Robust Parameter Estimation in a High-Mountainous Catchment showcasing Spotpy

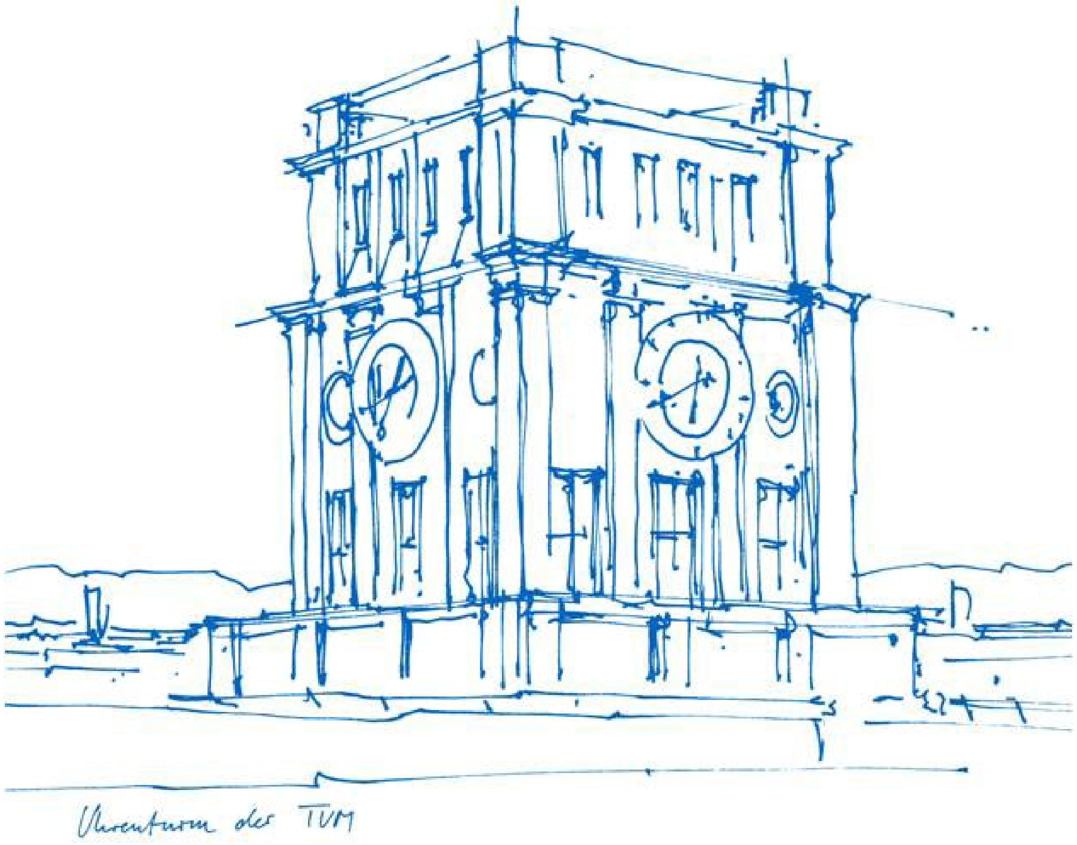
Lucas Alcamo[1], Timo Schaffhauser[1], Tobias Houska[2], Jingshui Huang[1], Markus Disse[1]

[1]Technical University of Munich, Germany

[2]Justus-Liebig-Universität Giessen, Germany

SWAT User Conference, Strasbourg

July 2024



Introduction – Two Topics

Setting-up robust SWAT+ Model

- **Central Asian catchment**
- **Multi-gauge and multi-parameter calibration**
- **Comparison ISIMIP2 and ISIMIP3**

SPOTPY

- **SPOTPY is a python-based calibration tool**
    - Multiple algorithms
    - Sensitivity analysis
    - Parallel processing
- **Introduce linkage between SWAT+ and SPOTPY**

Introduction – Why Central Asia?

- **Water highly contested resource**
- **„Aral Sea Disaster“**
  - Endorheic Lake – formerly 4th largest freshwater lake in the world
  - Volume decreased by 90% since 1960s
  - Caused largely by expansion of irrigation
- **Ferghana Valley**
  - Important agricultural area
  - Upstream hydropower generation threatens water supply during cropping seasons
- **Climate change**
  - Nivo-glacial dominated discharge regime

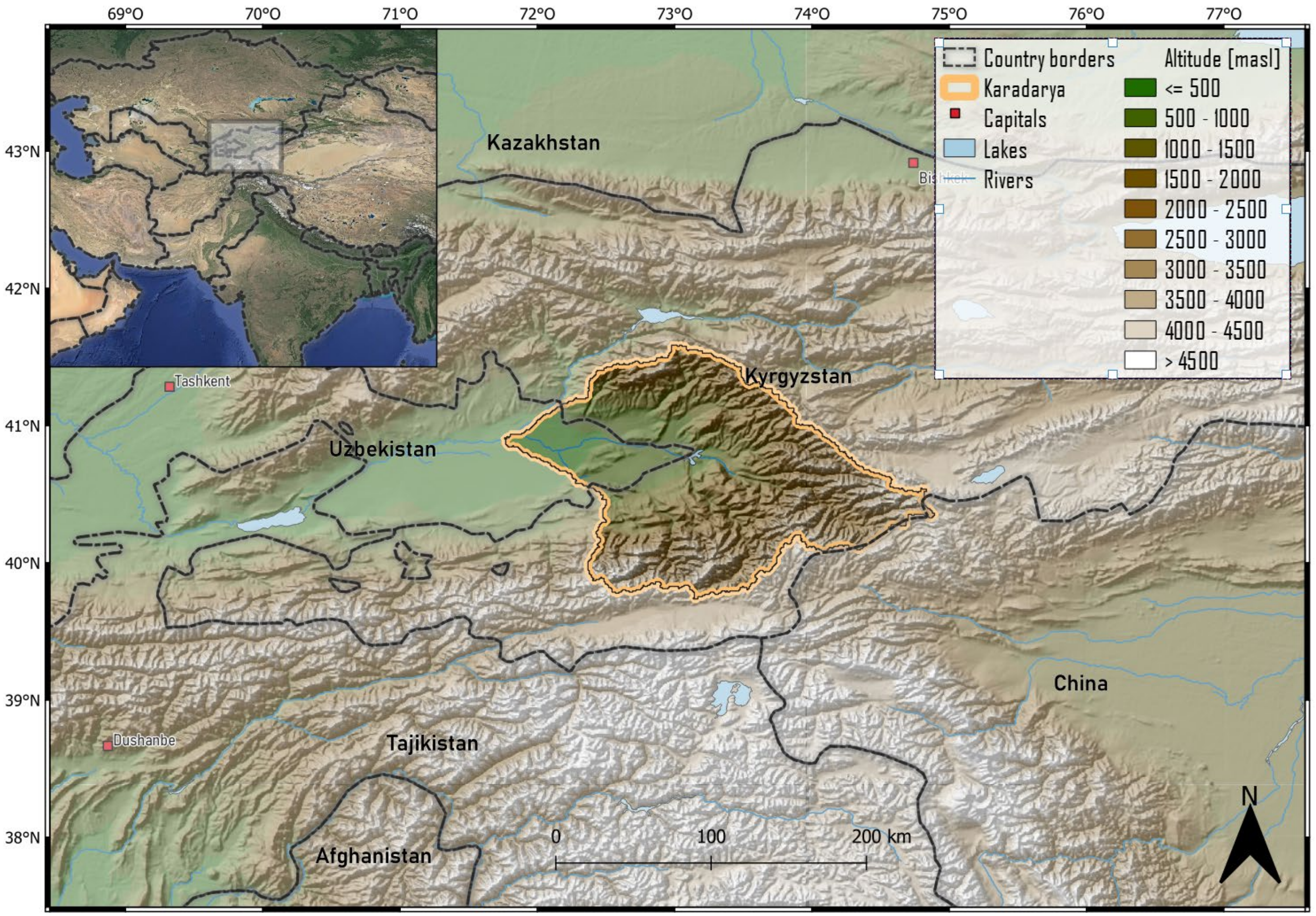
we act

Water Efficient Allocation in a Central Asian Transboundary River Basin

- Climate sensitive and sustainable water allocation
- Decision Support System

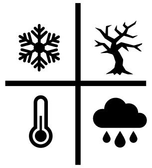
Introduction – Study Area

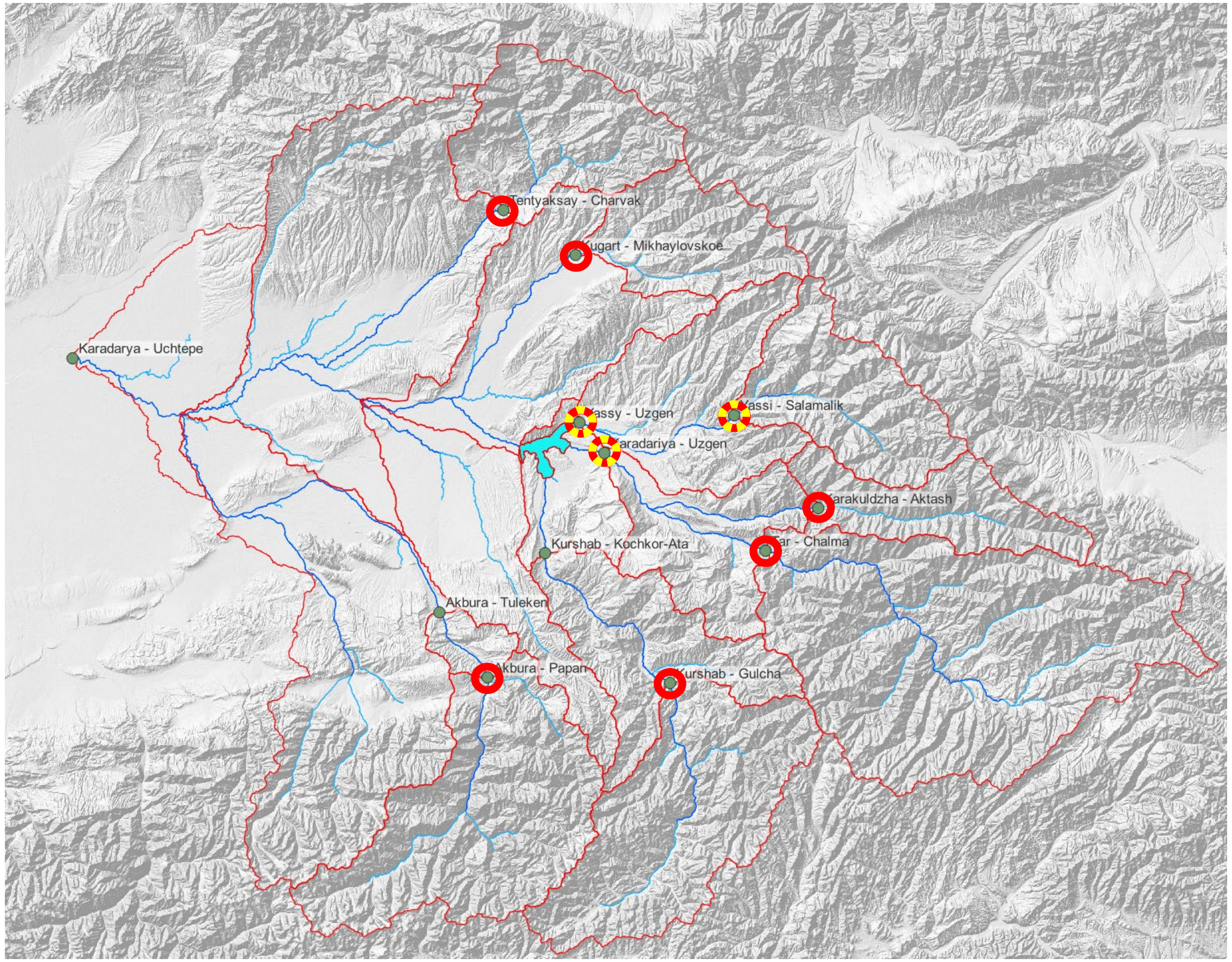
- Karadaraya catchment
- Headwater catchment of Syr Darya Rivers (tributary of Aral Sea)
- High mountains to the North, East and South
- Ferghana valley & agriculture in the West & lower areas
- Dry continental & semi-arid climate



Methodology

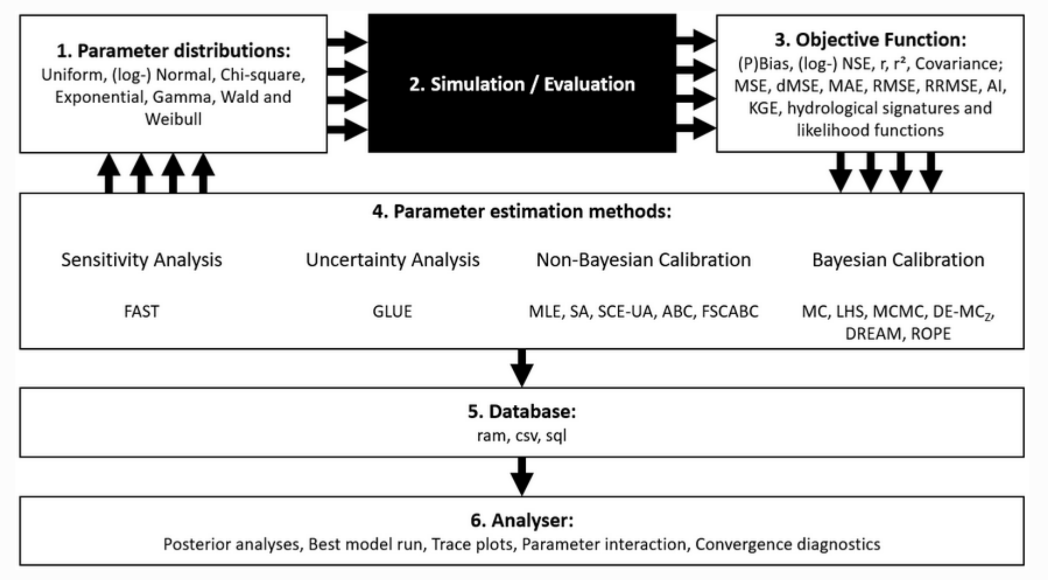
- **Input data**
  - GSWP-W5E5 (ISIMIP3)
- **Multi-site calibration**
  - 3 daily gauges
  - 6 monthly gauges
- **Multi-variable calibration**
  - Streamflow
  - Evapotranspiration
  - Soil moisture
- **Different climate regimes considered**
  - Hot/cold & wet/dry combinations
  - 1970s





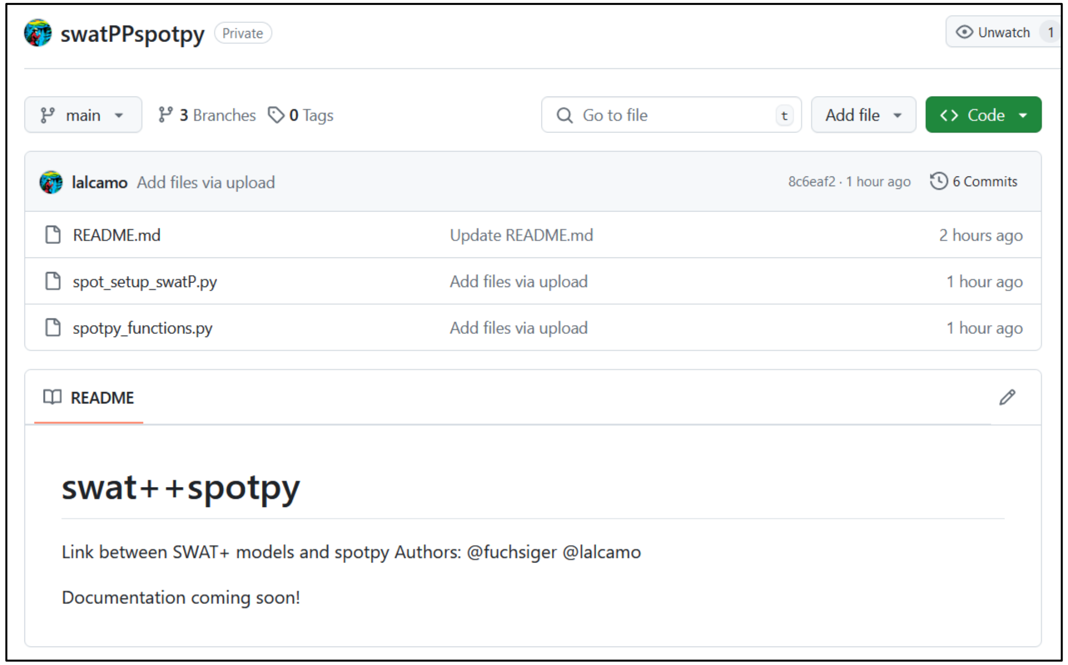
Methodology - SPOTPY

- **S**tatistical **P**arameter **O**ptimization **T**ool for **Py**thon
- Framework for optimization techniques for calibration
- Sensitivity analysis
- Parallel processing possible
- SWAT++SPOTPY creates interface between SWAT+ and SPOTPY



SWAT++SPOTPY – How to apply

- **Based on spotpy library**
  + python class „spot_setup_swatP“
- **SPOTPY setup**
  - Good documentation + installation
- **SWAT++SPOTPY**
  - 2 DataFrames needed to define parameters & variables
  - On occasion some changes to the code still needed
  - Available on Github



SWAT++SPOTPY – How to apply

- **Parameters (similar to calibration.cal)**

```
#                NAME          CHANGE_TYPE   LOWER_BOUND    Upper_BOUND   SUBBASIN (if all type 'None')
data_params = [['plaps',        'absval',      1.0,           None,          None       ],
               ['tlaps',        'absval',      3.0,           None,          None       ],
               ['cn2',          'pctchg',      0,             20,            [5]        ],
               ['cn2',          'pctchg',      10,            30,            [7]        ],
               ['perco',        'absval',      0.0,           1.0,           None       ]]

# Create DataFrame
df_params = pd.DataFrame(data_params, columns=['name', 'change_type', 'lower_bound', 'upper_bound', 'subbasins'])
```

Fixed parameters

Multiple subbasins calibrated

- **Variables**

```
#              VARIABLE (str)    TEMP_RES (str)        LOCATION (list)           OBS_FILE (str)                  CALIBRATION  WEIGHT       NAME
data_vars = [[  'discharge',      'daily',               [32],             'Discharge_Yassy_Uzgen.csv',        False,        1,     'Gauge Uzgen'  ],
             [  'discharge',      'monthly',             [32],             'Station_Data_16131_prep.csv',      True,         1,     'Gauge Uzgen'  ],
             [  'eta',            'monthly',      [1,2,3,4,5,6,7,8,9,10],  'kd_eta_m_obs_basin.csv',           True,         1,     'ETA basin'    ],
             [  'sm',             'monthly',      [1,2,3,4,5,6,7,8,9,10],  'kd_sm_m_obs_basin.csv',            True,         1,     'SM basin'     ]]
df_variables = pd.DataFrame(data_vars, columns=['variable', 'temp_res', 'location', 'obs_file','calib','weight', 'name'])
```

Channel number for discharge!

Mean eta / soil moisture for respective subbasins

Which parameters to calibrate – others are saved as well (validation)

SPOTPY – Sensitivity Analysis

- Global Sensitivity analysis with „FAST“
- Ammer catchment, Bavaria Germany
- 15 Parameters
- 26,000 runs

Parameter	Change Type	Lower bound	Upper bound
cn2	absval	35	95
cn3_swf	absval	0	1
perco	absval	0	1
esco	absval	0	1
epco	absval	0	1
snowfall_tmp	absval	-5	5
snomelt_tmp	absval	-5	5
snomelt_max	absval	0	10
snomelt_min	absval	0	10
snomelt_lag	absval	0	1
awc	absval	0.01	1
surlag	absval	0.05	24
deep_seep	absval	0.0001	0.4
plaps	absval	-10	10
tlaps	absval	-10	10

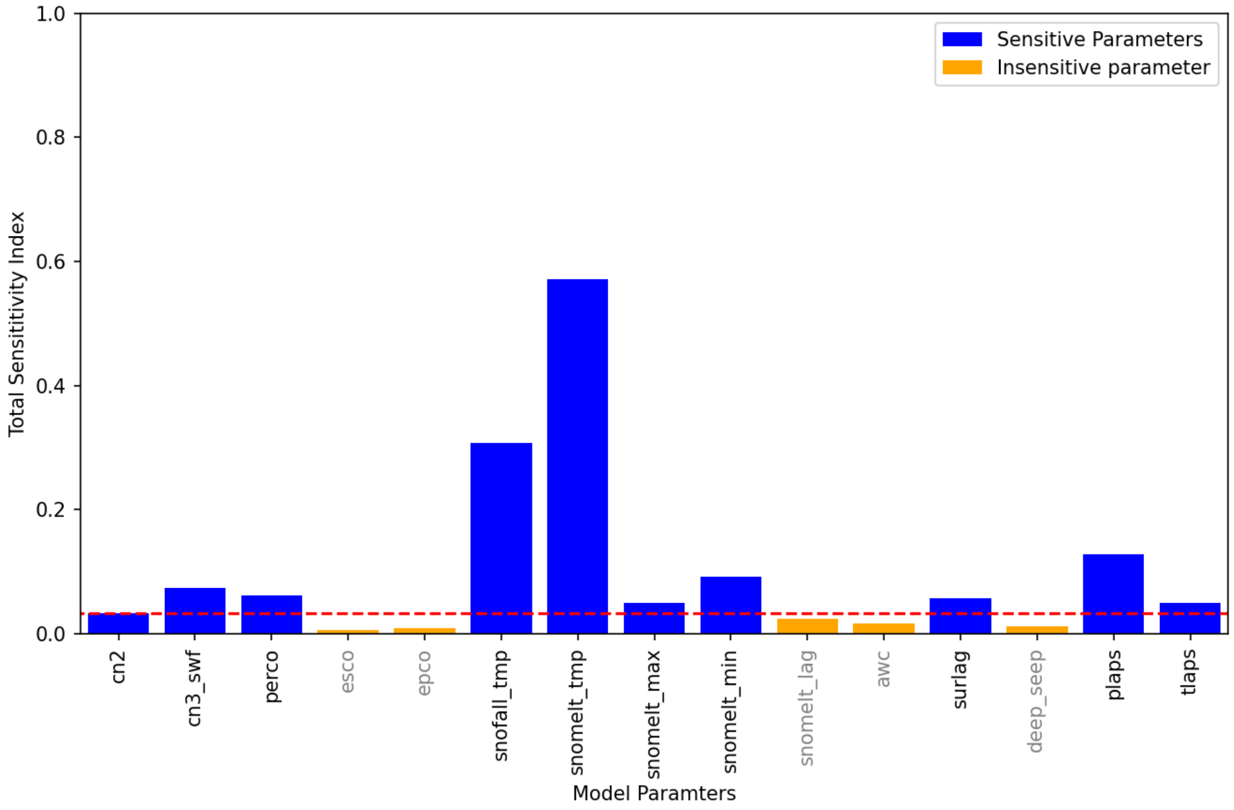


Figure: Results of Sensitivity analysis with FAST (Md Kazi Ismile Hossain)

SPOTPY – Calibration

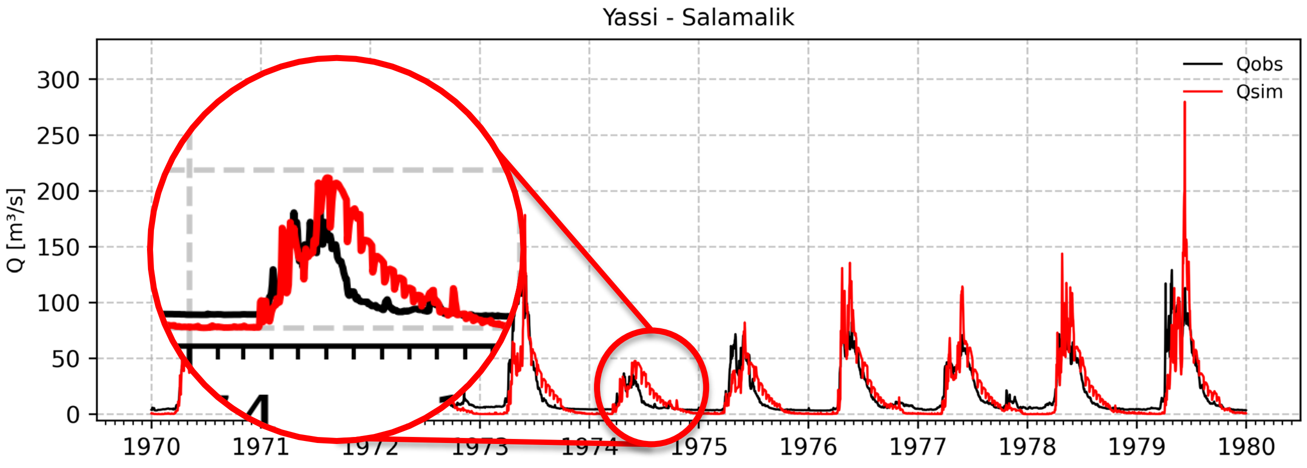
- Different Algorithms & variables tested (10,000 runs, 13 Parameters*)
- Evaluated for Gauge "Yassy Salamalik"

Algorithm	Objective Parameter	Temporal Resolution	Best Model Run Calibration [KGE]	Validation [KGE]
LHS	-	Daily	0.84	0.50
ROPE	Discharge	Daily	0.79	0.49
ROPE (8 Parameters)	Discharge	Daily	0.78	0.57
SCE-UA	Discharge	Daily	0.83	0.58
SCE-UA	Discharge + ETA	Monthly	0.89	0.76

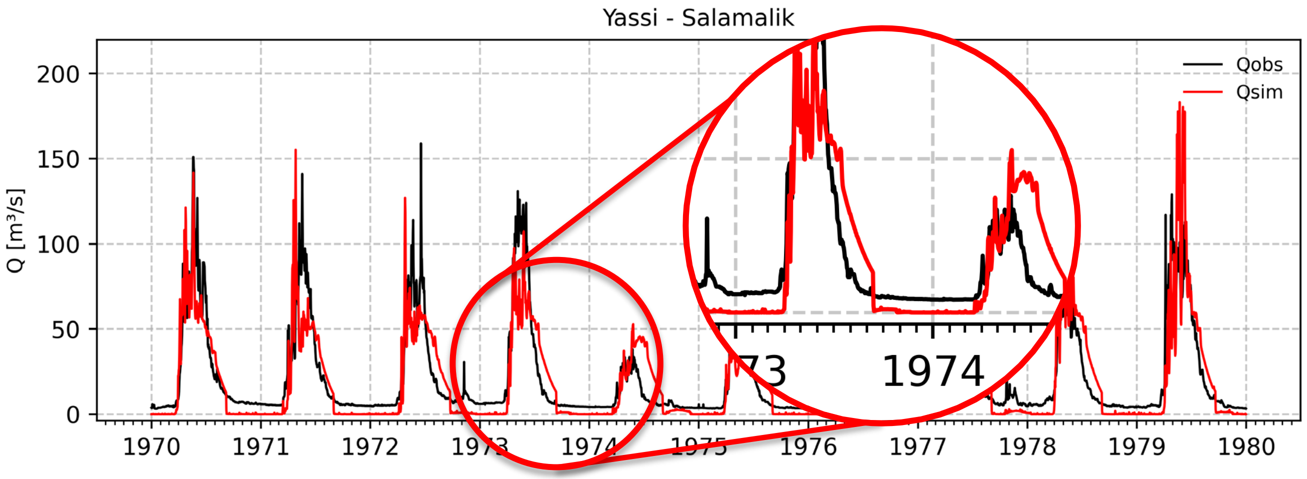
*unless indicated otherwise

SPOTPY – Calibration

- **Hydrographs**
  - SCE-UA (KGE=0.83)



  - ROPE (KGE=0.79)

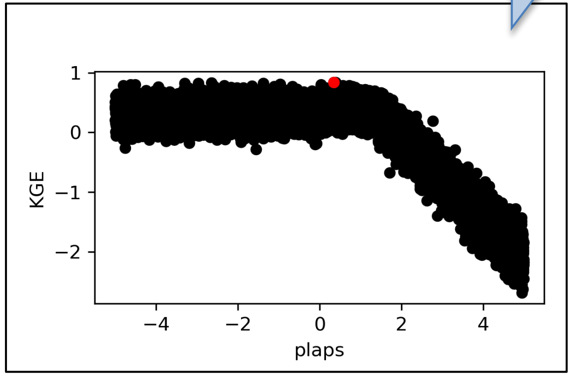


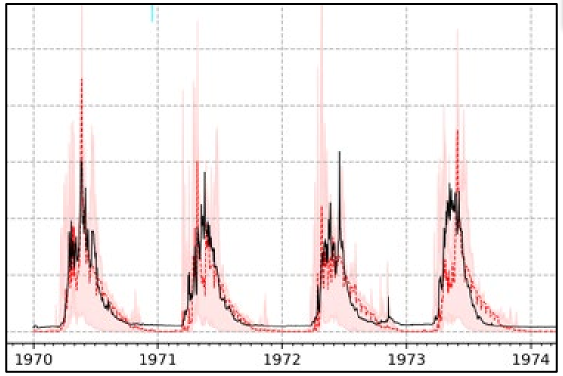
Manual calibration while considering the Hydrograph!

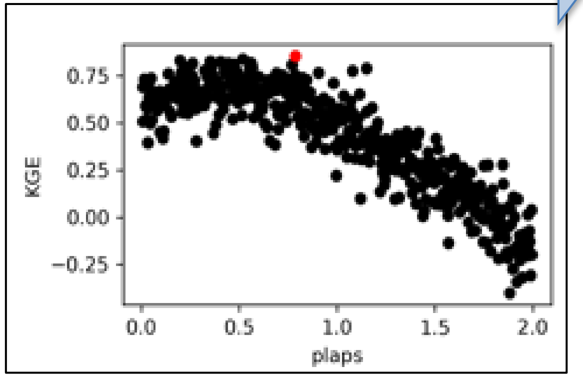
Calibration

- **Procedure:**

LHS with wide parameter ranges → Manual calibration for each gauge → SCE-UA with small parameter ranges







Results

Calibration:
Manual + Automatic

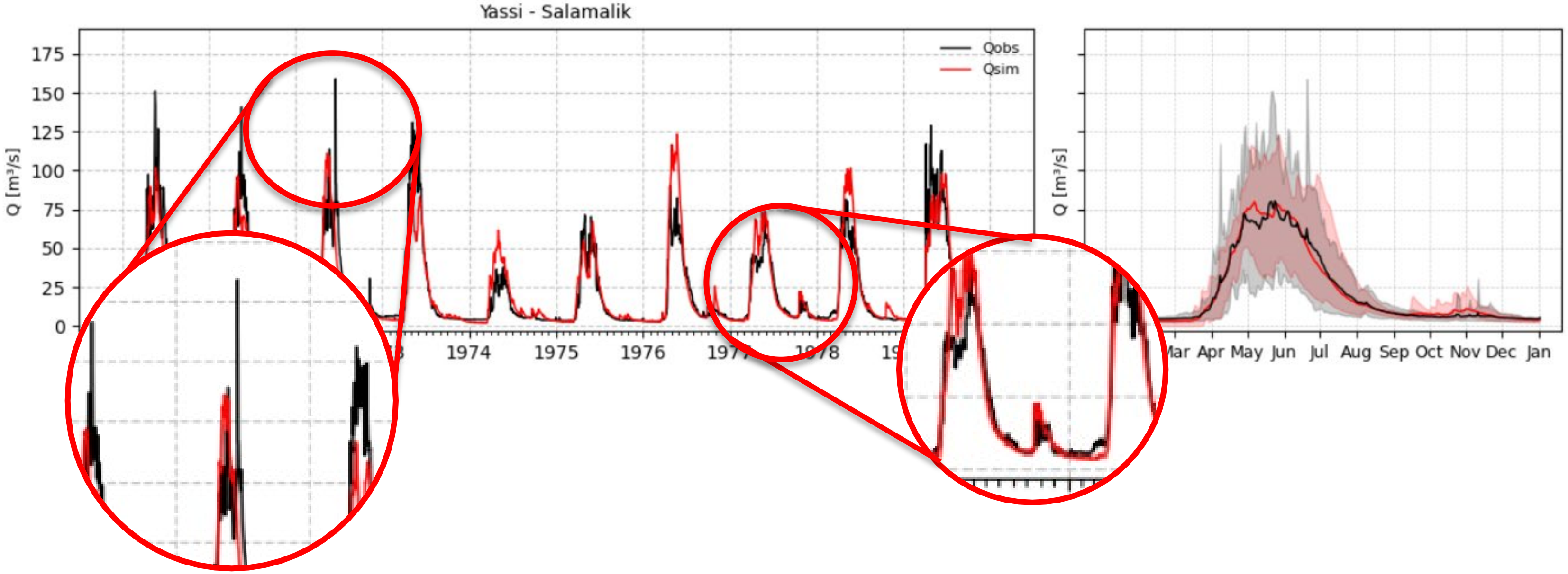
KGE: 0.90 (calibration)

KGE: 0.64 (validation)

Input Uncertainty:
Precipitation

→ lapse rates needed

→ only one lapse rate for catchment



Evaluation of areal precipitation estimates based on downscaled reanalysis and station data by hydrological modelling

D. Duethmann[1], J. Zimmer[1,*], A. Gafurov[1], A. Güntner[1], D. Kriegel[1], B. Merz[1], and S. Vorogushyn[1]

[1]GFZ German Research Centre for Geosciences, Telegrafenberg, 14473 Potsdam, Germany
[*]now at: Engineering Consultancy Load and Energy Management, Leipzig, Germany

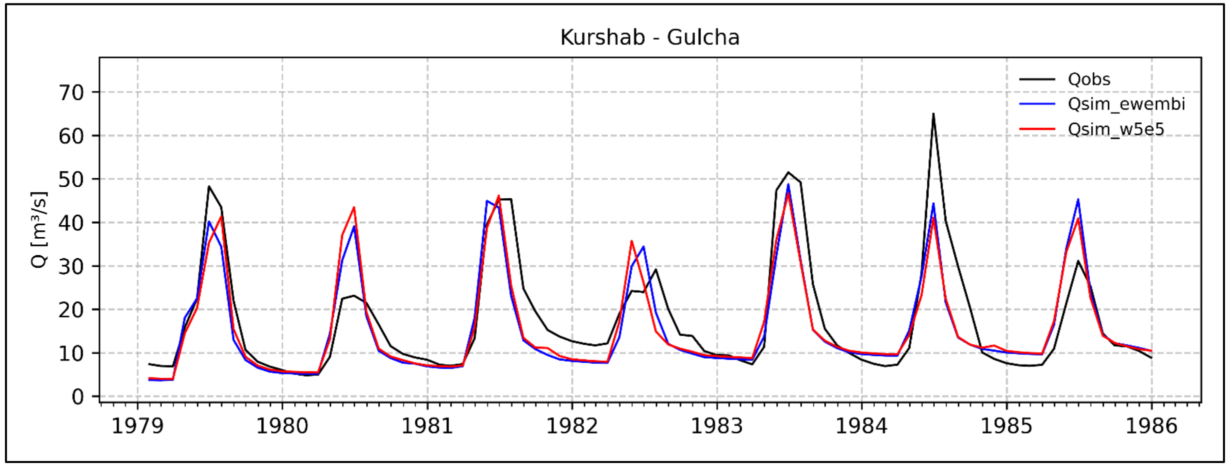
Correspondence to: D. Duethmann (doris.duethmann@gfz-potsdam.de)

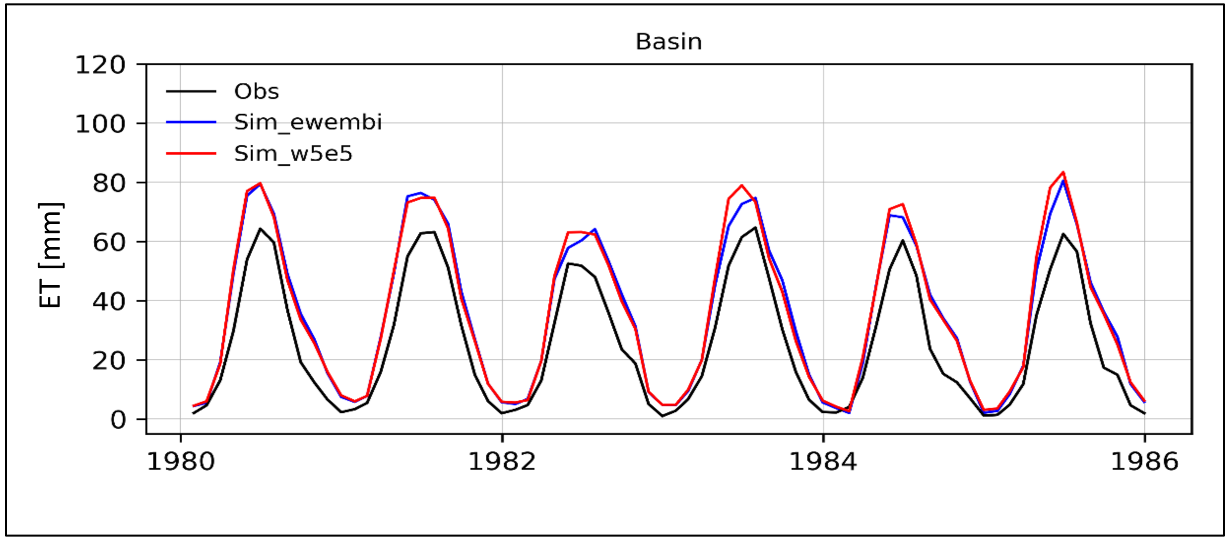
Received: 17 August 2012 – Published in Hydrol. Earth Syst. Sci. Discuss.: 20 September 2012
Revised: 6 May 2013 – Accepted: 16 May 2013 – Published: 2 July 2013

Results – W5E5 vs. EWEMBI

- **Discharge and ET**

	GSWP-W5E5	*GSWP-EWEMBI*
Gauge Gulcha	*NSE: 0.7* *KGE: 0.76*	*NSE: 0.7* *KGE: 0.77*
ET basin	*R: 0.98* *R^2: 0.97*	*R: 0.98* *R^2: 0.96*





Summary and Conclusions

- **SPOTPY: helpful for sensitivity analysis & calibration**
- **Simple application of automatic calibration tools might not be sufficient for SWAT+ in alpine catchments**
  - High number of parameters
  - Large interaction between parameters
- **Combined automatic & manual methods worked well to calibrate alpine Karadarya**

Outlook

- **SWAT++SPOTPY**
  - Basic structure and functions accessible on Github
  - Some Algorithms limited in current implementation (e.g.: NSGAII)
    - → Edit or add new version of NSGAII
  - Morris method for sensitivity analysis
    - → Simple global SA alternative
    - → Requires smaller number of runs (sufficient for larger parameter sets)

- **Climate and land-use change impact assessment**
  - ISIMIP3 datasets to run climate impact assessment
  - Scenarios of land-use change (reforestation, desertification)

References and Thanks

- Houska, T., Kraft, P., Chamorro-Chavez, A., Breuer, L., 2015. SPOTting Model Parameters Using a Ready-Made Python Package. PLOS ONE 10, e0145180.
- Duethmann, D., Zimmer, J., Gafurov, A., Güntner, A., Kriegel, D., Merz, B., Vorogushyn, S., 2013. Evaluation of areal precipitation estimates based on downscaled reanalysis and station data by hydrological modelling. Hydrology and Earth System Sciences 17, 2415–2434.

Special thanks to my co-authors and student, Md Kazi Ismile Hossain

Thank you for your attention!

Feedback & Questions welcome!

SPOTPY – SWAT+ linkage

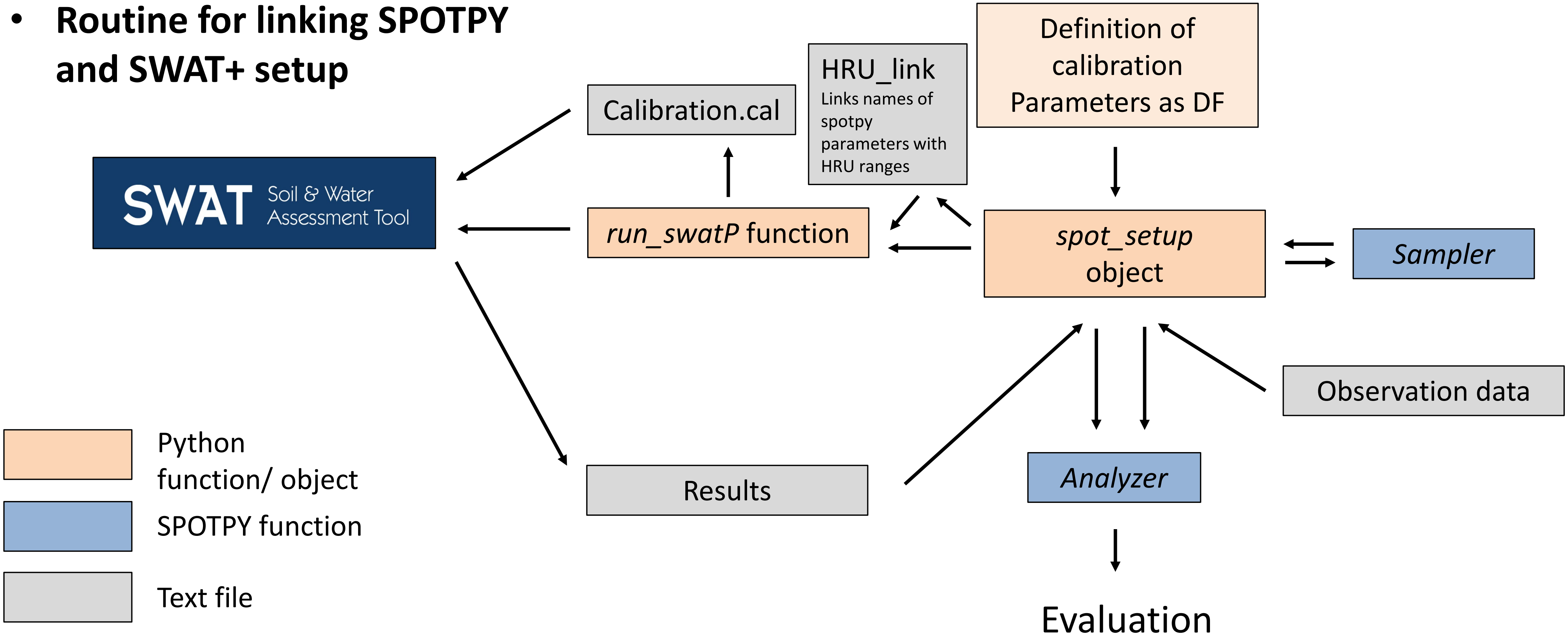
- **Own routine for linking SPOTPY and SWAT+ written**
  - Taking advantage of "calibration.cal" file (also used for SWAT+ Toolbox)
  - Two components only
    - "run_swatP" function
    - "spot_setup" object

```
def run_swatP(path_TxtInOut, name_exe, parameter_values, channel_id):

    # Give channel_id the default value if not defined or wrongly defined
    try:
        if channel_id == None or channel_id == 0 or type(channel_id) != int:
            channel_id = 1
    except NameError:
        channel_id = 1

    # Start time
    t_start = datetime.now()

    ########################################################
    # Edit calibration.cal file with given parameter_values
    ########################################################
    '''
    !!! Number and order of parameters and code have to be compatible !!!
    '''
    with open(f"{path_TxtInOut}calibration_read.cal", "r") as file:

        lines = file.readlines()
```

```
import sys
sys.path.insert(0, 'D:/spotpy_SWATp/python_files/')
from spotpy_functions import run_swatP

class spot_setup(object):

    # Define paramters
    cn2          = Uniform(name='cn2'   , low=-10.0 , high=15.0)
    perco        = Uniform(name='perco', low=0.0 , high=20.0)
    k            = Uniform(name='k'     , low=-30.0 , high=30.0)
    awc          = Uniform(name='awc'   , low=-30.0 , high=20.0)
    snofall_tmp = Uniform(name='snofall_tmp', low=-2.0 , high=5.0) # -5, 5 Maximum Range!
    snomelt_tmp = Uniform(name='snomelt_tmp', low=0.0 , high=5.0) # -5, 5 Maximum Range!
    snomelt_max = Uniform(name='snomelt_max', low=0.0  , high=3.0) # 0, 5 Maximum Range!
    snomelt_min = Uniform(name='snomelt_min', low=0.0  , high=4.0) # 0, 5 Maximum Range!
    snomelt_lag = Uniform(name='snomelt_lag', low=0.0  , high=1.0) # 0, 1Maximum Range!
    surlag       = Uniform(name='surlag'     , low=0.05 , high=20) # 0.05, 24 Maximum Range!

    def __init__(self, obj_func=None):

        self.obj_func       = obj_func
```

SPOTPY – SWAT+ linkage

- **Routine for linking SPOTPY and SWAT+ setup**



SPOTPY – SWAT+ linkage

- **“Spotswatplus” library exists!**
  - Linkage between spotpy and SWAT+
  - Used to perform sensitivity analyses and calibration of simulated streamflow of the Uruguay River
  - Documentation in progress
  - Personal contact established

water MDPI

Article

Representation of Hydrological Components under a Changing Climate—A Case Study of the Uruguay River Basin Using the New Version of the Soil and Water Assessment Tool Model (SWAT+)

Osvaldo Luis Barresi Armoa [1,*], Sabine Sauvage [1], Tobias Houska [2], Katrin Bieger [3], Christoph Schürz [4] and José Miguel Sánchez Pérez [1,*]