



Tobias K Andersen, Xiangzhen Kong

# Parallel sensitivity and auto-calibration tool (parsac) tutorial



# Parsac

## Parallel sensitivity and auto-calibration Tool

parsac is a Python package for sensitivity analysis and auto-calibration in parallel. It is designed for analysis of models that take significant time to run. It works with models that are run by calling one binary, that use text-based configuration files based on yaml or Fortran namelists, and that write their output to netcdf.

Developed by Jorn Bruggeman and Karsten Bolding.

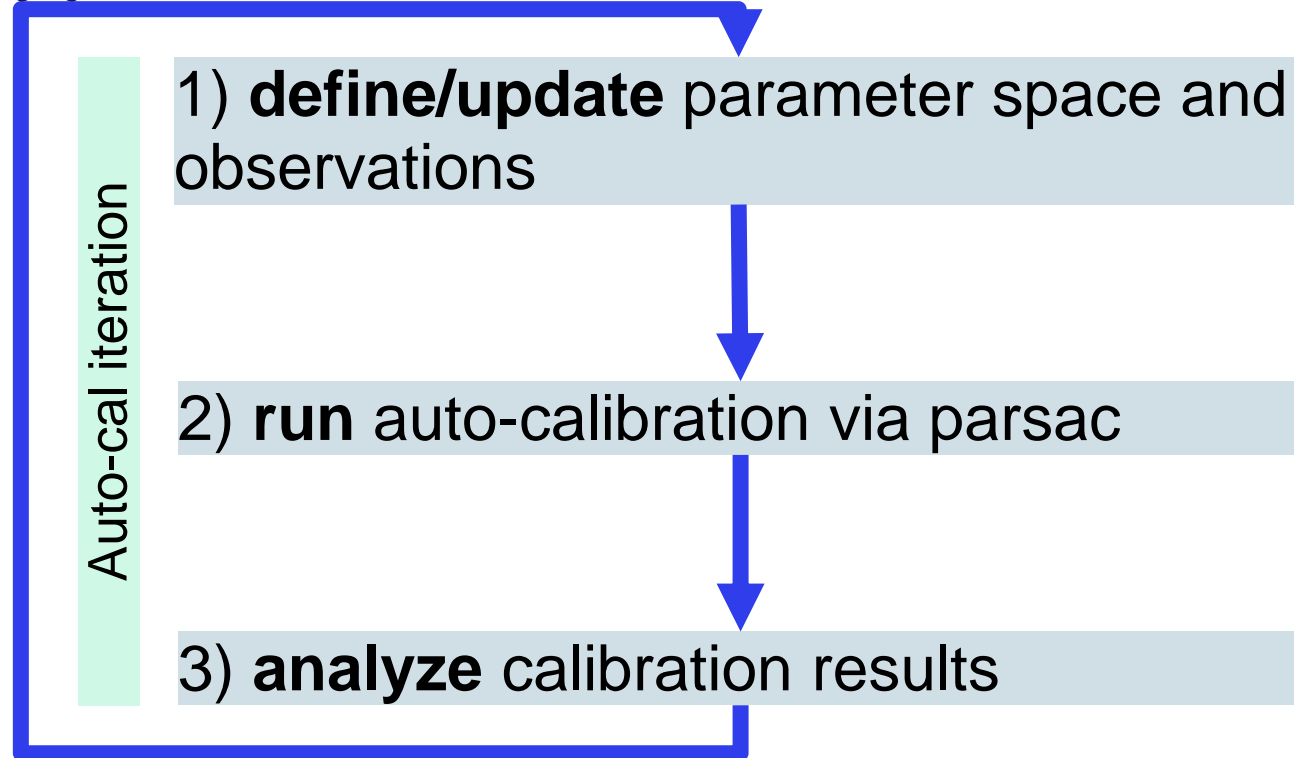
Source code: <https://github.com/BoldingBruggeman/parsac>

Installation: `pip install parsac --user` (in your Python console window)

The background is a dark blue illustration of a pond. It features several tall, thin reeds or grasses. Four fish, resembling striped bass or similar species, are depicted in various positions: one at the top left, one in the middle left, one in the middle right, and one at the bottom right. The fish have white stripes on their dark bodies. The overall style is minimalist and graphic.

# **Auto-calibration**

# Brief intro to auto-cal with parsac



**Danmarks Tekniske Universitet**

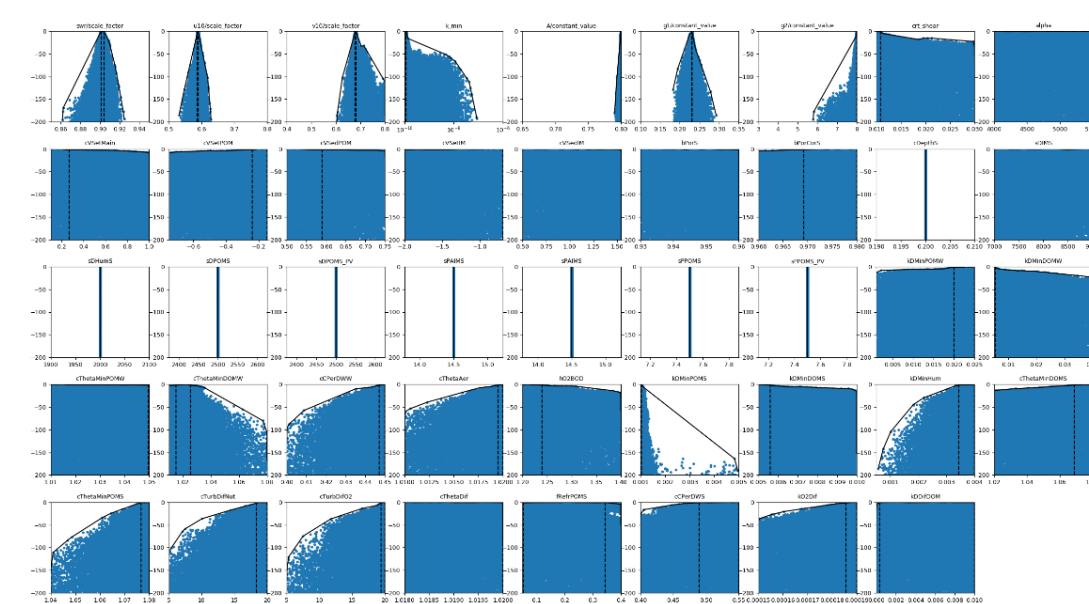
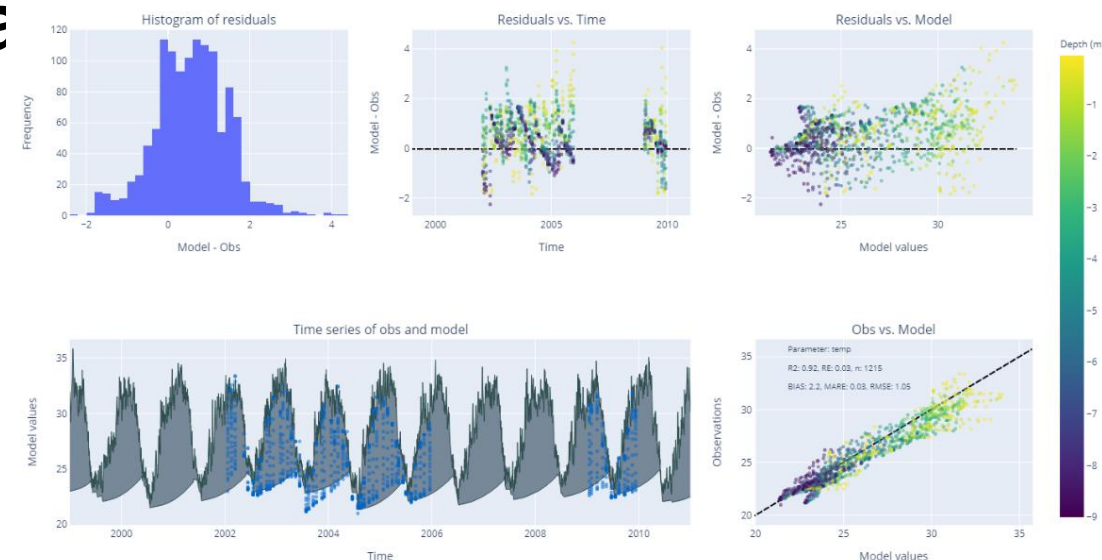
# Brief intro to auto-cal with parsac

Auto-cal iteration

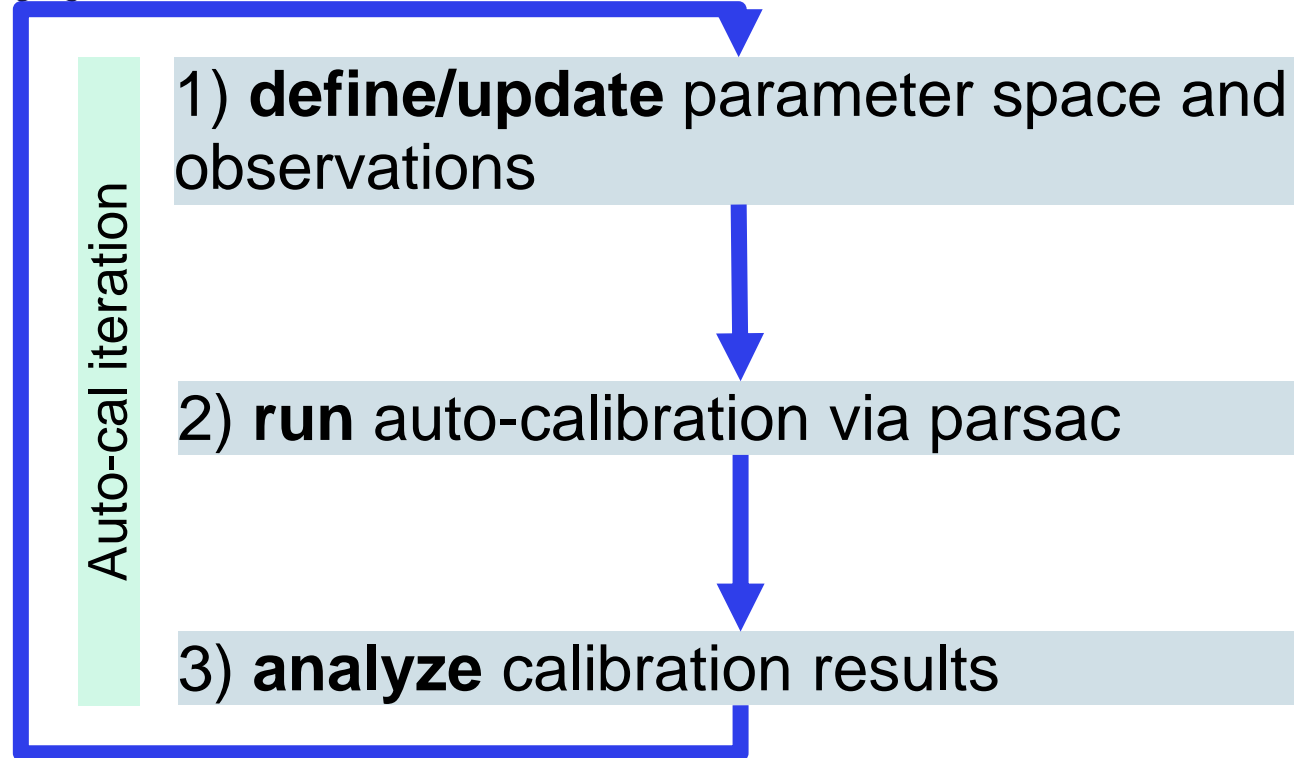
1) **define/update** parameter space and observations

2) **run** auto-calibration via parsac

3) **analyze** calibration results



# Brief intro to auto-cal with parsac





wet\_dashahe-step1.xml

```

1 <config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="../client/config.xsd">
2   <transports>
3     <transport type="sqlite" path="LakeDashahe_step5.db"/>
4   </transports>
5   <executable path="../gotm_release.exe"/>
6   <setup path="../"/>
7   <parameters>
8     <!-- all steps include plus/minus 40% of the default value or based on user experience -->
9     <!-- be cautious of sediment initialization values -->
10    <!-- STEP 1A: Calibration of physical dynamics - TEMPERATURE -->
11    <!-- BIOSHADE IN GOTM.YAML = FALSE -->
12    <!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
13    <!-- Physical parameters -->
14    <!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
15    <!-- wind speed in West-East direction @ 10 m -->
16    <parameter file="gotm.yaml" variable="surface/meteo/u10/scale_factor" minimum="0.6" maximum="1.4"/>
17    <!-- wind speed in North-South direction @ 10 m -->
18    <parameter file="gotm.yaml" variable="surface/meteo/v10/scale_factor" minimum="0.6" maximum="1.4"/>
19    <!-- shortwave radiation -->
20    <parameter file="gotm.yaml" variable="surface/meteo/swr/scale_factor" minimum="0.6" maximum="1.4"/>
21    <!-- cloud cover, can be calibrated -->
22    <!-- <parameter file="gotm.yaml" variable="surface/meteo/cloud/scale_factor" minimum="1.0" maximum="1.0"/> -->
23    <!-- minimum turbulent kinetic energy [m^2/s^2; min=0.0; default=1.00000000E-10] -->
24    <parameter file="gotm.yaml" variable="turbulence/turb_param/k_min" minimum="1e-9" maximum="5e-7" logscale="True"/>
25    <!-- non-visible fraction of shortwave radiation [fraction; min=0.0; max=1.0; default=0.7] -->
26    <parameter file="gotm.yaml" variable="light_extinction/A/constant_value" minimum="0.65" maximum="0.75"/>
27    <!-- e-folding depth of non-visible shortwave radiation [m; min=0.0; default=0.4] -->
28    <parameter file="gotm.yaml" variable="light_extinction/g1/constant_value" minimum="0.30" maximum="0.5"/>
29    <!-- e-folding depth of visible shortwave radiation [m; min=0.0; default=8.0] -->
30    <parameter file="gotm.yaml" variable="light_extinction/g2/constant_value" minimum="6.0" maximum="10.0"/>
31  </parameters>
32  <observations>
33    <variable source="tempv.obs" modelpath="output.nc" modelvariable="temp" spinupyears = "5"/>
34    <!-- <variable source="do.obs" modelpath="output.nc" modelvariable="abiotic_water_sO2W" spinupyears = "5"/> -->
35    <!-- <variable source="no3.obs" modelpath="output.nc" modelvariable="abiotic_water_sNO3W" spinupyears = "5"/> -->
36    <!-- <variable source="nh4.obs" modelpath="output.nc" modelvariable="abiotic_water_sNH4W" spinupyears = "5"/> -->
37    <!-- <variable source="po4.obs" modelpath="output.nc" modelvariable="abiotic_water_sPO4W" spinupyears = "5"/> -->
38    <!-- <variable source="tp.obs" modelpath="output.nc" modelvariable="wet_totP_calculator_result" spinupyears = "5"/> -->
39    <!-- <variable source="tn.obs" modelpath="output.nc" modelvariable="wet_totN_calculator_result" spinupyears = "5"/> -->
40    <!-- <variable source="chla.obs" modelpath="output.nc" modelvariable="wet_chla_calculator_result" spinupyears = "5"/> -->
41    <!-- <variable source="cyano_chla.obs" modelpath="output.nc" modelvariable="cyanobacteria_oChla" spinupyears = "5"/> -->
42    <!-- <variable source="diat_chla.obs" modelpath="output.nc" modelvariable="diatoms_oChla" spinupyears = "5"/> -->
43  </observations>
44  <extra_outputs>
45    <statistic name="correlation" expression="corrcoef(x,y) [0,1]"/>
46    <statistic name="rmse" expression="sqrt(mean((y-x)**2))"/>
47    <statistic name="R2" expression="corrcoef(x,y) [0,1]**2"/>
48    <statistic name="pbias" expression="sum((y-x)*100)/sum(x)"/>
49  </extra_outputs>
50 </config>
51
52

```

Specify database file name

Specify GOTM-WET  
executableSpecify GOTM-WET  
parameters and their  
rangesparsac only includes .obs  
files under <observation>Optional: include  
calculated performance  
metrics for each obs in  
database file



```
D:\Models\lake_dashaha_qwet\parsac>C:\Users\Administrator\anaconda3\Scripts\parsac calibration plotbest D:/Models/lake_d
ashahe_qwet/parsac/wet_dashaha-step5.xml
Loading cached copy of D:\Models\lake_dashaha_qwet\parsac\tempv.obs...
Loading cached copy of D:\Models\lake_dashaha_qwet\parsac\do.obs...
Loading cached copy of D:\Models\lake_dashaha_qwet\parsac\no3.obs...
Loading cached copy of D:\Models\lake_dashaha_qwet\parsac\nh4.obs...
Loading cached copy of D:\Models\lake_dashaha_qwet\parsac\po4.obs...
Loading cached copy of D:\Models\lake_dashaha_qwet\parsac\tp.obs...
Loading cached copy of D:\Models\lake_dashaha_qwet\parsac\tn.obs...
Loading cached copy of D:\Models\lake_dashaha_qwet\parsac\chl_a.obs...
Loading cached copy of D:\Models\lake_dashaha_qwet\parsac\cymo_chl_a.obs...
Loading cached copy of D:\Models\lake_dashaha_qwet\parsac\diat_chl_a.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/abiotic_sediment/parameters/kO2Dif = 0.000159337
fabm.yaml/instances/abiotic_sediment/parameters/cTurbDifO2 = 2.89222
fabm.yaml/instances/abiotic_water/parameters/kNMinPOMW = 0.0034557
fabm.yaml/instances/abiotic_water/parameters/kNMinDOMW = 0.0150304
fabm.yaml/instances/abiotic_water/parameters/kNitrW = 0.149681
fabm.yaml/instances/abiotic_water/parameters/hNO3DenitW = 1.51189
fabm.yaml/instances/abiotic_water/parameters/hO2BOD = 0.777001
fabm.yaml/instances/abiotic_water/parameters/hO2Nitr = 1.57426
fabm.yaml/instances/abiotic_water/parameters/NO3PerCW = 0.917556
```

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)

```

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.
init_observations_yaml
configure_streams_yaml
  done
init_turbulence_yaml
  done.

```

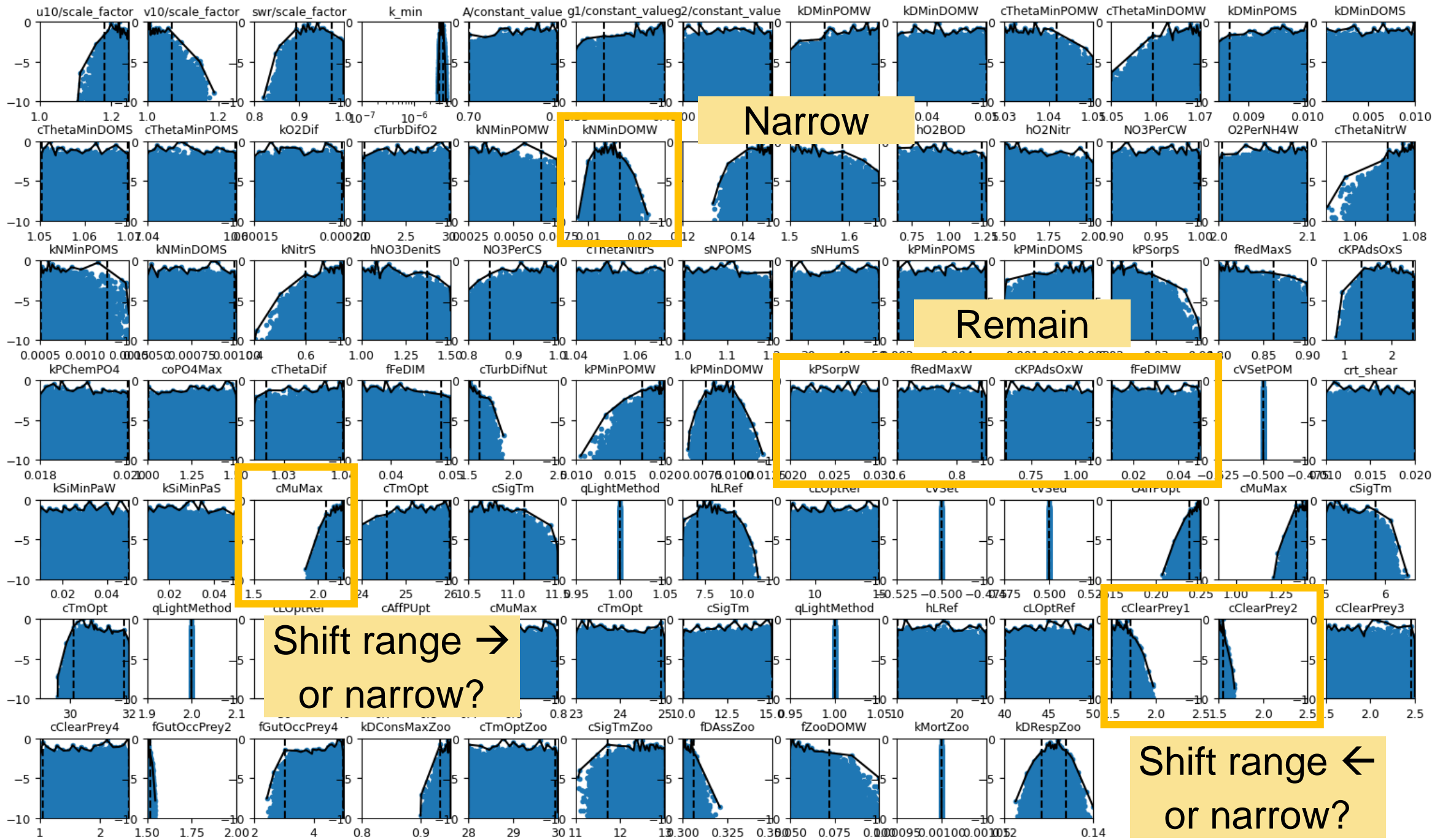
parsac created the  
plotbest folder and  
executed the model  
from this folder

```


C:\windows\system32\cmd.exe X + v
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
- cor = 0.8241
- s.d. mod = 0.3133
- s.d. obs = 0.3397
abiotic_water_sNH4W:
- bias: 0.009641
- mean absolute error = 0.02944
- rmse = 0.04026
- cor = 0.229
- s.d. mod = 0.0199
  
```

parsac estimated sd to use for weights in optimization

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






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

## Statistics

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

Select simulation to plot

External

D:/Models/lake\_dashahc-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): Depth range to display:

From

2006-1-2

Upper

0.0 ... (n= 3)

To

2013-12-31

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

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)



The background of the slide features a stylized illustration of three fish, likely striped bass, swimming in a pond. The fish are depicted in a light blue color against a dark blue background. They are surrounded by tall, thin reeds or grasses that also have a light blue outline. The overall style is minimalist and graphic.

# **Sensitivity analysis**

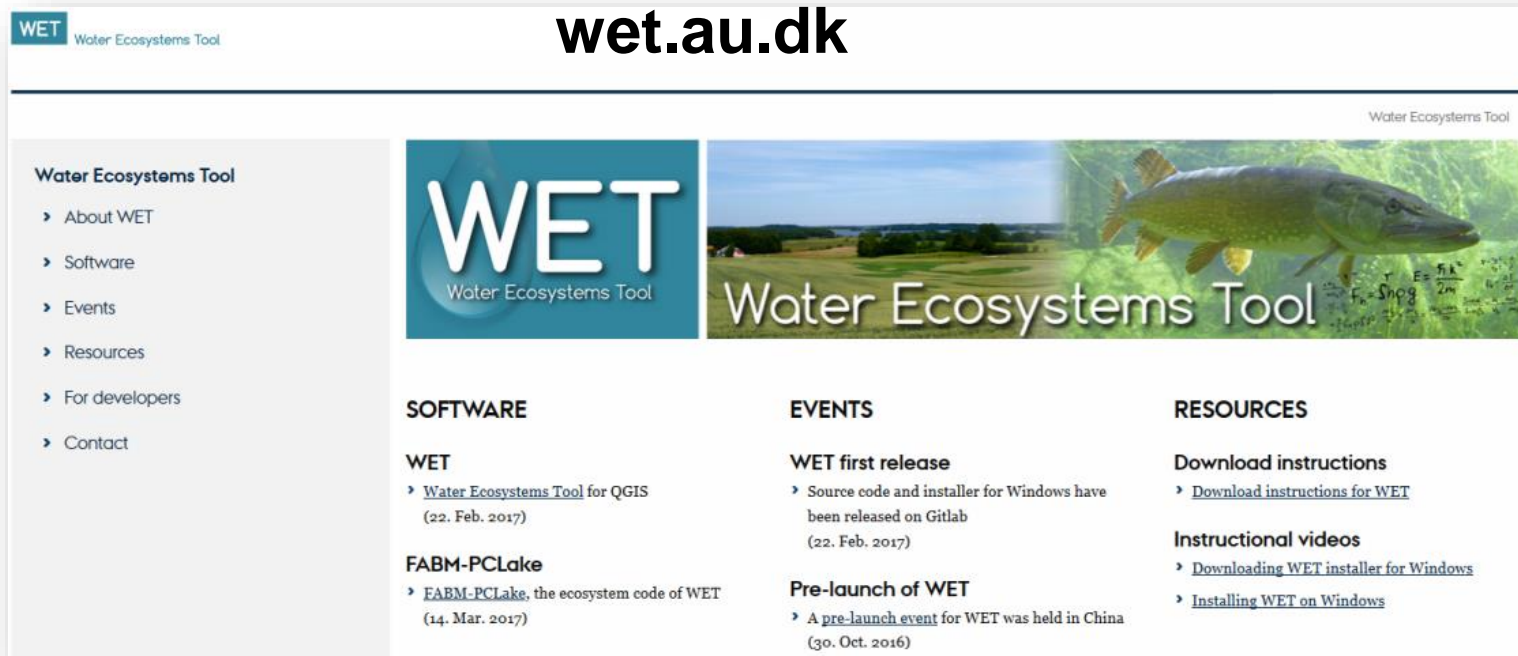


[tkan@aqu.dtu.dk](mailto:tkan@aqu.dtu.dk)

or you can write to me  
on wechat 😊

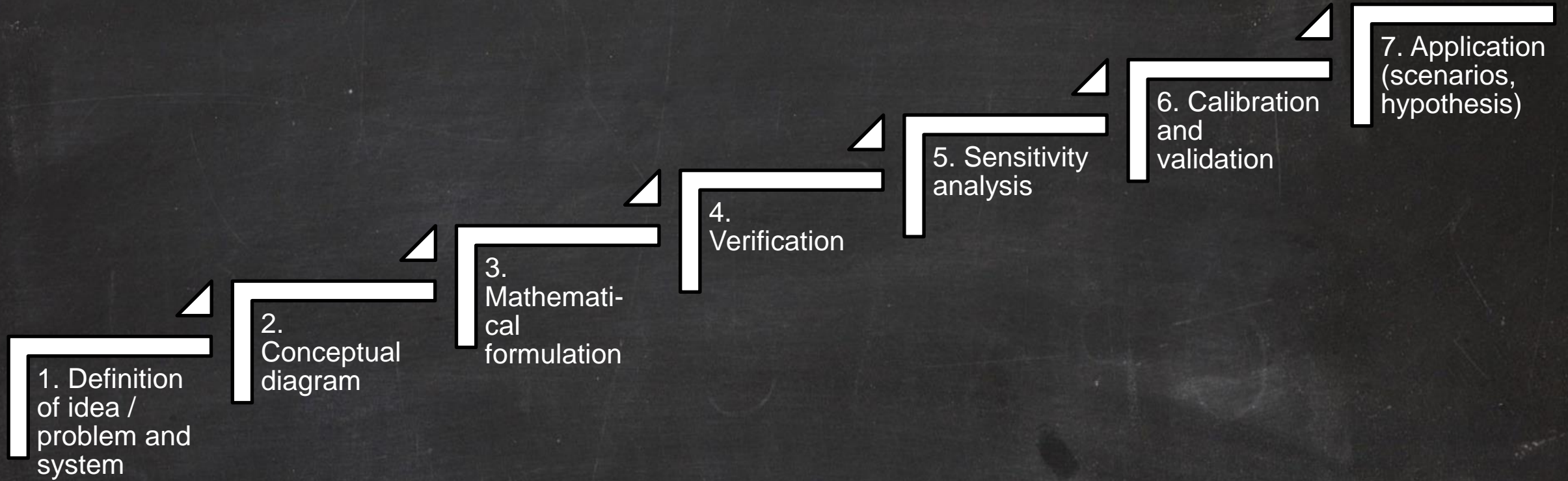
Currently, WET website at Aarhus University is not maintained. WET team is working on a new solution.

We let you know via wechat when new website with updated material is ready.





# The modelling process





# parsac and SALib

For sensitivity analysis, parsac relies on the Python package Sensitivity Analysis Library in Python (SALib) published by [Herman & Usher \(2017\)](#) and [Iwanaga, Usher and Herman \(2022\)](#).



**SALib: An open-source Python library for Sensitivity Analysis**  
Jon Herman<sup>1</sup> and Will Usher<sup>2</sup>  
<sup>1</sup> University of California, Davis <sup>2</sup> University of Oxford

DOI: 10.21105/joss.00097

Software  
• Review of  
• Revision of  
• Archive of

License  
Authors of JOSS papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC-BY).

## Summary

SALib contains Python implementations of commonly used global sensitivity analysis methods, including Sobol, Saltelli, Morris, and FAST. SALib is designed to be easy to use and to integrate with other Python libraries. It is a good starting point for those who want to perform sensitivity analysis. SALib is a good starting point for those who want to perform sensitivity analysis. SALib is a good starting point for those who want to perform sensitivity analysis.

## References

- Borgonovo, E. 2007. "A new sensitivity importance measure." *Reliability Engineering and System Safety* 92 (1): 71-144. doi:10.1016/j.ssci.2006.03.015.
- Chapman, J., Justin Charlton, and Andrew Saltelli. 2007. "An effective sensitivity analysis for complex systems of input-output." *Environmental Modelling* 21 (Supplement 1): 139-144. doi:10.1016/j.envmod.2006.03.015.
- Cukier, R. L., C. M. Fortin, R. C. Stein, A. G. Pritchard, and J. H. Saltelli. 1993. "Study of the sensitivity of coupled reaction systems to uncertainties in rate constants. I. Theory." *Journal of Chemical Physics* 98 (10): 2973-8. doi:10.1063/1.462171.
- Fiori, Massimo D. 1981. "Statistical Sampling Plans for Preliminary Computational Experiments." *Technometrics* 23 (4): 319-327.
- Plischke, Elmar, Edoardo Borgonovo, and Christa L. Smith. 2013. "Global sensitivity analysis from gene data." *European Journal of Operational Research* 228 (3): 39-50. doi:10.1016/j.ejor.2013.03.047.
- Saltelli, A., S. Tarantola, and K. P. S. Chan. 1999. "A Quantitative Model Independent Method for Global Sensitivity Analysis of Model Output." *Technometrics* 41 (1): 39-56. doi:10.1080/00141899908839004.
- Saltelli, A. 2000. "Making best use of model evaluation to compare sensitivity analysis." *Computational Physics Communications* 145 (2): 281-87. doi:10.1016/S0010-0194(00)00055-5.

## Toward SALib 2.0: Advancing the accessibility and interpretability of global sensitivity analyses

"Tackling legacy", will usher<sup>1</sup>, and jon herman<sup>2</sup>  
<sup>1</sup> Institute for Future, The Australian National University, Australia  
<sup>2</sup> Division of Energy Systems, School of Industrial Engineering and Management, City College of San Francisco, San Francisco, CA, USA  
<sup>3</sup> Department of Civil and Environmental Engineering, University of California, Davis, CA, USA

## Abstract

Sensitivity analysis is now considered a standard practice in environmental modeling. Sensitivity analysis is now considered a standard practice in environmental modeling. Sensitivity analysis is now considered a standard practice in environmental modeling. Sensitivity analysis is now considered a standard practice in environmental modeling. Sensitivity analysis is now considered a standard practice in environmental modeling.

## Keywords

sensitivity analysis; community of practice; software accessibility  
The SALib project is hosted on GitHub (<https://github.com/SALib/SALib>) and is made available under the MIT license. Data, code, and figures for further analysis conducted for this publication are found in <https://doi.org/10.21105/joss.00097>.

## 1. Introduction

The explosive growth and availability of computational power has increased the complexity of environmental modeling analyses. In turn, the complexity of these analyses, as well as the data produced and collected, has increased. This has led to a need for more sophisticated methods for analyzing the results of these analyses. This has led to a need for more sophisticated methods for analyzing the results of these analyses.

after this article has been accepted for publication, the publisher will send the author a PDF proof of the final version of the article, which will include any changes made by the publisher.

Copyright © 2017, by the author(s). All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or by any information storage and retrieval system, without permission in writing from the publisher.

This work is licensed under a Creative Commons Attribution 4.0 International License.

For more information on this journal, please visit the journal website at <https://joss.readthedocs.io>.

For more information on this journal, please visit the journal website at <https://joss.readthedocs.io>.

For more information on this journal, please visit the journal website at <https://joss.readthedocs.io>.

For more information on this journal, please visit the journal website at <https://joss.readthedocs.io>.

For more information on this journal, please visit the journal website at <https://joss.readthedocs.io>.

For more information on this journal, please visit the journal website at <https://joss.readthedocs.io>.

For more information on this journal, please visit the journal website at <https://joss.readthedocs.io>.

For more information on this journal, please visit the journal website at <https://joss.readthedocs.io>.

For more information on this journal, please visit the journal website at <https://joss.readthedocs.io>.

← → ↺ https://salib.readthedocs.io/en/latest/index.html 80% ☆

Import bookmarks... Getting Started Getting Started Most Visited Getting Started Example gallery — sea... WFD model evaluatio... Magic

SALIB

Getting started Basics SALib Interface Advanced Wrappers More ▾

Search Ctr1 + K Choose version ▾

## SALib - Sensitivity Analysis Library in Python

DOI: 10.5281/zenodo.160164 DOI: 10.18174/sesmo.18155 JOSS: 10.21105/joss.00097

Python implementations of commonly used sensitivity analysis methods, including Sobol, Morris, and FAST methods. Useful in systems modeling to calculate the effects of model inputs or exogenous factors on outputs of interest.

## Supported Methods

- Sobol Sensitivity Analysis ([Sobol 2001](#), [Saltelli 2002](#), [Saltelli et al. 2010](#))
- Method of Morris, including groups and optimal trajectories ([Morris 1991](#), [Campolongo et al. 2007](#))
- Fourier Amplitude Sensitivity Test (FAST) ([Cukier et al. 1973](#), [Saltelli et al. 1999](#))
- Random Balance Designs - Fourier Amplitude Sensitivity Test (RBD-FAST) ([Tarantola et al. 2006](#), [Elmar Plischke 2010](#), [Tisost et al. 2012](#))
- Delta Moment-Independent Measure ([Borgonovo 2007](#), [Plischke et al. 2013](#))
- Derivative-based Global Sensitivity Measure (DGSM) ([Sobol and Kucherenko 2009](#))
- Fractional Factorial Sensitivity Analysis ([Saltelli et al. 2008](#))
- High Dimensional Model Representation ([Li et al. 2010](#))
- PAWN ([Pianosi and Wagener 2018](#), [Pianosi and Wagener 2015](#))
- Regional Sensitivity Analysis (based on [Hornberger and Spear, 1981](#), [Saltelli et al. 2008](#), [Pianosi et al., 2016](#))



# SALib: Match sampling with SA method

Table 1. parsac supported SALib methods and their corresponding sampling schemes with their parsac names.

Sensitivity method	parsac name	Sampling	parsac name
Fourier Amplitude Sensitivity Test (FAST)	fast		fast
Random Balance Designs - Fourier Amplitude Sensitivity Test	rbd_fast	Latin hypercube sampling	latin
Method of Morris	morris		morris
Sobol Sensitivity Analysis	sobol	Saltelli's sampling	saltelli
Delta Moment-Independent Measure	delta	Latin hypercube sampling	latin
Derivative-based Global Sensitivity Measure	dgsm		
Fractional Factorial Sensitivity Analysis	ff		ff

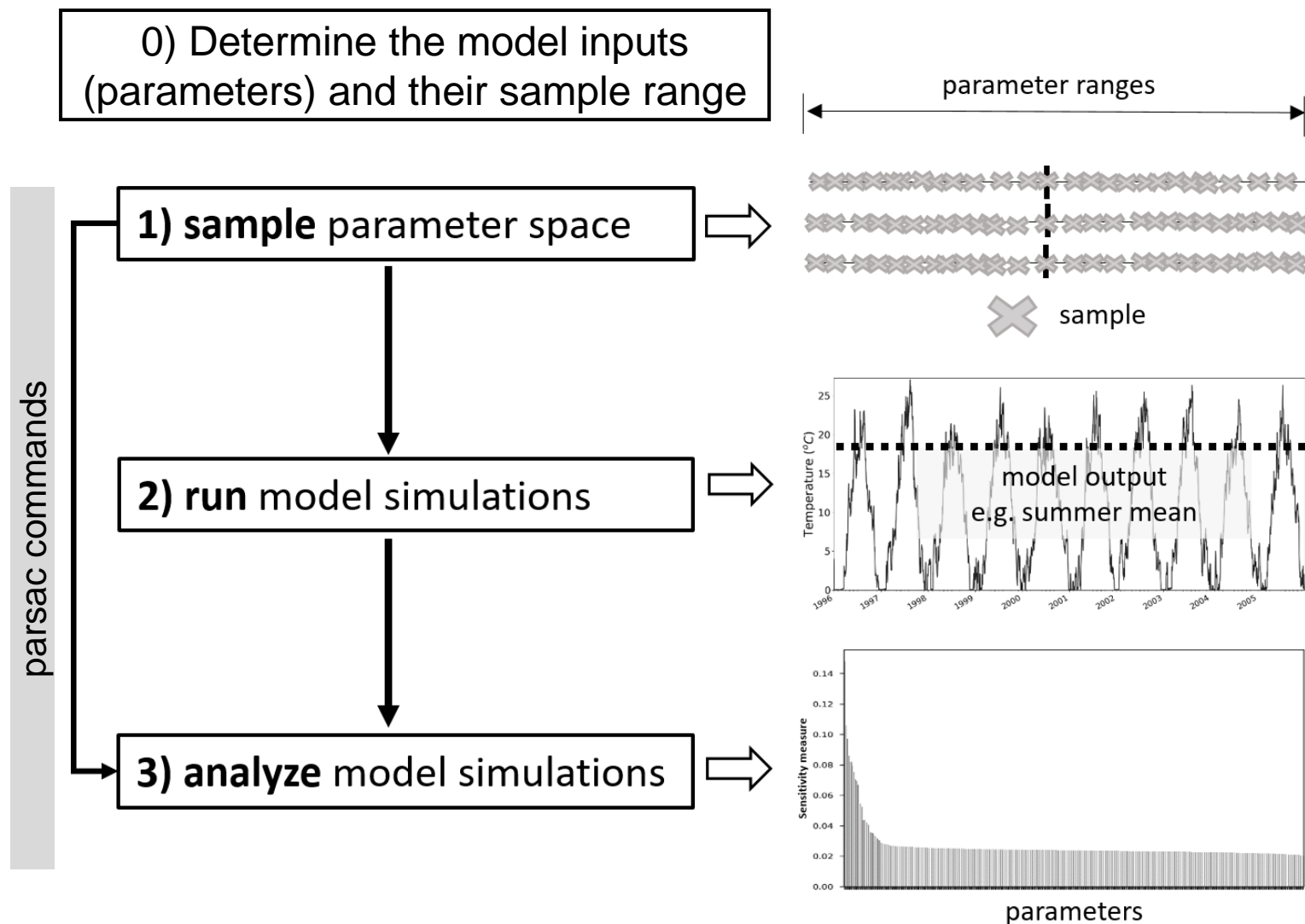
# Brief intro to sensitivity analysis with parsac

parsac relies on the Python package Sensitivity Analysis Library for Python (SALib) for SA methods.

Full SA procedure documented in Andersen et al. (2021)



## General parsac sensitivity analysis



# DTU Working parsac folder for SA

File Explorer window showing the 'parsac' folder. The address bar indicates the path: Tobias Kuhlmann Andersen > Model\_setups > lake\_shahe\_wetworkshop > parsac. The left sidebar shows the 'Model\_setups' folder selected under 'Quick access'. The main pane displays a list of files:

Name	Date modified	Type	Size
1-parsac_sample-latin-temp-n70.bat	15-10-2024 07:44	Windows Batch File	1 KB
2-parsac_run-temp-n70.bat	15-10-2024 07:44	Windows Batch File	1 KB
3-parsac_analyze-Delta_temp.bat	15-10-2024 07:44	Windows Batch File	1 KB
3-parsac_analyze-Delta_temp-print_to_console.bat	15-10-2024 07:44	Windows Batch File	1 KB
3-parsac_analyze-Delta_temp-print_to_pkl.bat	15-10-2024 07:44	Windows Batch File	1 KB
wet_dashahe-step1-SA.xml	14-10-2024 22:21	XML File	5 KB

6 items



bryrup\_sa\_epi\_summer\_tp\_25.xml

```

1 <config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.w3.org/2001/XMLSchema-instance" client/confi
2 <transports>
3   <transport type="sqlite" path="Arreskov_SA_all.db"/>
4 </transports>
5 <executable path="../gotm.exe"/>
6 <setup path="../"/>
7 <parameters>
8   <parameter dummy="True" maximum="1" minimum="0"/>
9   <parameter file="gotm.yaml" variable="surface/fluxes/heat/scale_factor" minimum="0.75" maximum="1.25"/>
10  <parameter file="gotm.yaml" variable="surface/meteo/swr/scale_factor" minimum="0.75" maximum="1.25"/>
11  <parameter file="gotm.yaml" variable="surface/meteo/u10/scale_factor" minimum="0.75" maximum="1.25"/>
12  <parameter file="gotm.yaml" variable="surface/meteo/v10/scale_factor" minimum="0.75" maximum="1.25"/>
13  <parameter file="gotm.yaml" variable="turbulence/turb_param/k_min" minimum="1E-10" maximum="1-E6" logscale="True"/>
14  <parameter file="gotm.yaml" variable="light_extinction/A/constant_value" minimum="0.45" maximum="0.75"/>
15  <parameter file="gotm.yaml" variable="light_extinction/g1/constant_value" minimum="0.3" maximum="0.5"/>
16  <parameter file="gotm.yaml" variable="light_extinction/g2/constant_value" minimum="6.0" maximum="10.0"/>
17  <parameter file="fabm.yaml" maximum="2.0" minimum="1.2" variable="instances/abiotic_sediment/parameters/cKPAdsOx"/>
18  <parameter file="fabm.yaml" maximum="18.47" minimum="11.082" variable="instances/abiotic_sediment/parameters/cTurbDifNut"/>
19  <parameter file="fabm.yaml" maximum="0.125" minimum="0.075" variable="instances/abiotic_sediment/parameters/cDepths"/>
20  <parameter file="fabm.yaml" maximum="0.92125" minimum="0.55275" variable="instances/abiotic_sediment/parameters/bPorCorS"/>
21  <parameter file="fabm.yaml" maximum="0.95" minimum="0.64425" variable="instances/abiotic_sediment/parameters/fRedMax"/>
22  <parameter file="fabm.yaml" maximum="0.625" minimum="0.375" variable="instances/abiotic_sediment/parameters/fDepthDifs"/>
23  <parameter file="fabm.yaml" maximum="0.0125" minimum="0.0075" variable="instances/abiotic_sediment/parameters/fALDIM"/>
24  <parameter file="fabm.yaml" maximum="0.20125" minimum="0.12075" variable="instances/abiotic_sediment/parameters/cRelPAdsAl"/>
25  <parameter file="fabm.yaml" maximum="0.00375" minimum="0.00225" variable="instances/abiotic_sediment/parameters/kDMinPOMS"/>
26 </parameters>
27 <observations>
28   <!-- linkage between observation files and model output for objective function in calibration -->
29   <!-- Step 1 -->
30   <!-- <variable source="./DO.obs" modelpath="output.nc" modelvariable="abiotic_water_sO2W" spinupyears="3"/> -->
31   <!-- <variable source="./CHLA.obs" modelpath="output.nc" modelvariable="pclake_chla_calculator_result" spinupyears="3"/> -->
32 </observations>
33 <targets>
34   <!-- MEAN -->
35   <target expression="filter_by_time(wet_totP_calculator_result[:, :], months=(1,2,3,4,5,6,7,8,9,10,11,12)).mean()" path="output.nc"/>
36   <target expression="filter_by_time(wet_totP_calculator_result[:, :], months=(1,2,12)).mean()" path="output.nc"/>
37   <target expression="filter_by_time(wet_totP_calculator_result[:, :], months=(3,4,5)).mean()" path="output.nc"/>
38   <target expression="filter_by_time(wet_totP_calculator_result[:, :], months=(6,7,8)).mean()" path="output.nc"/>
39   <target expression="filter_by_time(wet_totP_calculator_result[:, :], months=(9,10,11)).mean()" path="output.nc"/>
40 </targets>
41 </config>
42

```

Specify database file name  
(not used)

Specify GOTM-WET  
executable

Specify GOTM-WET  
parameters and their  
ranges

parsac sensitivity does not  
use observations

Specify GOTM-WET  
output, netcdf, time  
period and descriptive  
metric (mean, etc)

## What is the aim of the sensitivity analysis?

Some ideas to determine parameter ranges:

- % ranges from default values
- % ranges from calibrated values
- Min and max based on literature



## parsac SA commands

The corresponding parsac command lines, arguments and options are:

### 1. “parsac sensitivity sample xmlfile info {fast,latin,morris,saltelli,ff} samplesize”

xmlfile	XML formatted configuration file (xml file)
info	Path to output of the "sample" step (pickle file, .pkl)
samplesize	Number of samples generated by sampling scheme

### 2. “parsac sensitivity run info”

info	Path to output of the "sample" step (pickle file, .pkl)
------	---

### 3. “parsac sensitivity analyze [-h] [--print\_to\_console] [--select SELECT SELECT] [--pickle PICKLE] info {fast,rbd\_fast,morris,sobol,delta,dgsm,ff} --num\_resamples RESAMPLES”

info	Path to output of the "sample" step (pickle file, .pkl)
--print_to_console	Print results directly to console
--select SELECT SELECT	This requires two values: N OUTPUTXML. Selects the N most sensitive parameters for a calibration run and save it to OUTPUTXML
--pickle PICKLE	Path of pickle file to write with analysis results
--num_resamples	The number of resamples when computing confidence intervals

**Andersen et al (2020) used**

**Screening (all some GOTM + all WET parameters: approx. 360)**

100 samples per parameter

**In-depth SA (most sensitive parameters, 20-50)**

250 samples per parameter

Most important, check confidence intervals in SA results.

# Parsac SA command for delta analysis

## 1. **parsac sensitivity sample xmlfile info latin samplesize**

xmlfile = wet\_dashahe-step1-SA.xml

info = "C:\folder\_to\_\parsac\delta\_temp\_n30.pkl"

samplesize = 30

## 2. **parsac sensitivity run info**

## 3. **parsac sensitivity analyze --print\_to\_console info delta**

## 3. **parsac sensitivity analyze --pickle pickle info delta**

pickle = delta\_temp\_SA\_results.pkl



# Example of SA results

```
A fatal error has occurred during the function execution
```

```
File "C:\Users\tkan\AppData\Local\anaconda3\Lib\site-packages\pp-1.6.4.4-py3.11.egg\ppworker.py", line 103, in run
    __args = pickle.loads(ppc.b(__sargs))
              ^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
File "C:\Users\tkan\AppData\Local\anaconda3\Lib\site-packages\dill\session.py", line 32, in <module>
    import pathlib
```


```
File "C:\Users\tkan\AppData\Local\anaconda3\Lib\site-packages\pathlib.py", line 10, in <module>
    from collections import Sequence
```

```
Traceback (most recent call last):
```

File "&lt;frozen runpy&gt;", line 88, in run code

```
File "C:\Users\tkan\AppData\Local\anaconda3\Lib\site-packages\parsac\parsac_run.py", line 12, in main
    parsac.main()
```

File "C:\Users\tkan\AppData\Local\anaconda3\Lib\site-packages\parsac\sensitivity.py", line 225, in main



File "C:\Users\tkan\AppData\Local\anaconda3\Lib\site-packages\parsac\job\shared.py", line 276, in evaluate ensemble



Evaluating fitness with parameter set [1.17867,0.938987,0.691318,0.31

6997,7.77171].

Starting model run...

Model run took 257.1 s.

Evaluating fitness with parameter set [1.05591,1.16061,0.747843,0.40

3331,6.99702].

Starting model run...

Model run took 253.5 s.

Updating sensitivity info in C:\Users\tkan\Model\_setups\lake\_shahe\_workshop\parsac\delta\_temp\_n50.pkl with model result  
s...

ERROR: The process "18760" not found.

ERROR: The process "23008" not found.

ERROR: The process "25460" not found.

ERROR: The process with PID 21520 could not be terminated.

Reason: There is no running instance of the task.

(base) C:\Users\tkan>ERROR: The process "20604" not found.

ERROR: The process "4172" not found.

```
Anaconda Prompt

(base) C:\Users\tkan>parsac sensitivity analyze --pickle C:\Users\tkan\Model_setups\lake_shahe_workshop\parsac\delta_tem
mp_SA_results.pkl C:\Users\tkan\Model_setups\lake_shahe_workshop\parsac\delta_temp_n50.pkl delta --num_resamples 100
Reading sensitivity samples from C:\Users\tkan\Model_setups\lake_shahe_workshop\parsac\delta_temp_n50.pkl...
Reading configuration from C:\Users\tkan\Model_setups\lake_shahe_workshop\parsac\wet_dashahe-step1.xml...
Target 0 (output.nc/filter_by_time(temp[:,:],months=(1,2,3,4,5,6,7,8,9,10,11,12)).mean())
- gotm.yaml/surface/meteo/v10/scale_factor (0.41209798355387056)
- gotm.yaml/light_extinction/g2/constant_value (0.34144913782427877)
- gotm.yaml/surface/meteo/u10/scale_factor (0.14255830286030505)
- gotm.yaml/light_extinction/g1/constant_value (0.09012025527739215)
- gotm.yaml/light_extinction/A/constant_value (0.08663801836324822)
Writing analysis result to pickle C:\Users\tkan\Model_setups\lake_shahe_workshop\parsac\delta_temp_SA_results.pkl.

(base) C:\Users\tkan>
```

```
(base) C:\Users\tkan>parsac sensitivity analyze --print_to_console C:\Users\tkan\Model_setups\lake_shahe_workshop\parsa
c\delta_temp_n50.pkl delta --num_resamples 100
Reading sensitivity samples from C:\Users\tkan\Model_setups\lake_shahe_workshop\parsac\delta_temp_n50.pkl...
Reading configuration from C:\Users\tkan\Model_setups\lake_shahe_workshop\parsac\wet_dashahe-step1.xml...

              delta  delta_conf          S1    S1_conf
gotm.yaml/surface/meteo/u10/scale_factor    0.194747    0.065448  0.189696  0.215314
gotm.yaml/surface/meteo/v10/scale_factor    0.407474    0.087800  0.520617  0.119472
gotm.yaml/light_extinction/A/constant_value  0.141332    0.079204  0.070337  0.151453
gotm.yaml/light_extinction/g1/constant_value 0.232874    0.076295  0.172186  0.148453
gotm.yaml/light_extinction/g2/constant_value 0.089950    0.074734  0.163239  0.178264
Target 0 (output.nc/filter_by_time(temp[:, :], months=(1,2,3,4,5,6,7,8,9,10,11,12)).mean())
- gotm.yaml/surface/meteo/v10/scale_factor (0.40747447691683986)
- gotm.yaml/light_extinction/g1/constant_value (0.23287378411118495)
- gotm.yaml/surface/meteo/u10/scale_factor (0.19474707038533715)
- gotm.yaml/light_extinction/A/constant_value (0.14133170293054328)
- gotm.yaml/light_extinction/g2/constant_value (0.08994990623596802)

(base) C:\Users\tkan>
```

# Example of SA results

