Software components

Goal 1: dissolved methane calculation

1. get_data function

   Purpose: Acquired COD and BOD data from the spreadsheet.

   Input: Data is in the spreadsheet.

   Columns (input): COD, BOD, Temperature, Salinity, Sulfate, TSS, VSS, etc.

   Index (input): time in days

   Unit of values (input): mg/L, %, ℃, etc.

   Output: array of float (BOD), array of float (COD)

   Index (output): time in days

   Unit of values (output): mg/L

2. methane_production function

   Purpose: Calculating methane production from BOD. I decide to use BOD for further methane calculation instead of COD.

   Input: array of float (BOD)

   Index (input): time in days

   Unit of values (input): mg/L

   Output: numpy array (methane_production)

   Index (output): time in days

   Unit of values (output): ml/L

3. get_data_for_Bunsen_coefficient

   Purpose: Acquired salinity and methane data from the spreadsheet for calculating Bunsen coefficient.

   Input: Data is in the spreadsheet.

   Columns (input): COD, BOD, Temperature, Salinity, Sulfate, TSS, VSS, etc.

   Index (input): time in days

   Unit of values (input): mg/L, %, ℃, etc.

   Output: array of float (salinity), array of float (temperature)

   Index (output): time in days

   Unit of values (output): %, ℃

4. Temperature_conversion

   Purpose: Change the unit of temperature from Celsius to Kelvin.

   Input: array of float (temperature)

   Index (input): time in days

   Unit of values (input): ℃

   Output: numpy array (temperature)

   Index (output): time in days

   Unit of values (output): K

5. Bunsen_coefficient
   Purpose: Calculate Bunsen coefficient based on temperature and salinity.
   Input: numpy array (temperature_conversion), numpy array (salinity)
   Index (input): time in days
   Unit of values (input): K, %
   Output: numpy array (result_of_Bunsen_coefficient_input)
   Index (output): time in days
   Unit of values (output): ml CH_4/ ml H_2O
6. Dissolved_methane
   Purpose: Calculate dissolved_methane from Bunsen_coefficient and methane_production.
   Input: numpy array (result_of_Bunsen_coefficient_input), numpy array (methane_production)
   Index (input): time in days
   Unit of values (input): ml CH_4/ ml H_2O, ml/L
   Output: numpy array (dissolved_methane)
   Index (output): time in days
   Unit of values (output): ml/L
7. Molar_dissolved_methane
   Purpose: Calculate molar_dissolved_methane from dissolved_methane
   Input: numpy array (dissolved_methane), numpy array (temperature_conversion)
   Index (input): time in days
   Unit of values (input): ml/L, K
   Output: numpy array (molar_dissolved_methane)
   Index (output): time in days
   Unit of values (output): mmol/L
8. check_N_DAMO_process(minimum_DM_value)
   Purpose: Check if N-DAMO process exists based on the minimum value in the numpy array (molar_dissolved_methane).
   Input: numpy array (molar_dissolved_methane)
   Unit of values (input): mmol/L
   Output: string message

Goal 2: Evaluating nitrite and nitrate concentration by N-DAMO process based on dissolved methane calculating from goal 1.

1. plot_model
   Purpose: Create a roadrunner object and plots the model.
   Input: model string

*Output of this function will be present in the next function.

2. process_and_plot
   Purpose: Make the decision based on the N-DAMO process check result and plot the model if the process exists.
   Input: value of minimum molar_dissolved_methane, model string and the related value for N-DAMO reaction.
   Unit of values (input): mmol/L, 1/hour
   Output: string message and figure

Interactions to accomplish use cases

Goal 1: Interaction in dissolved methane calculation will give a sentence for preliminary evaluating if dissolved methane is enough for N-DAMO process. Data input for evaluation is in the spreadsheet. Data has different columns, which are different water quality indicators over time. Data used in this project are created from the effluent of an anaerobic fluidized bioreactor (Shin et al. 2014). Individuals can easily utilize their data, typically stored in spreadsheet formats. In this project, spreadsheet called “Reactor test” and “Reactor test2” can show the results of “N-DAMO process does not exist” and “N-DAMO process exists”, respectively. If the concentration of dissolved methane is insufficient, a notification will be displayed indicating that the N-DAMO process is not applicable to this wastewater. Consequently, the assessment of nitrite and nitrate concentrations will be terminated. If there is a sufficient amount of dissolved methane, the civil engineer can input the fraction of wastewater (with a default value of 0.3) designated for the assessment of the N-DAMO process. As a result, the civil engineer should have basic python ability to read the code and type the information.

Goal 2: Interaction for the user and nitrogen removal potential by N-DAMO process is that the user can see the figure showing up in their screen to see the concentration of nitrite, nitrate, nitrogen and methane over time. Civil engineer can thus get the preliminary result if the wastewater can conduct N-DAMO process and remove the targeted nitrite or nitrate.

Reference

Shin, Chungheon, Perry L.McCarty, JeonghwanKim, andJaehoBae. 2014. “Pilot-Scale Temperate-Climate Treatment of Domestic Wastewater with a Staged Anaerobic Fluidized Membrane Bioreactor (SAF-MBR).” *Bioresource Technology* 159: 95–103. https://doi.org/10.1016/j.biortech.2014.02.060.