3-day workshop on modelling lake ecosystems with WET:
Tutorial on auto-calibration with parsac

This tutorial introduces installation and application of the parsac sensitivity analysis and auto-calibration tool by Jorn Bruggeman of Bolding and Bruggeman ApS to a workable GOTM-WET model setup. On the use of parsac, this tutorial should be used as a supplement to the information provided on https://pypi.org/project/parsac/ and https://bolding-bruggeman.com/portfolio/parsac/ (please visit these sites for more information). For calibration, parsac applies the parallel direct search method Differential Evolution (Storn and Price 1997) to estimate the most optimal choice of model parameter values within predefined, parameter-specific ranges based on optimization of a Maximum Likelihood multi-objective function.

This tutorial builds upon the lecture on calibration and model evaluation of complex models and aims to teach how to perform supervised auto-calibration of your GOTM-WET model with parsac. The tutorial requires a workable GOTM-WET model set-up, observations to be used in the calibration and installed Python package parsac.

Note that the calibration is currently done independently from the QWET interface. In practice, users can first create a complete QWET model project, then exit the QWET (and QGIS) interface, and then navigate to the model directory (and work directly in this, or make a copy of this into a new folder for calibration purposes): \WET_model_example\project_name\LAKEMODEL\Default

<u>Tutorial materials</u>

Workable GOTM-WET setup for Shahe Reservoir

For our case study, we will work with the Chinese Shahe reservoir. See the QWET tutorials and exercise on how to configure a GOTM-WET model to Shahe Reservoir with lake specific hypsography and inflow and weather forcing as well as in-lake observations and files with calibrated parameters.

Observation files

In the bottom of the parsac xml configuration file, users must point to the file(s) containing observations, which will be used to train the model through the parameter calibration. The format of the observation files matches exactly that of the observation files that may imported into QWET.

<u>Observations for Shahe Reservoir are provided, so you do not need to prepare obs files for Shahe case study.</u>

Preparing .obs files for QWET or parsac

If you work with another case study, here is how to prepare .obs files. The observation files must comprise four columns including (in this order):

1. Date [yyyy-mm-dd]
2. Time [hh:mm:ss]
3. Depth from surface [m] (Negative values. For example, -1 represents 1 meters depth below surface)
4. Value of state variable concentration (in units corresponding to model output units, e.g. Celsius for temperature, mg/L for nutrients and oxygen, and µg/L for chlorophyll *a*)

An example of an **.obs* file of correct format can be found here. The names of the individual observations files must match the working .xml file in the "parsac" folder. For further details, please watch this tutorial video on how to format and work with observations in QWET.

parsac

parsac is a Python-based tool for sensitivity analysis and auto-calibration in parallel developed by Bolding&Bruggeman. It is designed for analysis of models that take significant time to run. To download parsac, you will need a Python working environment and can install via Python pip: "pip install parsac –user". Parsac requires the Python package Parallel Python to parallelize the auto-calibration process. For more details on parsac, we recommend checking out its Github page: https://github.com/BoldingBruggeman/parsac.

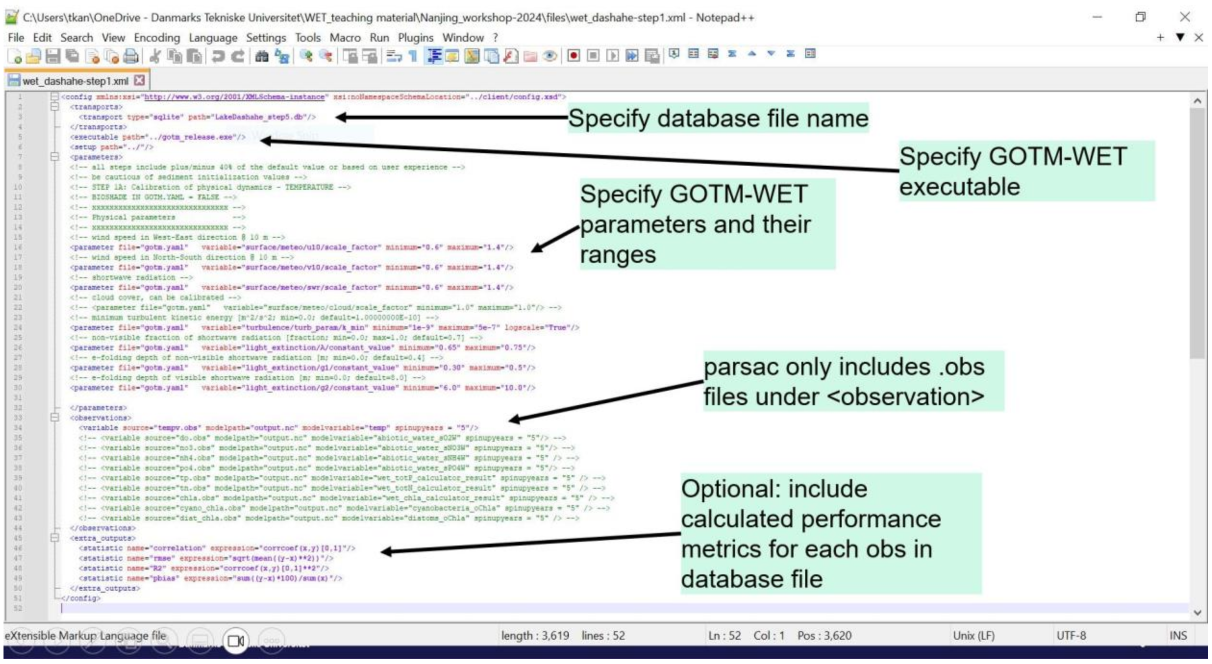
For installing Python 3, parsac, and required packages (pp (parallel python) and SALib), see installation guide in WeChat group.

DB Browser for SQLite (optional)
If you want to open the database file parsac creates with model simulation information, you can download the latest version DB Browser for SQLite or use the Windows installer "DB.Browser.for.SQLite-v3.13.0-win64.msi" in *Materials folder* provided on Day 1 by USB stick.

Folder setup for parsac
Within your working model folder, you will need to create a folder named *parsac*. In the *parsac* folder you should include observation files (*.obs) for Shahe Reservoir and a xml file with the configurations of parsac. Optionally, you can include bat files to execute the different parsac steps (see below).

Configuring parsac xml file
All calibration steps with parsac are configured through an xml file, which allows users to point to the parameters targeted for calibration and assign their ranges (and add or remove parameters for calibration). Be aware that parsac or GOTM-WET does not check if parameter values are within theoretical and technical ranges. So, users can specify parameter values that are not valid.

A comprehensive example of a xml file (wet_dashahe-step5.xml) is available in the wet-workshop-Day2\Exercises\Exercise 2 – Calibration.


C:\Users\tkan\OneDrive - Danmarks Tekniske Universitet\WET_teaching material\Nanjing_workshop-2024\files\wet_dashahe-step1.xml - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
wet_dashahe-step1.xml
Specify database file name
Specify GOTM-WET executable
Specify GOTM-WET parameters and their ranges
parsac only includes .obs files under <observation>
Optional: include calculated performance metrics for each obs in database file
eXtensible Markup Language file
length : 3,619 lines : 52 Ln : 52 Col : 1 Pos : 3,620 Unix (LF) UTF-8 INS

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="../client/config.xsd">
  <transports>
    <transport type="sqlite" path="LakeDashahe_step5.db"/>
  </transports>
  <executable path="../gotm_release.exe"/>
  <setup path="../"/>
  <parameters>
  <!-- all steps include plus/minus 40% of the default value or based on user experience -->
  <!-- be cautious of sediment initialization values -->
  <!-- STEP 1A: Calibration of physical dynamics - TEMPERATURE -->
  <!-- BIOSHADE IN GOTM.YAML = FALSE -->
  <!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
  <!-- Physical parameters            -->
  <!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
  <!-- wind speed in West-East direction @ 10 m -->
  <parameter file="gotm.yaml"   variable="surface/meteo/u10/scale_factor" minimum="0.6" maximum="1.4"/>
  <!-- wind speed in North-South direction @ 10 m -->
  <parameter file="gotm.yaml"   variable="surface/meteo/v10/scale_factor" minimum="0.6" maximum="1.4"/>
  <!-- shortwave radiation -->
  <parameter file="gotm.yaml"   variable="surface/meteo/swr/scale_factor" minimum="0.6" maximum="1.4"/>
  <!-- cloud cover, can be calibrated -->
  <!-- <parameter file="gotm.yaml"   variable="surface/meteo/cloud/scale_factor" minimum="1.0" maximum="1.0"/> -->
  <!-- minimum turbulent kinetic energy [m^2/s^2; min=0.0; default=1.00000000E-10] -->
  <parameter file="gotm.yaml"   variable="turbulence/turb_param/k_min" minimum="1e-9" maximum="5e-7" logscale="True"/>
  <!-- non-visible fraction of shortwave radiation [fraction; min=0.0; max=1.0; default=0.7] -->
  <parameter file="gotm.yaml"   variable="light_extinction/A/constant_value" minimum="0.65" maximum="0.75"/>
  <!-- e-folding depth of non-visible shortwave radiation [m; min=0.0; default=0.4] -->
  <parameter file="gotm.yaml"   variable="light_extinction/g1/constant_value" minimum="0.30" maximum="0.5"/>
  <!-- e-folding depth of visible shortwave radiation [m; min=0.0; default=8.0] -->
  <parameter file="gotm.yaml"   variable="light_extinction/g2/constant_value" minimum="6.0" maximum="10.0"/>

  </parameters>
  <observations>
    <variable source="tempv.obs" modelpath="output.nc" modelvariable="temp" spinupyears = "5"/>
    <!-- <variable source="do.obs" modelpath="output.nc" modelvariable="abiotic_water_sO2W" spinupyears = "5"/> -->
    <!-- <variable source="no3.obs" modelpath="output.nc" modelvariable="abiotic_water_sNO3W" spinupyears = "5"/> -->
    <!-- <variable source="nh4.obs" modelpath="output.nc" modelvariable="abiotic_water_sNH4W" spinupyears = "5" /> -->
    <!-- <variable source="po4.obs" modelpath="output.nc" modelvariable="abiotic_water_sPO4W" spinupyears = "5"/> -->
    <!-- <variable source="tp.obs" modelpath="output.nc" modelvariable="wet_totP_calculator_result" spinupyears = "5" /> -->
    <!-- <variable source="tn.obs" modelpath="output.nc" modelvariable="wet_totN_calculator_result" spinupyears = "5" /> -->
    <!-- <variable source="chla.obs" modelpath="output.nc" modelvariable="wet_chla_calculator_result" spinupyears = "5" /> -->
    <!-- <variable source="cyano_chla.obs" modelpath="output.nc" modelvariable="cyanobacteria_oChla" spinupyears = "5" /> -->
    <!-- <variable source="diat_chla.obs" modelpath="output.nc" modelvariable="diatoms_oChla" spinupyears = "5" /> -->
  </observations>
  <extra_outputs>
    <statistic name="correlation" expression="corrcoef(x,y)[0,1]"/>
    <statistic name="rmse" expression="sqrt(mean((y-x)**2))"/>
    <statistic name="R2" expression="corrcoef(x,y)[0,1]**2"/>
    <statistic name="pbias" expression="sum((y-x)*100)/sum(x)"/>
  </extra_outputs>
</config>
```

Calibration routine with parsac
The calibration routine consists of three actions in parsac: 1) executing the calibration, 2) plotting model performance against parameter value ranges and 3) running best performing model with their respective parsac commands in your Python console (for instance Anaconda Prompt):

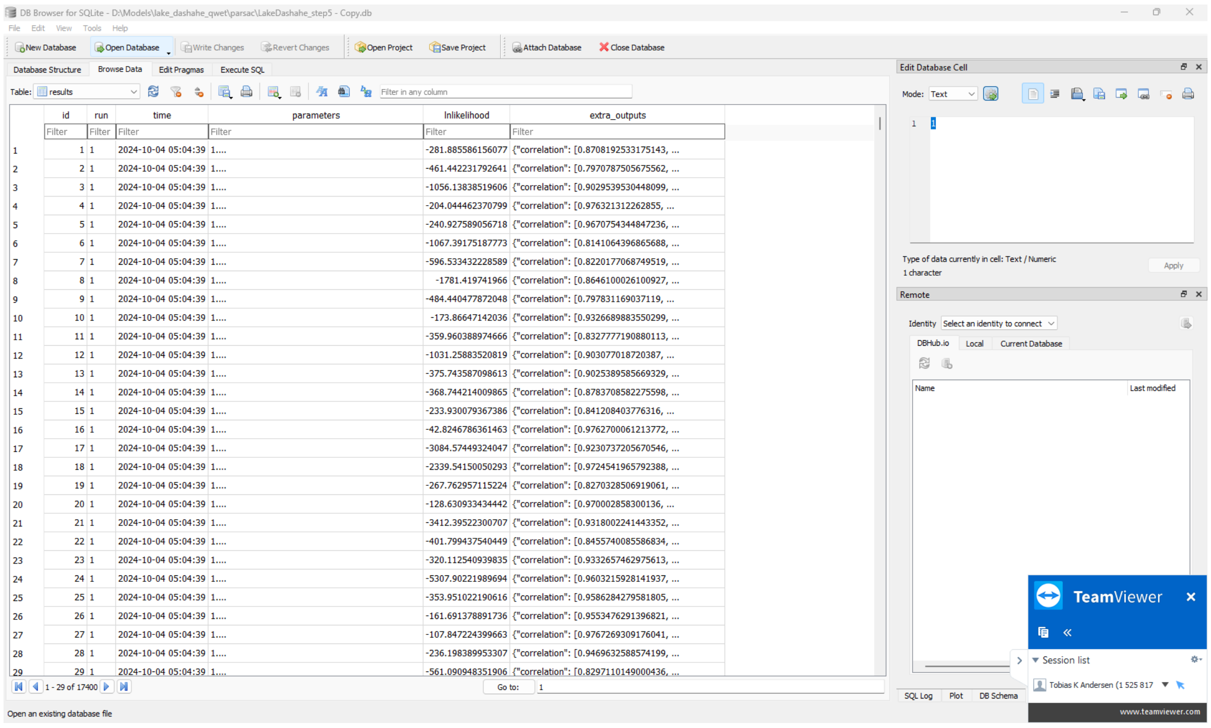
1. "parsac calibration run xmlfile"
2. "parsac calibration plot xmlfile"

3. “parsac calibration plotbest xmlfile”

xmlfile refers to the path and xml file name, for instance C:/”path_to_parsac_folder”/config_parsac.xml

In this tutorial, we refer to a calibration iteration when the modelers have completed the three actions above. Calibration of a GOTM-WET model with parsac will include many calibration iterations and will depend on the number of observations, state variable and parameters included. In this tutorial we also refer to calibration steps, which is a subset of all the calibration iterations that focus on calibrating a specific model process and/or reproduce specific observations (see *Calibration strategy and iterations* below for more information).

parsac run
During calibration (“parsac calibration run”), a database file (*.db, named as specified in the .xml file) is created to store each model simulation with unique model execution id (id), calibration run (run), selected values for parameters in .xml (parameters), ln-likelihood model performance (lnlikelihood) and optional performance metrics specified in .xml (extra_outputs). Calibration run is specified as several calibration iterations can be performed on the same xml file to the same database file, which is recorded by parsac.



	id	run	time	parameters	lnlikelihood	extra_outputs
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	1	2024-10-04 05:04:39	1....	-281.885586156077	{"correlation": [0.8708192533175143, ...
2	2	1	2024-10-04 05:04:39	1....	-461.442231792641	{"correlation": [0.7970787505675562, ...
3	3	1	2024-10-04 05:04:39	1....	-1056.13838519606	{"correlation": [0.9029539530448099, ...
4	4	1	2024-10-04 05:04:39	1....	-204.044462370799	{"correlation": [0.976321312262855, ...
5	5	1	2024-10-04 05:04:39	1....	-240.927589056718	{"correlation": [0.9670754344847236, ...
6	6	1	2024-10-04 05:04:39	1....	-1067.39175187773	{"correlation": [0.8141064396865688, ...
7	7	1	2024-10-04 05:04:39	1....	-596.533432228589	{"correlation": [0.8220177068749519, ...
8	8	1	2024-10-04 05:04:39	1....	-1781.419741966	{"correlation": [0.8646100026100927, ...
9	9	1	2024-10-04 05:04:39	1....	-484.440477872048	{"correlation": [0.797831169037119, ...
10	10	1	2024-10-04 05:04:39	1....	-173.86647142036	{"correlation": [0.9326689883550299, ...
11	11	1	2024-10-04 05:04:39	1....	-359.960388974666	{"correlation": [0.8327777190880113, ...
12	12	1	2024-10-04 05:04:39	1....	-1031.25883520819	{"correlation": [0.903077018720387, ...
13	13	1	2024-10-04 05:04:39	1....	-375.743587098613	{"correlation": [0.9025389585669329, ...
14	14	1	2024-10-04 05:04:39	1....	-368.744214009865	{"correlation": [0.8783708582275598, ...
15	15	1	2024-10-04 05:04:39	1....	-233.930079367386	{"correlation": [0.841208403776316, ...
16	16	1	2024-10-04 05:04:39	1....	-42.8246786361463	{"correlation": [0.9762700061213772, ...
17	17	1	2024-10-04 05:04:39	1....	-3084.57449324047	{"correlation": [0.9230737205670546, ...
18	18	1	2024-10-04 05:04:39	1....	-2339.54150050293	{"correlation": [0.9724541965792388, ...
19	19	1	2024-10-04 05:04:39	1....	-267.762957115224	{"correlation": [0.8270328506919061, ...
20	20	1	2024-10-04 05:04:39	1....	-128.630933434442	{"correlation": [0.970002858300136, ...
21	21	1	2024-10-04 05:04:39	1....	-3412.39522300707	{"correlation": [0.9318002241443352, ...
22	22	1	2024-10-04 05:04:39	1....	-401.799437540449	{"correlation": [0.8455740085586834, ...
23	23	1	2024-10-04 05:04:39	1....	-320.112540939835	{"correlation": [0.9332657462975613, ...
24	24	1	2024-10-04 05:04:39	1....	-5307.90221989694	{"correlation": [0.9603215928141937, ...
25	25	1	2024-10-04 05:04:39	1....	-353.951022190616	{"correlation": [0.9586284279581805, ...
26	26	1	2024-10-04 05:04:39	1....	-161.691378891736	{"correlation": [0.9553476291396821, ...
27	27	1	2024-10-04 05:04:39	1....	-107.847224399663	{"correlation": [0.9767269309176041, ...
28	28	1	2024-10-04 05:04:39	1....	-236.198389953307	{"correlation": [0.9469632588574199, ...
29	29	1	2024-10-04 05:04:39	1....	-561.090948351906	{"correlation": [0.8297110149000436, ...

1 - 29 of 17400

Weighting of observations with different variables
If several state variables and their observations are included in the calibration, parsac automatically weights the observations. As Jorn Bruggemann writes on parsac’s wiki page:

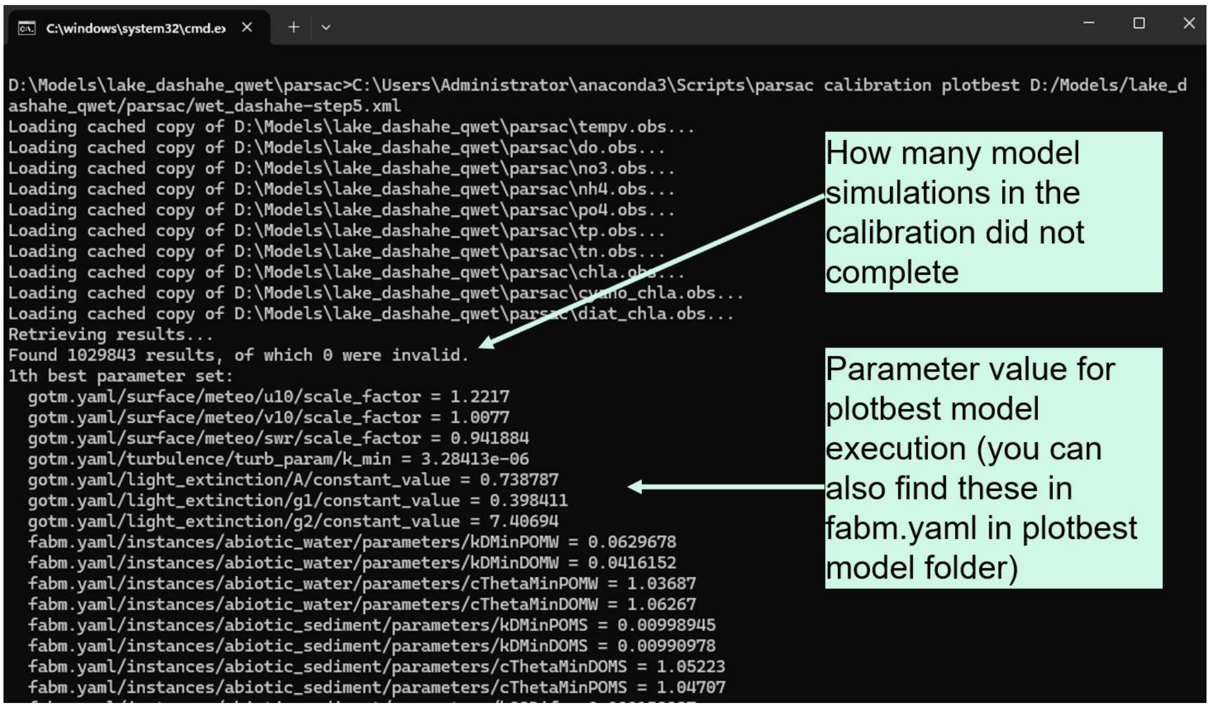
“parsac’s calibration approach maximizes the (log) likelihood. This combines all model-observations differences (after transformation, if that’s activated), each weighted by the reciprocal of its “standard deviation” (formally, squares of the differences divided by their variance). This standard deviation is currently a constant for each observed variable, it cannot yet depend on time and/or depth. It can be prescribed in your xml configuration file by adding an attribute sd="<VALUE>" to the observed variable. If it is not prescribed, parsac will instead estimate it from the model-observation differences. In that case, a variable that the model cannot capture well will automatically get a higher sd. All observations are assumed to be independent and (thus) uncorrelated.”

For details on the consequences of this approach, check out parsac’s wiki page.

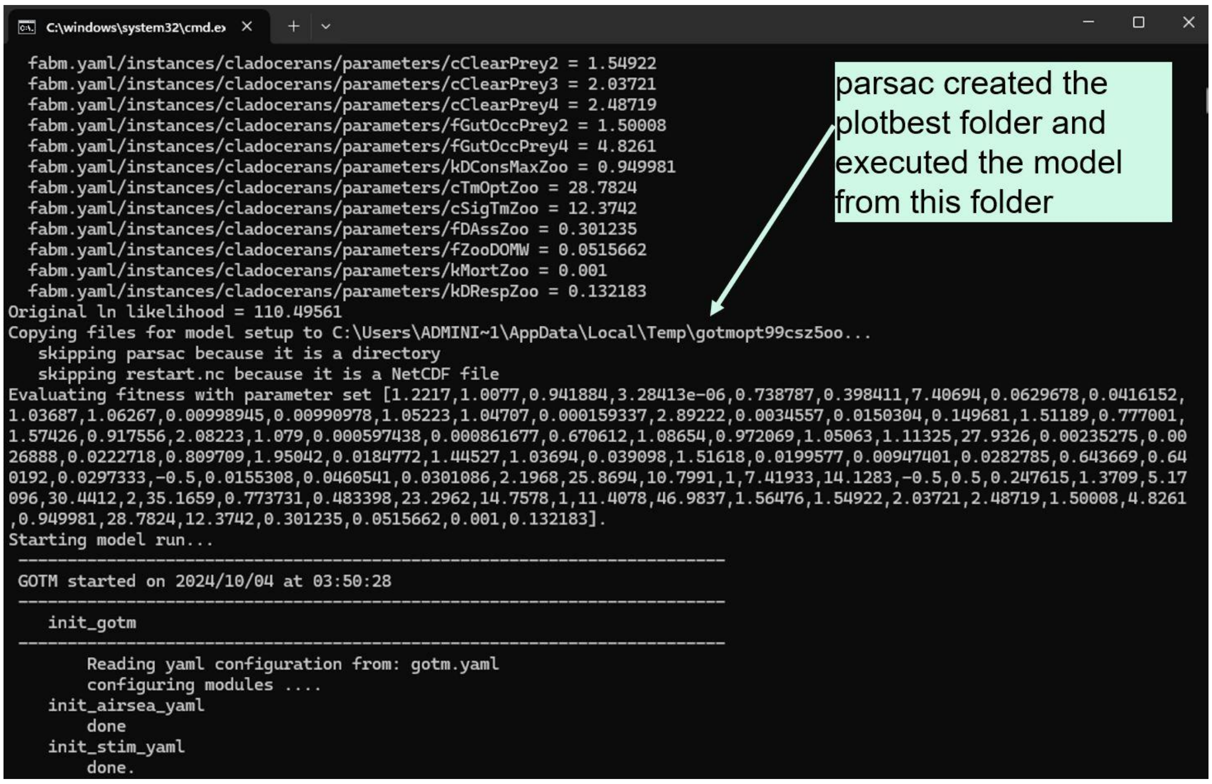
Notice that users can leverage sd="<VALUE>" to manually weigh the optimization and thereby focus the calibration on specified state variables with observations.

parsac plotbest
plotbest function will execute the GOTM-WET model with the best performance in the database file.

In the command window, parsac will print several useful information:

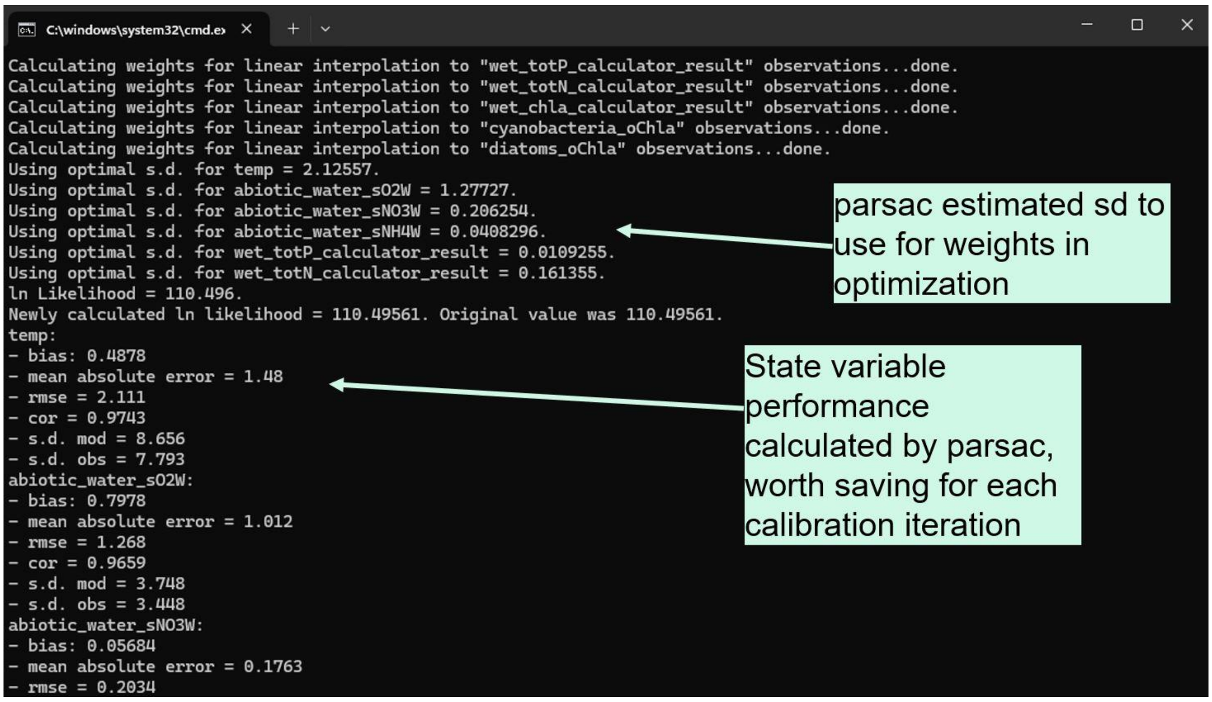

C:\windows\system32\cmd.e›
How many model simulations in the calibration did not complete
Parameter value for plotbest model execution (you can also find these in fabm.yaml in plotbest model folder)

```
D:\Models\lake_dashahe_qwet\parsac>C:\Users\Administrator\anaconda3\Scripts\parsac calibration plotbest D:/Models/lake_d
ashahe_qwet/parsac/wet_dashahe-step5.xml
Loading cached copy of D:\Models\lake_dashahe_qwet\parsac\tempv.obs...
Loading cached copy of D:\Models\lake_dashahe_qwet\parsac\do.obs...
Loading cached copy of D:\Models\lake_dashahe_qwet\parsac\no3.obs...
Loading cached copy of D:\Models\lake_dashahe_qwet\parsac\nh4.obs...
Loading cached copy of D:\Models\lake_dashahe_qwet\parsac\po4.obs...
Loading cached copy of D:\Models\lake_dashahe_qwet\parsac\tp.obs...
Loading cached copy of D:\Models\lake_dashahe_qwet\parsac\tn.obs...
Loading cached copy of D:\Models\lake_dashahe_qwet\parsac\chla.obs...
Loading cached copy of D:\Models\lake_dashahe_qwet\parsac\cyano_chla.obs...
Loading cached copy of D:\Models\lake_dashahe_qwet\parsac\diat_chla.obs...
Retrieving results...
Found 1029843 results, of which 0 were invalid.
1th best parameter set:
  gotm.yaml/surface/meteo/u10/scale_factor = 1.2217
  gotm.yaml/surface/meteo/v10/scale_factor = 1.0077
  gotm.yaml/surface/meteo/swr/scale_factor = 0.941884
  gotm.yaml/turbulence/turb_param/k_min = 3.28413e-06
  gotm.yaml/light_extinction/A/constant_value = 0.738787
  gotm.yaml/light_extinction/g1/constant_value = 0.398411
  gotm.yaml/light_extinction/g2/constant_value = 7.40694
  fabm.yaml/instances/abiotic_water/parameters/kDMinPOMW = 0.0629678
  fabm.yaml/instances/abiotic_water/parameters/kDMinDOMW = 0.0416152
  fabm.yaml/instances/abiotic_water/parameters/cThetaMinPOMW = 1.03687
  fabm.yaml/instances/abiotic_water/parameters/cThetaMinDOMW = 1.06267
  fabm.yaml/instances/abiotic_sediment/parameters/kDMinPOMS = 0.00998945
  fabm.yaml/instances/abiotic_sediment/parameters/kDMinDOMS = 0.00990978
  fabm.yaml/instances/abiotic_sediment/parameters/cThetaMinDOMS = 1.05223
  fabm.yaml/instances/abiotic_sediment/parameters/cThetaMinPOMS = 1.04707
```



```
  fabm.yaml/instances/cladocerans/parameters/cClearPrey2 = 1.54922
  fabm.yaml/instances/cladocerans/parameters/cClearPrey3 = 2.03721
  fabm.yaml/instances/cladocerans/parameters/cClearPrey4 = 2.48719
  fabm.yaml/instances/cladocerans/parameters/fGutOccPrey2 = 1.50008
  fabm.yaml/instances/cladocerans/parameters/fGutOccPrey4 = 4.8261
  fabm.yaml/instances/cladocerans/parameters/kDConsMaxZoo = 0.949981
  fabm.yaml/instances/cladocerans/parameters/cTmOptZoo = 28.7824
  fabm.yaml/instances/cladocerans/parameters/cSigTmZoo = 12.3742
  fabm.yaml/instances/cladocerans/parameters/fDAssZoo = 0.301235
  fabm.yaml/instances/cladocerans/parameters/fZooDOMW = 0.0515662
  fabm.yaml/instances/cladocerans/parameters/kMortZoo = 0.001
  fabm.yaml/instances/cladocerans/parameters/kDRespZoo = 0.132183
Original ln likelihood = 110.49561
Copying files for model setup to C:\Users\ADMINI~1\AppData\Local\Temp\gotmopt99csz5oo...
   skipping parsac because it is a directory
   skipping restart.nc because it is a NetCDF file
Evaluating fitness with parameter set [1.2217,1.0077,0.941884,3.28413e-06,0.738787,0.398411,7.40694,0.0629678,0.0416152,
1.03687,1.06267,0.00998945,0.00990978,1.05223,1.04707,0.000159337,2.89222,0.0034557,0.0150304,0.149681,1.51189,0.777001,
1.57426,0.917556,2.08223,1.079,0.000597438,0.000861677,0.670612,1.08654,0.972069,1.05063,1.11325,27.9326,0.00235275,0.00
26888,0.0222718,0.809709,1.95042,0.0184772,1.44527,1.03694,0.039098,1.51618,0.0199577,0.00947401,0.0282785,0.643669,0.64
0192,0.0297333,-0.5,0.0155308,0.0460541,0.0301086,2.1968,25.8694,10.7991,1,7.41933,14.1283,-0.5,0.5,0.247615,1.3709,5.17
096,30.4412,2,35.1659,0.773731,0.483398,23.2962,14.7578,1,11.4078,46.9837,1.56476,1.54922,2.03721,2.48719,1.50008,4.8261
,0.949981,28.7824,12.3742,0.301235,0.0515662,0.001,0.132183].
Starting model run...
 ------------------------------------------------------------------------
 GOTM started on 2024/10/04 at 03:50:28
 ------------------------------------------------------------------------
    init_gotm
 ------------------------------------------------------------------------
        Reading yaml configuration from: gotm.yaml
        configuring modules ....
    init_airsea_yaml
        done
    init_stim_yaml
        done.
```

parsac created the plotbest folder and executed the model from this folder



```
Calculating weights for linear interpolation to "wet_totP_calculator_result" observations...done.
Calculating weights for linear interpolation to "wet_totN_calculator_result" observations...done.
Calculating weights for linear interpolation to "wet_chla_calculator_result" observations...done.
Calculating weights for linear interpolation to "cyanobacteria_oChla" observations...done.
Calculating weights for linear interpolation to "diatoms_oChla" observations...done.
Using optimal s.d. for temp = 2.12557.
Using optimal s.d. for abiotic_water_sO2W = 1.27727.
Using optimal s.d. for abiotic_water_sNO3W = 0.206254.
Using optimal s.d. for abiotic_water_sNH4W = 0.0408296.
Using optimal s.d. for wet_totP_calculator_result = 0.0109255.
Using optimal s.d. for wet_totN_calculator_result = 0.161355.
ln Likelihood = 110.496.
Newly calculated ln likelihood = 110.49561. Original value was 110.49561.
temp:
- bias: 0.4878
- mean absolute error = 1.48
- rmse = 2.111
- cor = 0.9743
- s.d. mod = 8.656
- s.d. obs = 7.793
abiotic_water_sO2W:
- bias: 0.7978
- mean absolute error = 1.012
- rmse = 1.268
- cor = 0.9659
- s.d. mod = 3.748
- s.d. obs = 3.448
abiotic_water_sNO3W:
- bias: 0.05684
- mean absolute error = 0.1763
- rmse = 0.2034
```

parsac estimated sd to use for weights in optimization

State variable performance calculated by parsac, worth saving for each calibration iteration

Users can also execute second, third and so on best performing model parameterizations by specifying model rank: “parsac calibration plotbest xmlfile -r RANK”.

parsac plot

The plot function creates a “dotty”-plot that shows the relative ln likelihood against the model simulation’s parameter values in subplots for each parameter. So, each dot in a subplot represents a relative model performance (that is equal across subplots and parameters) against a parameter value within the parameter values ranges specified in the xml file.

The best performing parameter range is indicated by parsac with vertical dashed lines.

In large database files, it is not convenient to plot all model simulations with plot function. Users can specify a lower boundary for relative ln likelihood (always < 0) to select subset of model simulations with “-r RANGE”: “parsac calibration plot xmlfile -r RANGE”.

A calibration “signal” can be interpreted as a parameter range where many of the dots fall within close the relative ln likelihood of 0. The strength of this signal can vary, and it is up to the modeler to interpret if the parameter ranges should be expanded, narrowed, remain the same or shifted in the next calibration iteration.

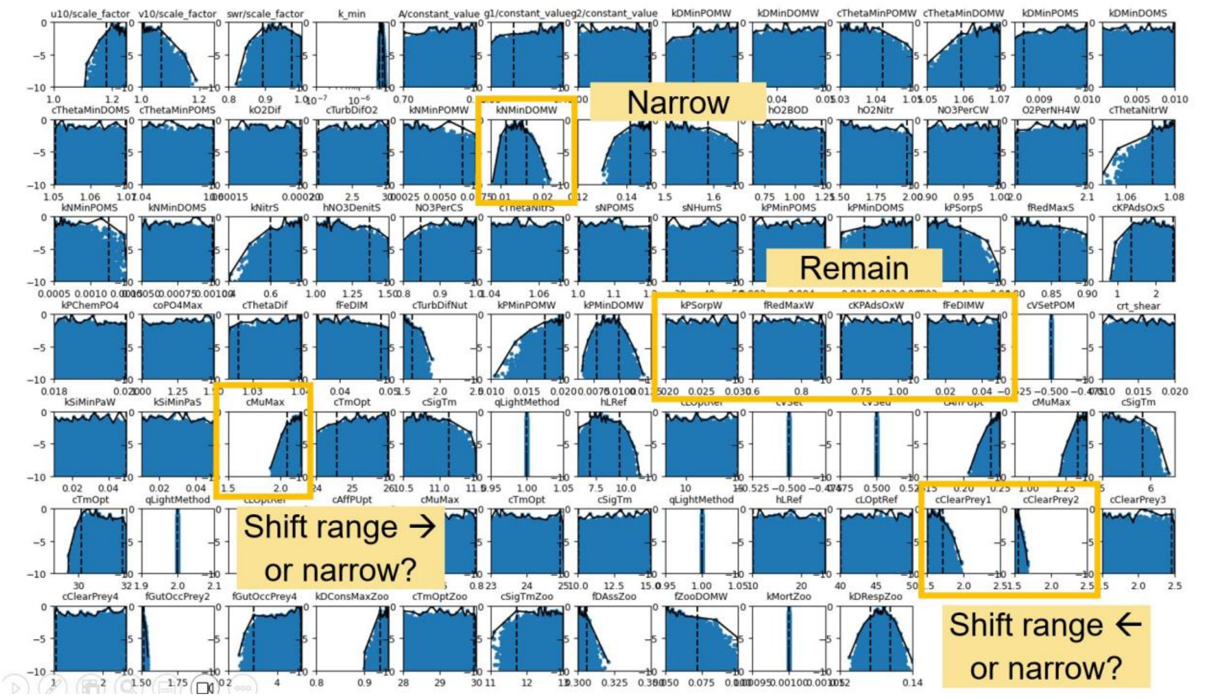
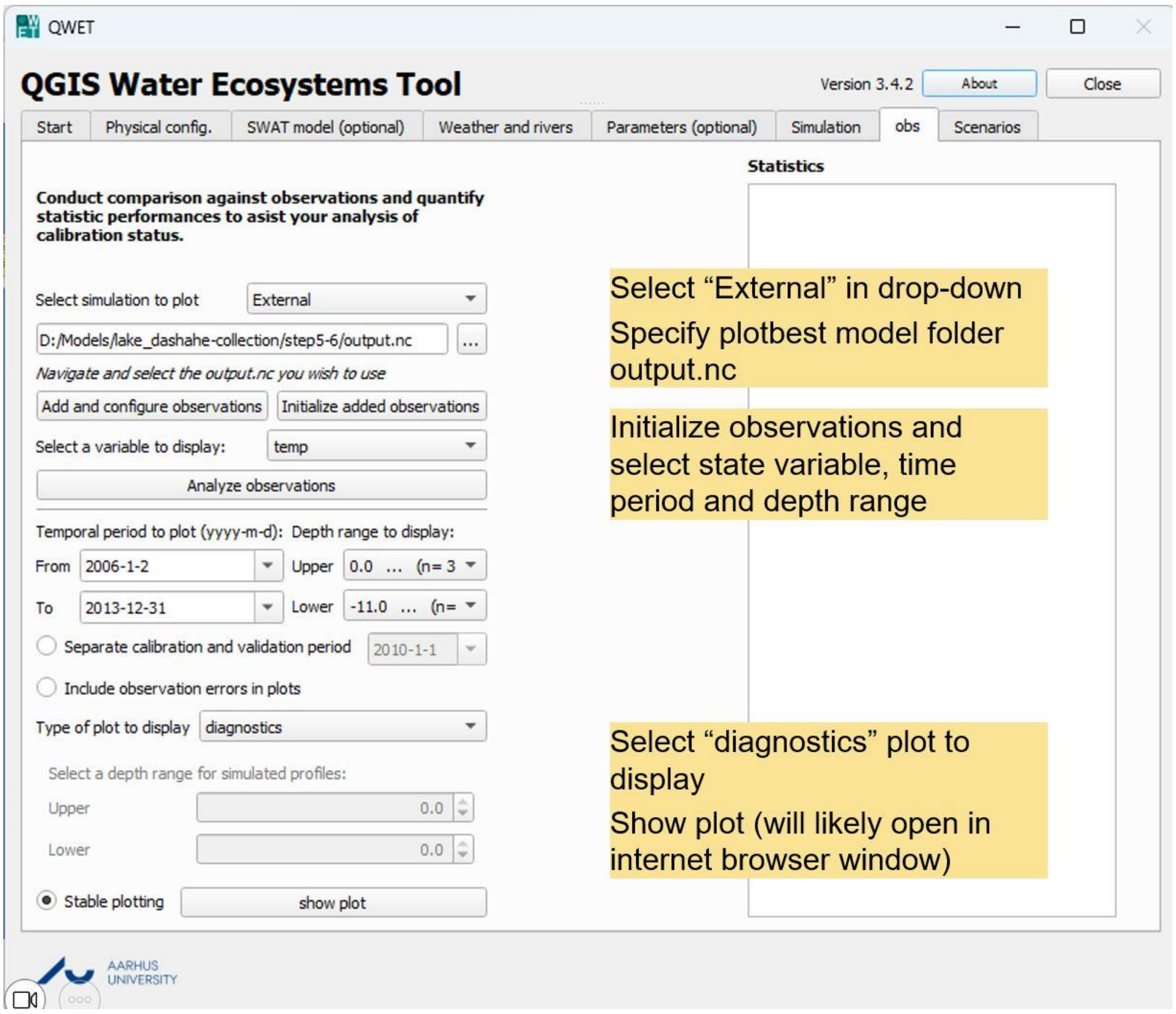


Figure 1. Dotty-plot from one calibration iteration for Shahe Reservoir with approx. 1 million model simulations in the database file. The range function has been used, so only the relative ln likelihood from 0 to 10 is plotted. Potential interpretations of the calibration signal of some parameters are included in orange.

The number of parameters with a calibration signal and the strength of the signal depends on the number of model simulations in the database file.

You should evaluate how the calibration signal and parameter range performances compare with the overall model behavior and simulated ecosystem dynamics. For this, we recommend to plot time series plots of state variables and observations, as well as other model evaluation plots, and evaluate if the calibration signal and parsac recommendations would increase the GOTM-WETS ability to reproduce the aquatic ecosystem dynamics and properties <u>for the right reasons</u> (i.e. parameter values are within technical and theoretical boundaries). parsac “recommendations” is model-agnostic and so all users should understand if shifts in parameter value ranges is within model theory and is realistic.

QWET plot diagnostics
QWET diagnostics plot (under the tab “obs” in QWET) is one such visualization tool that plots many common model evaluation visualizations.



QWET

QGIS Water Ecosystems Tool

Version 3.4.2 | About | Close

Start | Physical config. | SWAT model (optional) | Weather and rivers | Parameters (optional) | Simulation | obs | Scenarios

Conduct comparison against observations and quantify statistic performances to asist your analysis of calibration status.

Select simulation to plot: External

D:/Models/lake_dashahe-collection/step5-6/output.nc

Navigate and select the output.nc you wish to use

Add and configure observations | Initialize added observations

Select a variable to display: temp

Analyze observations

Temporal period to plot (yyyy-m-d): From 2006-1-2, To 2013-12-31

Depth range to display: Upper 0.0 ... (n= 3; Lower -11.0 ... (n=

Separate calibration and validation period: 2010-1-1

Include observation errors in plots

Type of plot to display: diagnostics

Select a depth range for simulated profiles: Upper 0.0; Lower 0.0

Stable plotting | show plot

Statistics

Select "External" in drop-down

Specify plotbest model folder output.nc

Initialize observations and select state variable, time period and depth range

Select "diagnostics" plot to display

Show plot (will likely open in internet browser window)

Calibration strategy and iterations

To facilitate the calibration process, the calibration procedure will typically be divided into a series of steps, where each steps adds additional parameters (and their ranges). Each step will target a selection of parameters that will have strong influence on the temporal and vertical dynamics of specific state variables. The aim is to achieve a strong signal in the objective function based on calibration of only a subset of model parameters, and thereby facilitate and enhance the ability to narrow the ranges of individual parameters following each calibration iteration. A somewhat generic approach would be to target (in this order): physical dynamics, e.g. temperature dynamics (and sediment resuspension if relevant, e.g. typically for shallow lakes) -> oxygen dynamics -> nutrient dynamics -> phytoplankton dynamics (and higher trophic levels). As WET is a complex ecosystem model, many parameters interact and so it is recommended to go through a calibration step approach and spend most of the time calibrating all selected parameters against all included observations.

You may find yourself needing to go back to a previous calibration iteration. For instance, if you learn something new about your case study and/or have constrained some parameter value ranges too much. Therefore, we recommend you save model folders from plotbest, model performance metrics, dotty-plots, xml files and optionally database files for each iteration, so you have a library and documentation of the calibration process.

Bat files (optional)

It is convenient to create a few bat files that will help you easily run ParSac. These bat files could be placed in the folder of you model set up. See the example xml file (which can be used and modified) for how to configure the calibration.

PARSAC CALIBRATION RUN
Example of content of bat for running "parsac calibration run" (in one line):
C:\Python3\Scripts\parsac calibration run
C:\Data\lake_models\lake_example\parsac\config_parsac.xml

PARSAC CALIBRATION PLOT
Example of content of bat for running "parsac calibration plot" (in one line):
C:\Python3\Scripts\parsac calibration plot
C:\Data\lake_models\lake_example\parsac\config_parsac.xml

PARSAC CALIBRATION PLOT -r
(Plot results with a zoom to higher/better likelihoods (dotty plots, i.e. parameter values vs objective function)
Example of content of bat for running "parsac calibration plot -r" (in one line):
C:\Python3\Scripts\parsac calibration plot -r -200
C:\Data\lake_models\lake_example\parsac\config_parsac.xml

PARSAC CALIBRATION PLOT -u (Plot results that will be updated on-the-fly)
Example of content of bat for running "parsac calibration plot -u" (in one line):
C:\Python3\Scripts\parsac calibration plot -u
C:\Data\lake_models\lake_example\parsac\config_parsac.xml

PARSAC CALIBRATION PLOTBEST (Plot results from the currently best performing simulation)
Example of content of bat for running "parsac calibration plot plotbest" (in one line):
C:\Python3\Scripts\parsac calibration plotbest
C:\Data\lake_models\lake_example\parsac\config_parsac.xml

PARSAC CALIBRATION PLOTBEST -r (Plot results from the currently a specific rank of the best performing simulations)
Example of content of bat for running "parsac calibration plot plotbest" (in one line):
C:\Python3\Scripts\parsac calibration plotbest -r 2
C:\Data\lake_models\lake_example\parsac\config_parsac.xml