pyClim-SDM: User Manual

v3.7 2023-08-25

Versions history

- v3.7: additional predictors introduced
- v3.6: Added multiprocessing
- v3.5: Added control of versions. When a new version is executed, default settings are restored
- v3.4: Robust filters for the spatial domains (selected and contained in input files) implemented
- v3.3: Fixed incompatibilities between the selected spatial domain and the spatial domain contained at the input files
- v3.2: Added User Manual
- v3.1: New variables included: specific humidity, radiation, evaporation, pressure, runoff, soil mositure
- v3.0:
  - a large number of target variables have been added: tas, uas, vas, sfcWind, hurs, clt
  - additionally, users can define their own target variable
  - many names have changed to follow the CMIP convention
  - toy datasets from GCMs have been included in the input_data_template
  - it has been tested for newer versions of the required python libraries, so installation instructions and requirements.txt have changed

Glossary

AEMET - Spanish Meteorological Agency
GCM - Global Climate Model
GUI - Graphical User Interface
HPC- High Performance Computing
SAF - Synoptic Analogy Field
SLURM - Simple Linux Utility for Resource Management
SDM - Statistical Downscaling Model

Introduction

pyClim-SDM is a software for statistical downscaling of climate change projections for the following daily surface variables: mean, maximum and minimum temperature, precipitation, zonal and meridional wind components, relative and specific humidity, cloud cover, surface downwelling shortwave and longwave radiation, evaporation, potential evaporation, sea level pressure, surface pressure, total runoff and soil water content. Additionally, it is prepared for downscaling any other user defined variable.

pyClim-SDM has been developed by the AEMET and it is freely available at https://github.com/ahernanzl/pyClim-SDM

pyClim-SDM has been dotted with a GUI interface/mode and this manual explains the different options that the user can select from it. **Nonetheless, the GUI displays many informative messages for the different options when the mouse pointer is on top of them.**

A screen resolution of at least 1280 x 620 (width x height) is needed.

Input data

Three types of datasets are needed:

1. hres: high-resolution observations.
2. reanalysis: predictors from a reanalysis
3. models: predictors from GCMs

Format:

- hres format (high-resolution observations):
    - One row per date. The first column corresponds to the date yyyymmdd, and the other rows (as many as target points) contain data (if observations come from a regular 2D grid, they need to be flattened to a 1D list of points). Missing data must be coded as -999.
    - Temperature (tas/tasmax/tasmin) in degrees
    - Precipitation (pr) in mm
    - Wind (uas/vas/sfcWind) in m/s
    - Relative humidity (hurs) in %
    - Specific humidity (huss) dimensionless
    - Cloud cover (clt) in %
    - Radiation (shortwave rsds and longwave rlds) in W/m2
    - Evaporation (evspsbl) and potential evaporation (evspsblpot) in kg m-2 s-1
    - Sea level and surface pressure (psl and ps) in Pa
    - Total runoff (mrro) in kg m-2 s-1
    - Soil water content (mrso) in kg m-2
- Reanalysis and models format (low resolution predictors for calibration and downscaling): One netCDF file per variable, models and scene, with all pressure levels, in a regular 2D grid.
- Filenames: filenames are composed of specific fields separated by '_', so the use of this symbol inside a field (the reanalysis name, for example) must be avoided.

Outputs

Outputs are stored at ‘results/’ into different subfolders. Each experiment (see tab: Experiment and Steps) produces a different folder:

- PRECONTROL: this folder will contain auxiliary data, but the actual output of this experiment are figures stored at the ‘Figures/’ folder.
- EVALUATION: this folder contains different subfolders named by the target variable and the downscaling method. Then two subfolders (‘daily_data’ and ‘climdex’) contain the outputs in netCDF.
- PROJECTIONS: same as EVALUATION.

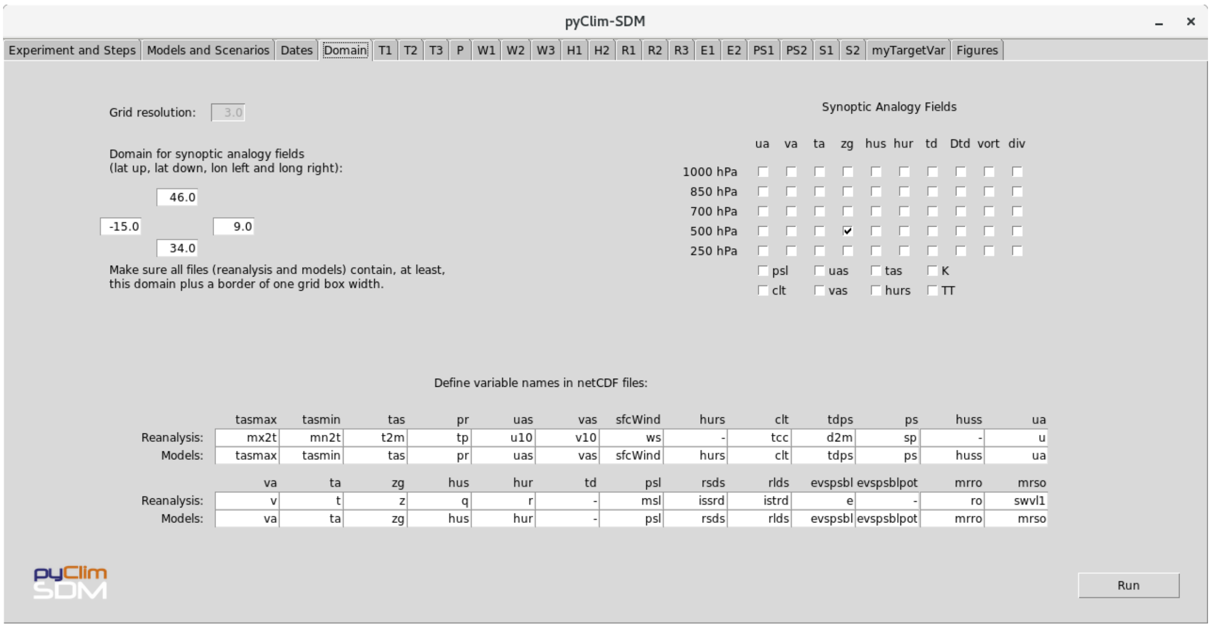
When netCDFs are converted to ASCII files, two new folders are generated: EVALUATION_ASCII and PROJECTIONS_ASCII.

When a bias correction is applied, these folders are duplicated with the suffix ‘-BC’ followed by the bias correction method used, and ‘-s’ if it has been applied by season.

Getting started

- For your first steps you can use some example datsets included in the 'input_data_template' just by renaming this folder as 'input_data', but **limit your selection to the default sets of predictors and target variables**. Be aware that only a few predictors have been included as well as few target points, so no conclusion about the methods skill must be reached using these data
- Open a terminal, go to the src/ directory and run 'python gui_mode.py'
- Alternatively, pyClim-SDM can be used without the graphical interface by running manual_mode.py and tuning the config/manual_settings.py file.
- In order to use your own datasets, spatial domain, etc., prepare your 'input_data' directory following the structure and format of the 'input_data_template'. Beware that the targetVariables themselves, given by reanalysis/GCMs are mandatory files for some methods and purposes.

tab: Domain



Grid resolution: 3.0

Domain for synoptic analogy fields
(lat up, lat down, lon left and long right):

46.0
-15.0 9.0
34.0

Make sure all files (reanalysis and models) contain, at least,
this domain plus a border of one grid box width.

Synoptic Analogy Fields

	ua	va	ta	zg	hus	hur	td	Dtd	vort	div
1000 hPa	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
850 hPa	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
700 hPa	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
500 hPa	☐	☐	☐	☑	☐	☐	☐	☐	☐	☐
250 hPa	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

☐ psl ☐ uas ☐ tas ☐ K
☐ clt ☐ vas ☐ hurs ☐ TT

Define variable names in netCDF files:

	tasmax	tasmin	tas	pr	uas	vas	sfcWind	hurs	clt	tdps	ps	huss	ua
Reanalysis:	mx2t	mn2t	t2m	tp	u10	v10	ws	-	tcc	d2m	sp	-	u
Models:	tasmax	tasmin	tas	pr	uas	vas	sfcWind	hurs	clt	tdps	ps	huss	ua

	va	ta	zg	hus	hur	td	psl	rsds	rlds	evspsbl	evspsblpot	mrro	mrso
Reanalysis:	v	t	z	q	r	-	msl	issrd	istrd	e	-	ro	swvl1
Models:	va	ta	zg	hus	hur	-	psl	rsds	rlds	evspsbl	evspsblpot	mrro	mrso

Grid resolution: the low grid resolution that reanalysis and models share is automatically detected from netCDF files.

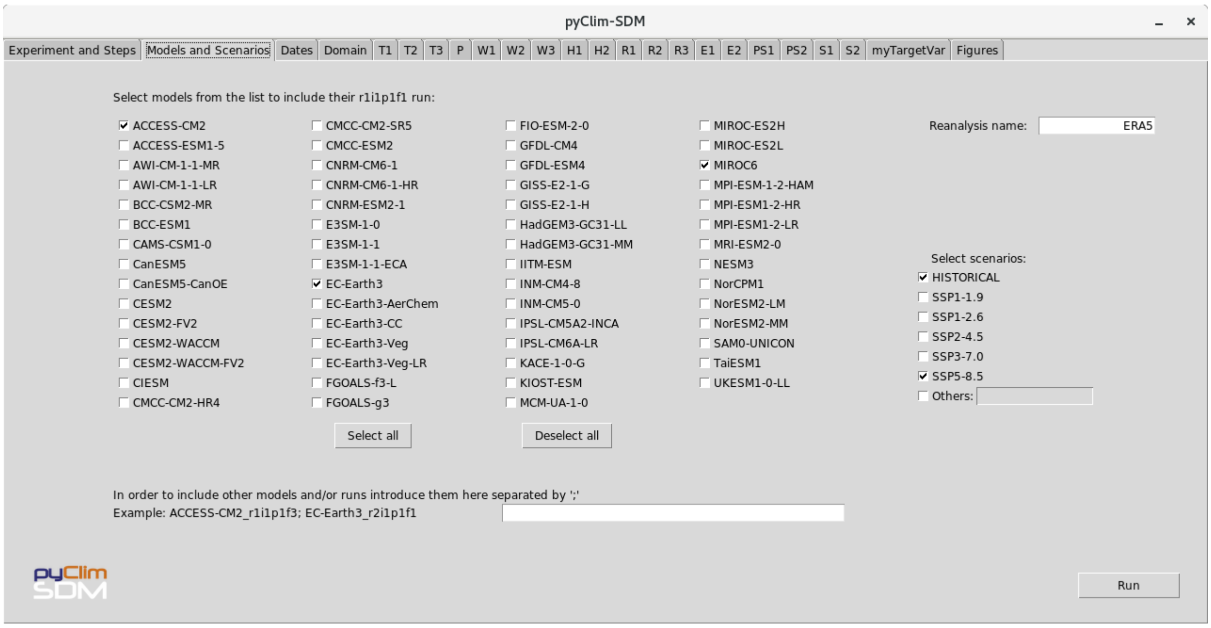
Domain for synoptic analogy fields: define the limits of your spatial domain (not your target points, but a larger domain in which the synoptic patterns will be analyzed for methods based on Analogs or in Weather Types). netCDFs from reanalysis and models need to contain, at least, this domain plus a one-gridbox border.

Synoptic Analogy Fields: select which fields will be used for Analogs and Weather Types methods.

Variable names: each variable is allowed to be named differently by the reanalysis and GCMs, so it needs to be specified. Default names correspond to the ERA5 reanalysis (ECMWF) and CMIP6 models.

Unless the user needs to analyze different fields for Analogs/Weather Types methods, this tab is defined only once at the beginning of a new project.

tab: Models and Scenarios



Select models from the list to include their r1i1p1f1 run:

☑ ACCESS-CM2
□ ACCESS-ESM1-5
□ AWI-CM-1-1-MR
□ AWI-CM-1-1-LR
□ BCC-CSM2-MR
□ BCC-ESM1
□ CAMS-CSM1-0
□ CanESM5
□ CanESM5-CanOE
□ CESM2
□ CESM2-FV2
□ CESM2-WACCM
□ CESM2-WACCM-FV2
□ CIESM
□ CMCC-CM2-HR4
□ CMCC-CM2-SR5
□ CMCC-ESM2
□ CNRM-CM6-1
□ CNRM-CM6-1-HR
□ CNRM-ESM2-1
□ E3SM-1-0
□ E3SM-1-1
□ E3SM-1-1-ECA
☑ EC-Earth3
□ EC-Earth3-AerChem
□ EC-Earth3-CC
□ EC-Earth3-Veg
□ EC-Earth3-Veg-LR
□ FGOALS-f3-L
□ FGOALS-g3
□ FIO-ESM-2-0
□ GFDL-CM4
□ GFDL-ESM4
□ GISS-E2-1-G
□ GISS-E2-1-H
□ HadGEM3-GC31-LL
□ HadGEM3-GC31-MM
□ IITM-ESM
□ INM-CM4-8
□ INM-CM5-0
□ IPSL-CM5A2-INCA
□ IPSL-CM6A-LR
□ KACE-1-0-G
□ KIOST-ESM
□ MCM-UA-1-0
□ MIROC-ES2H
□ MIROC-ES2L
☑ MIROC6
□ MPI-ESM-1-2-HAM
□ MPI-ESM1-2-HR
□ MPI-ESM1-2-LR
□ MRI-ESM2-0
□ NESM3
□ NorCPM1
□ NorESM2-LM
□ NorESM2-MM
□ SAM0-UNICON
□ TaiESM1
□ UKESM1-0-LL

Reanalysis name: ERA5

Select scenarios:
☑ HISTORICAL
□ SSP1-1.9
□ SSP1-2.6
□ SSP2-4.5
□ SSP3-7.0
☑ SSP5-8.5
□ Others:

In order to include other models and/or runs introduce them here separated by ';'
Example: ACCESS-CM2_r1i1p1f3; EC-Earth3_r2i1p1f1

In this tab the user needs to provide the following information:

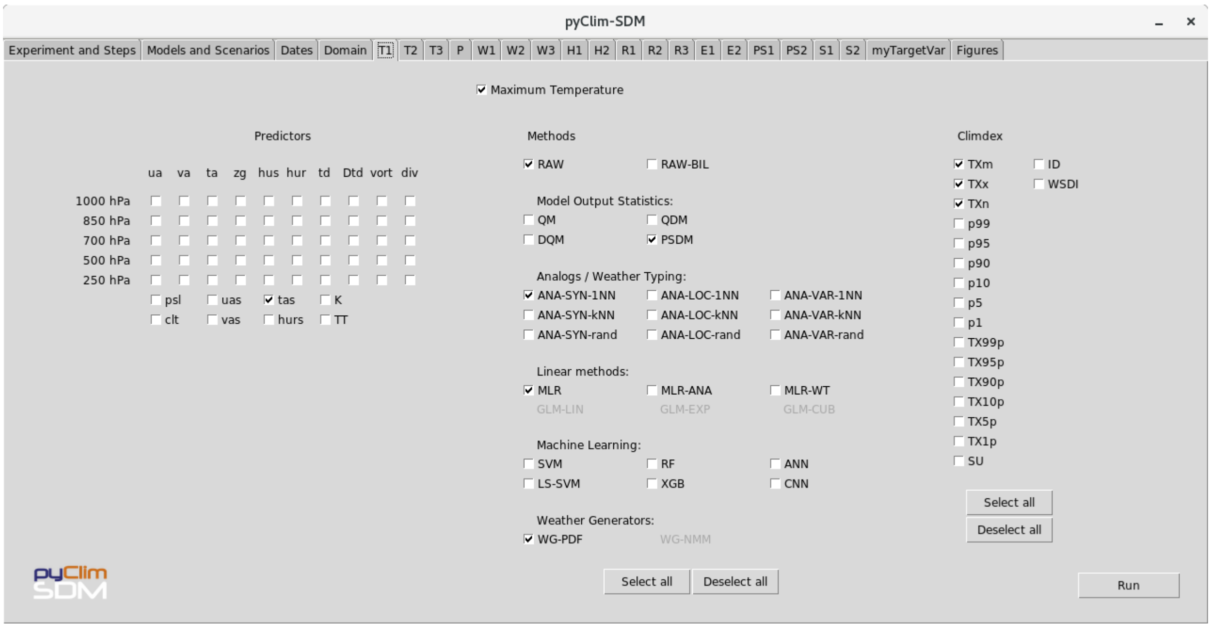
Reanalysis name: the name here specified must be the one in the reanalysis filenames.

Models: a list of CMIP6 GCMs for run r1i1p1f1 is offered, but other GCMs or runs can be used.

Scenarios: a list of frequently used emission scenarios is offered (SSPs), but others can be used (RCPs, SRES, …).

Unless the user needs to analyze different GCMs ensembles, this tab is defined only once.

tab: T1 to S2 (Target Variables)


pyClim-SDM
Experiment and Steps | Models and Scenarios | Dates | Domain | T1 | T2 | T3 | P | W1 | W2 | W3 | H1 | H2 | R1 | R2 | R3 | E1 | E2 | PS1 | PS2 | S1 | S2 | myTargetVar | Figures
Select all
Deselect all
Run

☑ Maximum Temperature

Predictors

	ua	va	ta	zg	hus	hur	td	Dtd	vort	div
1000 hPa	□	□	□	□	□	□	□	□	□	□
850 hPa	□	□	□	□	□	□	□	□	□	□
700 hPa	□	□	□	□	□	□	□	□	□	□
500 hPa	□	□	□	□	□	□	□	□	□	□
250 hPa	□	□	□	□	□	□	□	□	□	□

□ psl □ uas ☑ tas □ K
□ clt □ vas □ hurs □ TT

Methods

☑ RAW □ RAW-BIL

Model Output Statistics:
□ QM □ QDM
□ DQM ☑ PSDM

Analogs / Weather Typing:
☑ ANA-SYN-1NN □ ANA-LOC-1NN □ ANA-VAR-1NN
□ ANA-SYN-kNN □ ANA-LOC-kNN □ ANA-VAR-kNN
□ ANA-SYN-rand □ ANA-LOC-rand □ ANA-VAR-rand

Linear methods:
☑ MLR □ MLR-ANA □ MLR-WT
GLM-LIN GLM-EXP GLM-CUB

Machine Learning:
□ SVM □ RF □ ANN
□ LS-SVM □ XGB □ CNN

Weather Generators:
☑ WG-PDF WG-NMM

Climdex

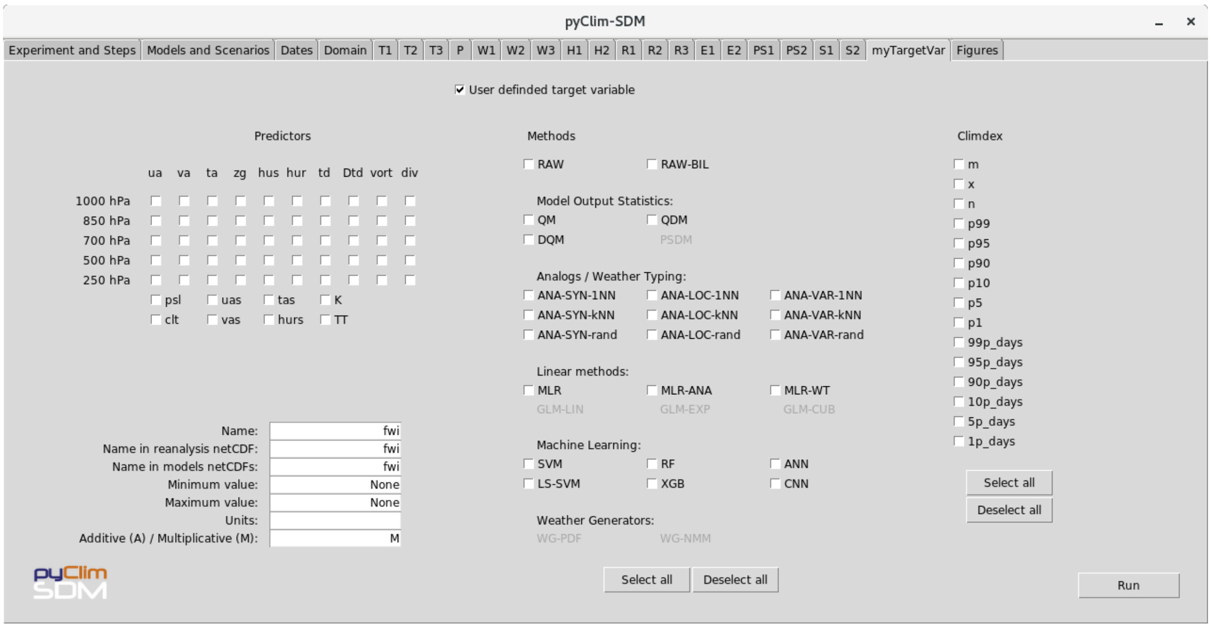
☑ TXm □ ID
☑ TXx □ WSDI
☑ TXn
□ p99
□ p95
□ p90
□ p10
□ p5
□ p1
□ TX99p
□ TX95p
□ TX90p
□ TX10p
□ TX5p
□ TX1p
□ SU

Different target variables are offered in tabs from T1 to S2, and myTargetVar tab offers the opportunity to define any not included potential target variable.

Each target variable can be **activated/deactivated** with the top check button.

For each target variable different **predictors** and **methods** can be used, and each target variable offers a list of **climdex** to be calculated (mean/accumulated values, extremes, spells, etc). Note that some methods will use the selected predictors (‘pred’), but other methods will use the Synoptic Analogy Fields (‘saf’) selected at tab Domain, and other methods will use the target variable itself (‘var’). Each method uses certain fields (‘pred’, ‘saf’ or ‘var’), which is defined at ‘config/advanced_settings.py’. Thus, the user just needs to define which variables to use as ‘pred’ and ‘saf’.

tab: myTargetVar


pyClim-SDM
Experiment and Steps | Models and Scenarios | Dates | Domain | T1 | T2 | T3 | P | W1 | W2 | W3 | H1 | H2 | R1 | R2 | R3 | E1 | E2 | PS1 | PS2 | S1 | S2 | myTargetVar | Figures
Select all
Deselect all
Run

☑ User definded target variable

Predictors

	ua	va	ta	zg	hus	hur	td	Dtd	vort	div
1000 hPa	□	□	□	□	□	□	□	□	□	□
850 hPa	□	□	□	□	□	□	□	□	□	□
700 hPa	□	□	□	□	□	□	□	□	□	□
500 hPa	□	□	□	□	□	□	□	□	□	□
250 hPa	□	□	□	□	□	□	□	□	□	□

□ psl □ uas □ tas □ K
□ clt □ vas □ hurs □ TT

Name: fwi
Name in reanalysis netCDF: fwi
Name in models netCDFs: fwi
Minimum value: None
Maximum value: None
Units:
Additive (A) / Multiplicative (M): M

Methods

□ RAW □ RAW-BIL

Model Output Statistics:
□ QM □ QDM
□ DQM PSDM

Analogs / Weather Typing:
□ ANA-SYN-1NN □ ANA-LOC-1NN □ ANA-VAR-1NN
□ ANA-SYN-kNN □ ANA-LOC-kNN □ ANA-VAR-kNN
□ ANA-SYN-rand □ ANA-LOC-rand □ ANA-VAR-rand

Linear methods:
□ MLR □ MLR-ANA □ MLR-WT
GLM-LIN GLM-EXP GLM-CUB

Machine Learning:
□ SVM □ RF □ ANN
□ LS-SVM □ XGB □ CNN

Weather Generators:
WG-PDF WG-NMM

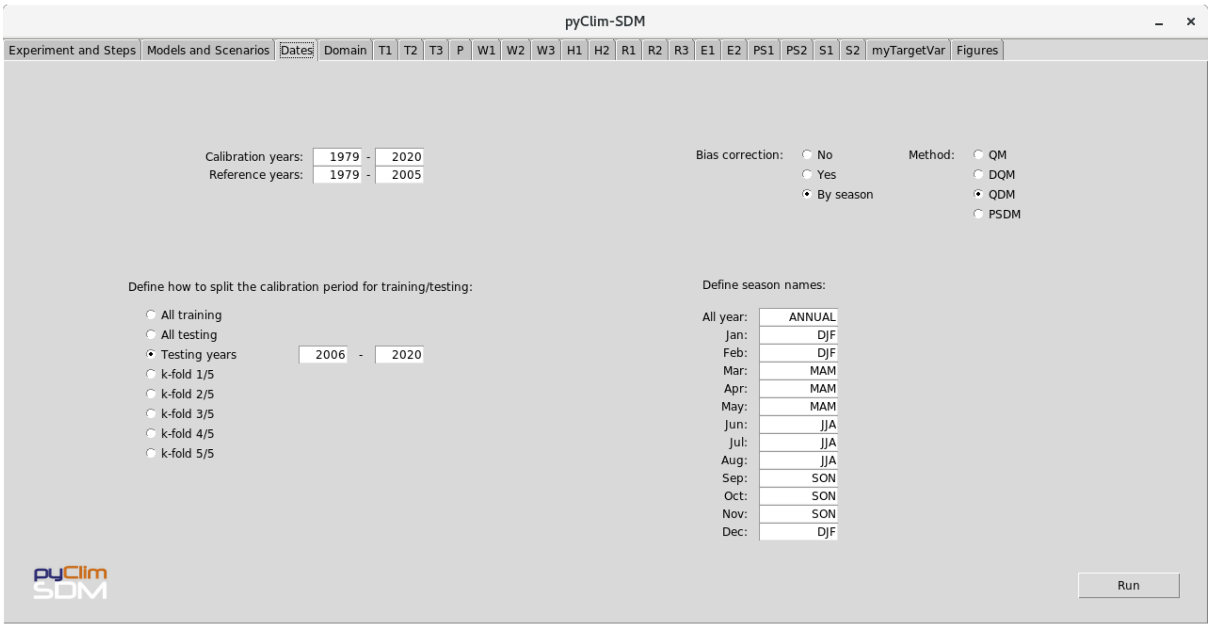
Climdex

□ m
□ x
□ n
□ p99
□ p95
□ p90
□ p10
□ p5
□ p1
□ 99p_days
□ 95p_days
□ 90p_days
□ 10p_days
□ 5p_days
□ 1p_days

For the user defined target variable, additional information is needed:

Name: in the hres filenames.
Name in the reanalysis netCDFs.
Name in the reanalysis netCDFs.
Minimum theoretical value (for example, 0 for precipitation).
Maximum theoretical value (for example, 100 for relative humidity).
Units (leave in blank if dimensionless).
Additive/Multiplicative: if biases (for evaluation) and future anomalies (for projections) must be calculated in an additive (absolute) or multiplicative (relative, %) way. For example, temperature is usually treated as additive, but precipitation as multiplicative.

tab: Dates



Calibration years: 1979 - 2020
Reference years: 1979 - 2005

Bias correction: ○ No ○ Yes ● By season

Method: ○ QM ○ DQM ● QDM ○ PSDM

Define how to split the calibration period for training/testing:

○ All training
○ All testing
● Testing years 2006 - 2020
○ k-fold 1/5
○ k-fold 2/5
○ k-fold 3/5
○ k-fold 4/5
○ k-fold 5/5

Define season names:

All year:	ANNUAL
Jan:	DJF
Feb:	DJF
Mar:	MAM
Apr:	MAM
May:	MAM
Jun:	JJA
Jul:	JJA
Aug:	JJA
Sep:	SON
Oct:	SON
Nov:	SON
Dec:	DJF

Calibration years: the user needs to specify the longest period available in both reanalysis and hres.
Reference years: the user needs to specify a reference period, which has to be available by all datasets (hres, reanalysis and models).
All this information is specified only once at the beginning of a new project.

Training/testing split:
The calibration years can be used for different purposes. For evaluation purposes, the calibration period must be split in training and testing.This is done by selecting **Single train/test split** and defining the years used for testing (the rest would be used for training). If our calibration period is not long enough and splitting it in train/test would lead to a poor training or testing, a k-fold approach should be used. This consists in training the methods with part of the dataset and downscaling the rest five times, so finally the whole calibration period has been downscaling (preserving the independence between the training and testing datasets). For this option, the user must select **k-fold 1/5**, define the testing years of the first split and run steps 'preprocess', 'train methods' and 'downscale' selecting experiment 'EVALUATION' in the Experiment and Steps tab (explained later). These three steps need to be run four more times, selecting k-folds from 2 to 5 and defining their corresponding testing years. Each fold will produce an output file with the suffix 1 to 5. Finally, when the **k-fold 5/5**

is done, the five files are automatically joined in a single output file containing the downscaled whole calibration period. For the next steps, the user needs to change to **All testing**, so the software knows that evaluation metrics must be computed over the whole calibration period. On the other hand, when our methods have been evaluated and we are ready to generate downscaled projections using GCMs, it is recommended to train methods using as much data as possible. For this purpose (experiment PROJECTIONS), **All training** should be selected.

Bias Correction:
After the 'downscale' step, a bias correction can be performed (optional). In order to perform bias correction, the user needs to run the 'bias correction' step selecting **Yes** or **By season**. In this last case the bias correction is done for each season separately. The bias correction can be performed using different methods. Bias corrected outputs are stored in folders with suffixes 'BC' + method used for bias correction (+'s', if it has been done by season). Next steps ('Calculate climdex', 'Plot results' and 'Convert binary files to ASCII') will use bias corrected data. If the user wants to compare bias corrected data with original data, or even a regular bias correction with a bias correction by season, these steps can be run three times, one for each of the three different options for bias correction: No, Yes and By season.
Note that MOS downscaling methods and bias correction methods are the same. Applying a MOS method is equivalent to applying RAW-BIL with a posterior bias correction. Thus, if the user wants to apply a MOS method by seasons, the strategy would be to apply RAW-BIL, and the bias correct it selecting By season.

Seasons:
The user can define different seasons by indicating the season name (label) corresponding to each month. Thus, all months with the same label will form a season. These seasons are used for bias correction (if applied by season), for seasonal evaluation metrics and for seasonal future projected anomalies. Default seasons correspond to DJF, MAM, JJA and SON, but the year can be divided in for example two seasons: WET and DRY.

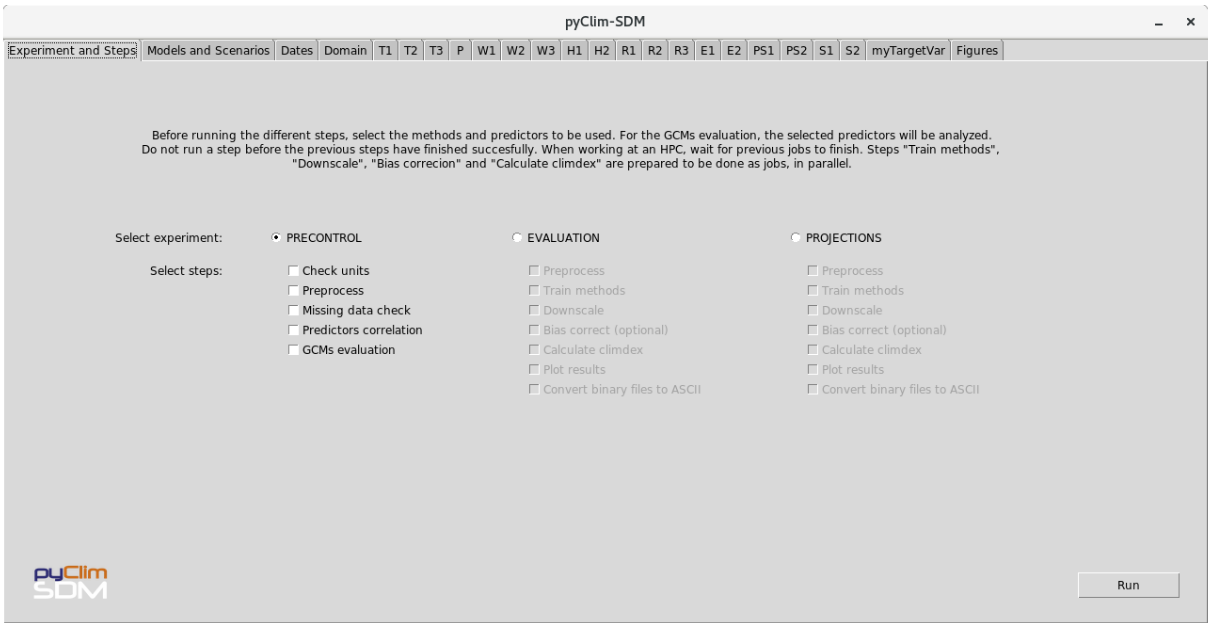
tab: Experiment and Steps

The three possible experiments are:

- PRECONTROL: to decide which predictors are relevant, evaluate GCMs or check for missing data.
- EVALUATION: to evaluate and intercompare the performance of different SDMs in the present climate.
- PROJECTIONS: to generate downscaled climate projections.

Each experiment contains different steps, which need to be run in order (except the ones indicated as optional).

PRECONTROL



Before running the different steps, select the methods and predictors to be used. For the GCMs evaluation, the selected predictors will be analyzed. Do not run a step before the previous steps have finished succesfully. When working at an HPC, wait for previous jobs to finish. Steps "Train methods", "Downscale", "Bias correcion" and "Calculate climdex" are prepared to be done as jobs, in parallel.

Select experiment: ● PRECONTROL ○ EVALUATION ○ PROJECTIONS

Select steps:

PRECONTROL: □ Check units □ Preprocess □ Missing data check □ Predictors correlation □ GCMs evaluation

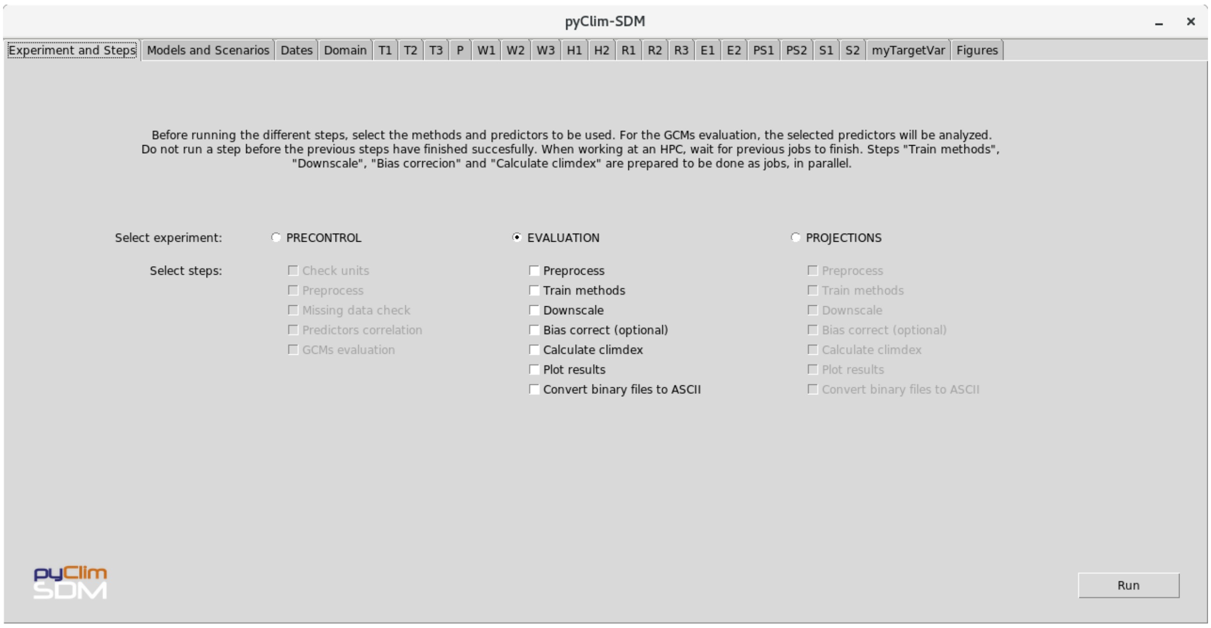
EVALUATION: □ Preprocess □ Train methods □ Downscale □ Bias correct (optional) □ Calculate climdex □ Plot results □ Convert binary files to ASCII

PROJECTIONS: □ Preprocess □ Train methods □ Downscale □ Bias correct (optional) □ Calculate climdex □ Plot results □ Convert binary files to ASCII

- **Check units**: this step checks units in reanalysis and GCMs netCDFs
- **Preprocess**: this step prepares data for the following steps, for example, by establishing an association between the low resolution grid and the high resolution target points, standardizing predictors or performing the train/test split.
- **Missing data check**: this step informs on the presence of missing data in GCM predictors. It can be used to discard predictors and/or GCMs.
- **Predictors correlations**: this step informs on the linear correlation coefficients between each target variable and all the selected predictors.
- **GCMs evaluation**: this step generates several figures helpful to evaluate GCMs representation of the present climate as well as to explore the signal of change given by each GCM and the spread among all selected GCMs. This step can be used to discard GCMs with a low skill in the present and to select a subset of GCMs representative of all future scenarios.

Bear in mind that for this experiment only the selected target variables and their predictors will be used, as well as the selected GCMs and scenarios. Usually, for this experiment the user should select a large set of predictors and GCMs, analyze the information (stored at /results/Figures) and then maybe reduce them for other experiments. The training/testing split and bias correction options (tab: Dates) are not used in this experiment.

EVALUATION



pyClim-SDM
Experiment and Steps | Models and Scenarios | Dates | Domain | T1 | T2 | T3 | P | W1 | W2 | W3 | H1 | H2 | R1 | R2 | R3 | E1 | E2 | PS1 | PS2 | S1 | S2 | myTargetVar | Figures
Run

Before running the different steps, select the methods and predictors to be used. For the GCMs evaluation, the selected predictors will be analyzed.
Do not run a step before the previous steps have finished succesfully. When working at an HPC, wait for previous jobs to finish. Steps "Train methods",
"Downscale", "Bias correcion" and "Calculate climdex" are prepared to be done as jobs, in parallel.

Select experiment: ○ PRECONTROL ● EVALUATION ○ PROJECTIONS

Select steps:

PRECONTROL:
□ Check units
□ Preprocess
□ Missing data check
□ Predictors correlation
□ GCMs evaluation

EVALUATION:
□ Preprocess
□ Train methods
□ Downscale
□ Bias correct (optional)
□ Calculate climdex
□ Plot results
□ Convert binary files to ASCII

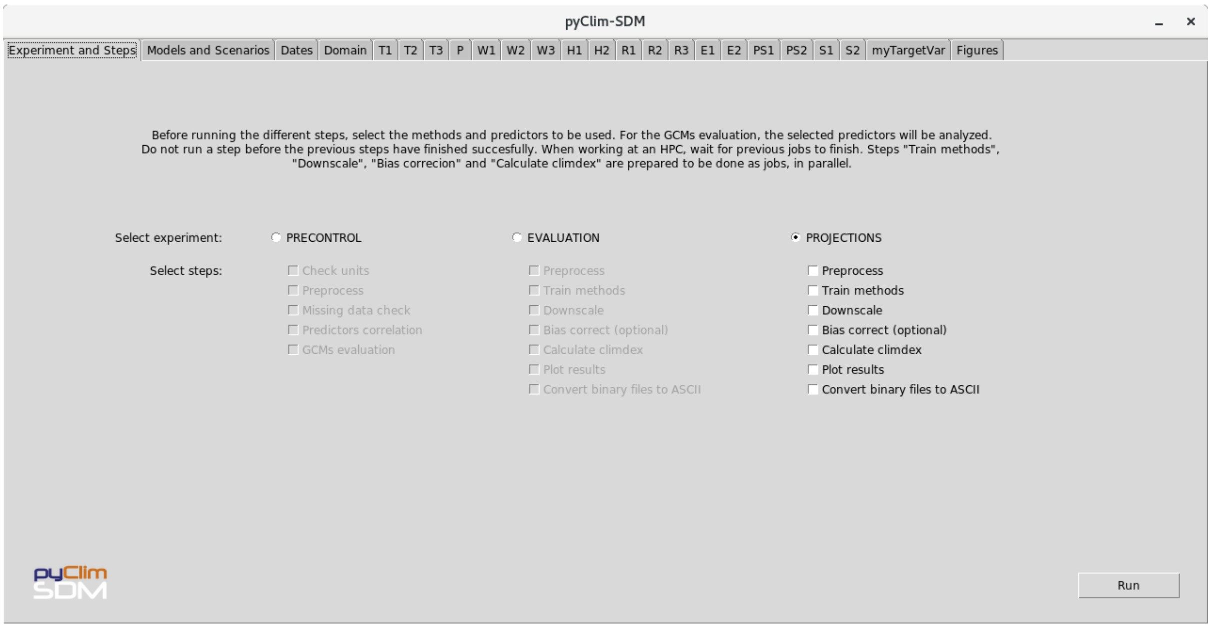
PROJECTIONS:
□ Preprocess
□ Train methods
□ Downscale
□ Bias correct (optional)
□ Calculate climdex
□ Plot results
□ Convert binary files to ASCII

- **Preprocess**: see PRECONTROL
- **Train methods**: this step trains the selected methods for the active target variables. When working at an HPC, each method will be processed in parallel by a different job. For the training, the part of the calibration period that has not been selected for testing will be used.
- **Downscale**: this step applies the trained methods over the testing period of the reanalysis. When working at an HPC, each method will be processed in parallel by a different job.

When following a k-fold approach, these first three steps need to be run five times and then select ‘All testing’ for the next steps (see tab: Dates).

- **Bias correction (optional)**: this step is optional and it corrects biases in the testing period using the bias correction method selected at tab Dates and using the whole year or differentiating by seasons. The user must be aware that, for bias correction, a long testing period is needed. Thus, when this option is used, the recommendation is to use the whole calibration period for testing following a k-fold approach. This step produces bias corrected daily data in different folders with suffixes ‘BC’ + method used for bias correction (+’s’, if it has been done by season). Now the user has downscaled outputs with and without correction, and the following steps will be applied over one or the other depending on the selection made at tab Dates for the Bias correction options. When working at an HPC, each method will be processed in parallel by a different job.
- **Calculate climdex:** this step computes derived climate indexes (mean values, percentiles, spells, etc) for the active target variables and the selected methods and climdex. When working at an HPC, each method will be processed in parallel by a different job.
- **Plot results**: this step produces a large set of figures with different evaluation metrics, for the active target variables and the selected methods and climdex. They are stored at /results/Figures (see Figures description)
- **Convert binary files to ASCII**: this step generates ASCII files from the original netCDFs, both for daily data and for climdex.

PROJECTIONS



Before running the different steps, select the methods and predictors to be used. For the GCMs evaluation, the selected predictors will be analyzed.
Do not run a step before the previous steps have finished succesfully. When working at an HPC, wait for previous jobs to finish. Steps "Train methods", "Downscale", "Bias correcion" and "Calculate climdex" are prepared to be done as jobs, in parallel.

Select experiment: ○ PRECONTROL ○ EVALUATION ● PROJECTIONS

Select steps:

PRECONTROL: □ Check units □ Preprocess □ Missing data check □ Predictors correlation □ GCMs evaluation

EVALUATION: □ Preprocess □ Train methods □ Downscale □ Bias correct (optional) □ Calculate climdex □ Plot results □ Convert binary files to ASCII

PROJECTIONS: □ Preprocess □ Train methods □ Downscale □ Bias correct (optional) □ Calculate climdex □ Plot results □ Convert binary files to ASCII

- **Preprocess**: see PRECONTROL.
- **Train methods**: see EVALUATION. For PROJECTIONS, it is recommended to train methods using the whole calibration period (All training at tab Dates)
- **Downscale**: see EVALUATION. For PROJECTIONS, the testing period is not used. Instead the methods are applied over the selected GCMs and scenarios. This step checks if a specific GCM and scenario has already been downscaled or not, which is configurable at advanced_settings (‘force_downscaling’, default False, i.e. skip), so existing GCMs/scenarios can be replaced or skipped. A maximum number of living jobs is set at advanced_settings (max_nJobs, default 5).
- **Bias correction (optional)**: see EVALUATION. For PROJECTIONS, the period used for bias correction is the reference period, defined at tab Dates. A maximum number of living jobs is set at advanced_settings (max_nJobs, default 5).
- **Calculate climdex:** see EVALUATION. For PROJECTIONS, it checks whether a climdex/GCM/scenario already exists, so it is skipped or replaced (configurable at advanced_settings, ‘force_climdex’, default False, i.e. skip). A maximum number of living jobs is set at advanced_settings (max_nJobs, default 5).
- **Plot results**: this step produces a large set of figures describing future projections in the form of evolution graphics and maps. It also generates figures named ‘evolTrengRaw’ which are extremely important to analyze which methods preserve trends given by GCMs. For this particular figure, climdex by RAW method and with no bias correction are mandatory, for each method is compared with them. All figures are stored at /results/Figures (see Figures description)
- **Convert binary files to ASCII**: see EVALUATION.

Methods

pyClim-SDM includes the following families of methods:

- Raw: no downscaling, interpolation.
- Model Output Statistics
- Analogs / Weather Typing
- Transfer Function: Linear and Machine Learning
- Weather Generators

Raw

- **RAW**: nearest neighbor interpolation. Each target point takes the value from the closest low resolution grid point.
- **RAW-BIL**: bilinear interpolation. Each target point takes the value from the four closest low resolution grid points, with a bilinear interpolation.

Model Output Statistics

- **QM**: empirical Quantile Mapping. Each target point takes the value of the low resolution grid bilinearly interpolated. Then an adjustment in the distributions is done using an empirical Quantile Mapping (Themeßl *et al.*, 2011).
- **DQM**: Detrended Quantile Mapping. Similar to QM but removing the long term trend before the adjustment (Cannon *et al.*, 2015).
- **QDM**: Quantile Delta Mapping. Similar to QM but instead of applying a bias adjustment over the distribution, it applies a delta change over the quantiles (Cannon *et al.*, 2015). An adjustment in the wet/dry frequency is also performed.
- **PSDM**: (Parametric) Scaled Distribution Mapping. Adjustment of quantiles using parametric distributions (normal/gamma), detrending series and treating explicitly the precipitation frequency (Switanek *et al.*, 2017).

Analogs / Weather Typing

- **ANA-SYN-1NN**: Nearest analog based on synoptic analogy. For each target day, a search of similar days in the past is done, and it takes the values of the observations corresponding to the closest analog. The similitude is established using the user defined Synoptic Analogy Fields, measured by the Euclidean distance. Previously these fields have been reduced to principal components preserving the 95% (configurable) of the variance.
- **ANA-SYN-kNN**: Similar to ANA-SYN-1NN but instead of taking values of the closest analog, it averages the k (configurable) closest analogs.
- **ANA-SYN-random**: Similar to ANA-SYN-kNN but instead of averaging the k closest analogs, it takes one of them randomly, having each analog an associated probability depending on the similitude.
- **ANA-LOC:** similar to ANA-SYN, but instead of using the synoptic analogy, it uses a combination of synoptic and local analogies. The local similitude, for each target point, is established by the Euclidean distance over the significant predictors. Significant predictors have been previously detected for each weather type and target point based on their correlation with the target variable. Weather types have been previously defined by a k-means clustering algorithm.

- **ANA-VAR**: similar to ANA-SYN but using the spatial pattern of the target variable itself instead of the Synoptic Analogy Fields.

Linear methods

- **MLR**: multiple linear regression. For each target point a multiple linear regression with bilinearly interpolated predictors is established. Not available for precipitation.
- **MLR-ANA**: multiple linear regression based on analogs. For each day and target point, a multiple linear regression is established, but using only predictors from analog days. Not available for precipitation.
- **MLR-WT**: multiple linear regression based on weather types. Similar to ANA-MLR but using precalibrated relationships for each weather type. Not available for precipitation.
- **GLM**: Generalized Linear Model. Only available for precipitation. The precipitation occurrence is modeled by a logistic regression, using 0.1 mm as threshold. Then, for wet days the intensity is modeled by a multiple linear (**LIN**) regression, with the possibility of performing an exponential (**EXP**) or cubic (**CUB**) transformation.

Machine Learning methods

- **SVM**: Support Vector Machine. Non-linear machine learning classification/regression.
- **LS-SVM**: Least Square Support Vector Machine. Non-linear machine learning classification/regression.
- **RF**: Random Forest. Non-linear machine learning classification/regression. This method is combined with a MLR to extrapolate to values out of the observed range (configurable).
- **XGB**: eXtreme Gradient Boost. Non-linear machine learning classification/regression. This method is combined with a MLR to extrapolate to values out of the observed range (configurable).
- **ANN**: Artificial Neural Networks. Non-linear machine learning classification/regression.
- **CNN**: Convolutional Neural Networks. Non-linear machine learning classification/regression.

Weather Generators

- **WG-PDF**: For each target point, monthly means/accumulations are calculated. Then, a linear regression between those statistics is performed. Finally, daily data for each month is randomly generated using a normal/exponential distribution conditioned on monthly statistics.
- **WG-NMM:** Only available for precipitation. For each target point, precipitation in low resolution is used to divide the time series in intensity intervals. Then, for each interval, the wet/dry transitions and intensities are calculated, in terms of probabilities. Finally, new data is generated, randomly, conditioned on the precipitation occurrence of the previous day and the probabilities given by the correspondent interval.

Parallel processing

Some steps can be run in serial or parallel processing, in different ways depending on whether pyClim-SDM is executed at a HPC cluster (with SLURM) or at a regular workstation with multiple CPUs (with multiprocessing).

Parallel processing with multiprocessing (regular workstation with multiple CPUs)

The user can define the number of CPUs at 'config/adevanced_settings.py', nCPUs_multiprocessing. If more than one CPU is used, steps 'downscale', 'bias correct' and 'calculate climdex' for experiment PROJECTIONS will be executed in parallel (each target variable/method/model/scene as a different process), as well as 'train methods' for Transfer Function methods (Linear and Machine Learning).

Parallel processing with SLURM (HPC cluster)

pyClim-SDM is designed to process some steps in parallel when executed at a HPC cluster managed with SLURM. The user has to indicate 'running_at_HPC' as True and 'HPC_partition' name at 'config/advanced_settings.py'.

The steps that can be run as jobs are:

- Train methods
- Downscale
- Bias correction
- Calculate climdex

Do not launch any step before jobs from the previous step have finished successfully.

When downscaling, bias correcting or calculating climdex of GCMs (experiment PROJECTIONS), usually too many jobs are launched. The maximum number of living jobs for these cases can be set at 'config/advanced_settings.py' with the parameter 'max_nJobs'.

Jobs are defined at 'lib/launch_jobs.py'. Different functions correspond to different steps and purposes. The user can tune them to set the number of parallel tasks (n) and memory (mem) for each job.

Machine Learning

Machine Learning methods have been designed and tested using several datasets in different regions. Nevertheless, for new datasets an advanced user might want to modify the original design. All Machine Learning models are in the 'train_point' function of 'lib/TF_lib.py'.

In order to tune these models, the user can set 'plot_hyperparameters_epochs_nEstimators_featureImportances' to True at 'config/advanced_settings.py'. This way some useful graphical information will be generated at 'aux/TRAINED_MODELS/' in order to help tuning Machine Learning models. This will be done only for a subsample of all target points, so the parameter must be set to False again after tuning.

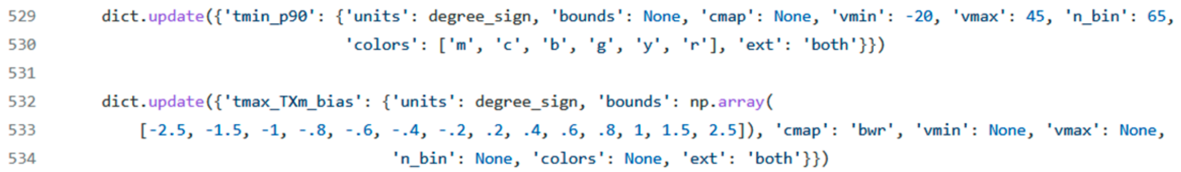
Beware that some Machine Learning models require a lot of disk storage.

Some Machine Learning models can perform wrong under extrapolation (i.e. when applied over predictors out of the training range) or even not being able to predict values out of the observed range. This last is the case of Random Forest and eXtreme Gradient Boost. The user can decide whether to replace any Machine Learning method by a Multiple Linear

Regression (for any target variable except for precipitation) at ‘config/advanced_settings.py’, with the parameter ‘methods_to_extrapolate_with_MLR’. Default methods are RF and XGB.

Plot

At ‘lib/plot.py’, function ‘map’, different color palettes are defined, and the user might want to modify them or build additional palettes. Two types of palettes are possible:



```
dict.update({'tmin_p90': {'units': degree_sign, 'bounds': None, 'cmap': None, 'vmin': -20, 'vmax': 45, 'n_bin': 65,
                          'colors': ['m', 'c', 'b', 'g', 'y', 'r'], 'ext': 'both'}})

dict.update({'tmax_TXm_bias': {'units': degree_sign, 'bounds': np.array(
    [-2.5, -1.5, -1, -.8, -.6, -.4, -.2, .2, .4, .6, .8, 1, 1.5, 2.5]), 'cmap': 'bwr', 'vmin': None, 'vmax': None,
                               'n_bin': None, 'colors': None, 'ext': 'both'}})
```

- defining min/max, the number of intervals, and a list of colors. This palette will produce regular intervals with a transition among colors.
- defining bounds (the specific intervals) and a cmap (color maps from pythons library matplotlib). With this option irregular intervals can be used.

Advanced settings

For advanced users, the config/advanced_settings.py file contains several parameters (explained at the file itself) that can be modified:

- interp_mode: predictors are trake from the nearest grid point or interpolated bilinearly from the four neighbors.
- plot_hyperparameters_epochs_nEstimators_featureImportances: if set to True, figures will be generated for machine learning hyperparameter tuning, and only some points (1 out of 500) will be trained. Once the machine learning hyperparameters have been modified, set to False so all points are trained.
- mean_and_std_from_GCM: predictors are standardized using the mean and std by reanalysis (if set to False) or by the same GCM (if set to True)
- force_downscaling: when downscaling GCMs, if set to False only GCMs not downscaled yet will be downscaled. If set to True, already dowscaled GCMs will be downscaled again anyway.
- force_climdex_calculation: same as force_downscaling but for climdex
- force_bias_correction : same as force_downscaling but for bias correction
- recalibrating_when_missing_preds: when downscaling each target point at Transfer Function methods for GCMs, if a predictor contains NaNs (missing data), two different strategies can be adopted: (True) recalibrating the method only with complete predictors and (False) setting to NaN days and points with at least one missing predictor.
- methods_colors: asigns one color to each method
- methods_linestyles asigns one linestyle to each method

Figures description

Figures generated by pyClim-SDM are listed and explained below:

id	experiment	figType	var	climdex/pred	method/model/scene	season
1	PRECONTROL	correlationMap	$var	$pred	None	$season
2	PRECONTROL	correlationBoxplot	$var	None	None	$season
3	PRECONTROL	nansMap	$var	$pred	$model-$scene	None
4	PRECONTROL	nansMatrix	$var	None	$scene	$season
5	PRECONTROL	biasBoxplot	$var	$pred	$scene	$season
6	PRECONTROL	biasMap	$var	$pred	$model-$scene	$season
7	PRECONTROL	evolSpaghetti	$var	$pred	$scene	$season
8	PRECONTROL	qqPlot	$var	$pred	None	$season
9	PRECONTROL	annualCycle	$var	$pred	$scene	None
10	PRECONTROL	changeMap	$var	$pred	$method-$scene-$years	$season
11	EVALUATION	annualCycle	$var	None	all	None
12	EVALUATION	rmseBoxplot	$var	None	all	$season
13	EVALUATION	correlationBoxplot	$var	None	all	$season
14	EVALUATION	varianceBoxplot	$var	None	all	$season
15	EVALUATION	spatialCorrBoxplot	$var	None	all	$season
16	EVALUATION	qqPlot	$var	None	$method	$season
17	EVALUATION	r2Map	$var	None	$method	$season
18	EVALUATION	rmseMap	$var	None	$method	$season
19	EVALUATION	accuracyMap	$var	None	$method	$season
20	EVALUATION	correlationMapMonthly	$var	None	$method	None
21	EVALUATION	r2MapMonthly	$var	None	$method	None
22	EVALUATION	correlationBoxplot Monthly	$var	None	$method	None
23	EVALUATION	r2BoxplotMonthly	$var	None	$method	None
24	EVALUATION	biasClimdexBoxplot	$var	$climdex	$method	$season
25	EVALUATION	TaylorDiagram	$var	$climdex	all	$season

26	EVALUATION	obsMap	$var	$climdex	None	$season
27	EVALUATION	estMap	$var	$climdex	$method	$season
28	EVALUATION	biasMap	$var	$climdex	$method	$season
29	EVALUATION	scatterPlot	$var	$climdex	$method	$season
30	PROJECTIONS	evolSpaghetti	$var	$climdex	$method	$season
31	PROJECTIONS	evolTube	$var	$climdex	$method	$season
32	PROJECTIONS	meanChangeMap	$var	$climdex	$method-$scene-$years	$season
33	PROJECTIONS	spreadChangeMap	$var	$climdex	$method-$scene-$years	$season
34	PROJECTIONS	evolTrendRaw	$var	$climdex	$method-$scene	$season

1. Correlation of the temporal daily series between one predictor and one predictand (Pearson coefficient for tmax/tmin and Spearman for pcp).
2. Correlation of the temporal daily series between all predictors and one predictand (Pearson coefficient for tmax/tmin and Spearman for pcp). Each box contains one value per grid point.
3. Map with percentage of missing data for one predictor, model and scene.
4. Percentage of missing data (spatially averaged) for one scene (all predictors and models).
5. Bias of all models compared to the reanalysis (in the mean value) in a historical period. For tmax/tmin absolute bias, for pcp relative bias and the rest standardized and absolute bias. Each box contains one value per grid point.
6. Bias of one model compared to the reanalysis (in the mean value) in a historical period. For tmax/tmin absolute bias, for pcp relative bias and the rest standardized and absolute bias.
7. Evolution of one predictor by all models in the form of anomaly with respect to the reference period (absolute anomaly for tmax/tmin, relative anomaly for pcp and absolute anomaly of the standardized variables for the rest).
8. QQ-plot for one variable by one model in historical vs. reanalysis.
9. Annual cycle for one variable by all models in historical and reanalysis (monthly means for tmax/tmin, monthly accumulations for pcp and monthly means of the standardized variable for the rest).
10. Change (abs/relative) in the mean value over a 30-year period by the middle and end of the century compared to the reference period
11. Annual cycle for one variable, downscaled by all methods vs. observation (monthly means for tmax/tmin and monthly accumulations for pcp).
12. RMSE of the daily series (downscaled and observed) by all methods. Boxes contain one value per grid point.
13. Correlation (Pearson for temperature and Spearman for precipitation) of the daily series (downscaled and observed) by all methods. Boxes contain one value per grid point.

14. Bias (relative, %) in the variance of the daily series (downscaled and observed) by all methods. Boxes contain one value per grid point.
15. Spatial correlation of the daily maps. Each box contains one value per day.
16. QQ-plot for one variable by one method vs. observations.
17. R2 score of the daily series (coefficient of determination) by one method.
18. RMSE of the daily series by one method.
19. Accuracy score for the daily series (only for wet/dry classification. Acc=corrects/total) by one method.
20. Correlation for the monthly (mean for tmax/tmin and accumulated for pcp) series by one method with observations. Pearson coefficient for tmax/tmin and Spearman for pcp.
21. R2 score (coefficient of determination) for the monthly (mean for tmax/tmin and accumulated for pcp) series by one method with observations.
22. Correlation for the monthly (mean for tmax/tmin and accumulated for pcp) series by one method with observations.
23. R2 score (coefficient of determination) for the monthly (mean for tmax/tmin and accumulated for pcp) series by one method with observations.
24. Bias (absolute/relative) for the mean climdex in the whole testing period by all methods. Boxes contain one value per grid point.
25. Taylor Diagram of the spatial distribution for the mean climdex in the whole testing period by all methods.
26. Mean observed values in the whole period.
27. Mean estimated (downscaled) values in the whole period by one method.
28. Bias (absolute/relative) in the whole period by one method.
29. Downscaled vs. observed climdex in the whole period each scatter point corresponds to a grid point.
30. Evolution of one variable by all models in the form of anomaly with respect to the reference period (absolute anomaly for tmax/tmin and relative anomaly for pcp).
31. Evolution graph of one variable by the multimodel ensemble (the central line represents 50th percentile and the shaded area represents IQR), in the form of anomaly with respect to the reference period (absolute anomaly for tmax/tmin and relative anomaly for pcp).
32. Anomaly in a future period with respect to a reference period given by the multimodel ensemble median (mean change). Absolute anomaly for tmax/tmin and relative anomaly for pcp.
33. Multimodel ensemble spread in the anomaly given by the difference between the 75th and 25th percentiles..
34. Evolution graph, by one method vs. raw models, of one variable by the multimodel ensemble (the central line represents 50th percentile and the shaded area represents IQR), in the form of anomaly with respect to the reference period (absolute anomaly for tmax/tmin and relative anomaly for pcp).