total P), KLFA (chl-a), FARGE (colour). Not sure about cyanobacteria, I got this data directly from lots of different people for the historic period as there are lots of gaps in the NIVABase data. There’s CYANOM, but that’s just the maximum for the season. Monthly values might be good too.
      3. Time period: previous ~18 months
   2. Calculate mean water chemistry for the previous summer (concentrations of TP, chl-a, cyanobacterial biovolume, lake colour)
2. Post-processing of met data. See NB2a.

**NB3b\_forecast: Set up and run the Bayesian Belief Network (BBN)**

Leah to do

1. Define the BBN structure
2. Read in the historic features used to create the BBN, updating with any more recent data
3. For each month in the coming season and ensemble member in the seasonal forecast:
   1. Run BBN using features generated in NB2 and save the output for all vars of interest (cyanobacterial biovolume, TP concentration, chl-a concentration, colour)
   2. Drop any warm-up months
   3. Calculate an ensemble median for each var of interest

**NB4b\_forecast: Visualise results – operational forecast, including pdf generation**

As for NB4a, but also:

1. Forecast for the coming season is compared to observations for previous seasons (probability of being below average, average or above average?)
2. **Auto-generate a pdf summarising results** (map of catchment, tercile plot, summary stats, skill score result from hindcast experiment, interpretative text).
3. This should use the elements in here as inspiration (but dropping all the time series plots): K:\Prosjekter\Ferskvann\O-17323 WATExR\05\_IntegratedTool\Norway\_Plugin\_Design\_2018-11.pptx

It also needs to be designed together with out stakeholder, and is a really important ouput.

1. Perhaps also auto-generate a simplified summary, suitable for emailing out to e.g. farmers.

----------------------------------------------------------------------------------------------------------------------------

Once the ‘b’ series of notebooks are done, we can think about moving a lot of the code into functions, and having a single place that we run to do everything from. In order of development priority, we’ll go for:

1. Jupyter notebook, which outputs directly to the notebook (plus generates pdf for emailing)
2. RShiny app. In first instance only accessible to people with niva affiliation, but plans to widen availability in future (and all code open source)

Sketch here for guidance on layout of output:

K:\Prosjekter\Ferskvann\O-17323 WATExR\05\_IntegratedTool\Norway\_Plugin\_Design\_2018-11.pptx

1. QGIS: I don’t see any life for a QGIS Plugin after the project, and there are lots of barriers to stakeholder use (esp. installation issues). So this is low priority for us. Could be explored though if people show easy, user-friendly ways of setting up R/Python together within QGIS at Magdeburg in autumn 2019.