
sterope Documentation

Rodrigo Santibáñez

Oct 09, 2020

Contents

1	Installation	3
1.1	Option 1: Install sterope natively on your computer	3
1.2	Option 2: Clone the Github repository	4
2	Global Sensitivity Analysis	5
3	Compiling Python3 from source	7
4	Installing SLURM in your machine	9
5	Indices and tables	11

Sterope is a python3 package that implement a method based on the Dynamic Influence Network (<https://www.ncbi.nlm.nih.gov/pubmed/28866584>) to analyze the sensitivity of parameter values in the response of a Rule-Based Model written in kappa (<https://kappalanguage.org/>)

Sterope creates models samples and analyze the Dynamic Influence Network employing the Sobol method included in the SALib python package ([SALibpaper](#)). After samples are created, Sterope simulates them in parallel employing SLURM ([SLURM](#)) or the python multiprocessing API.

Sterope is part of the plan to add methods into Pleiades (<https://github.com/glucksfall/pleiades>) includes a parameterization employing a Genetic Algorithm and other analysis methods that are typical of modeling frameworks like Ordinary Differential Equations.

You could write us if you wish to add methods into pleione or aid in the development of them.

There are two different ways to install sterope:

1. **Install sterope natively (Recommended).**

OR

2. **Clone the Github repository.** If you are familiar with git, sterope can be cloned and the respective folder added to the python path. Further details are below.

Note: Need Help? If you run into any problems with installation, please visit our chat room: <https://gitter.im/glucksfall/pleiades>

1.1 Option 1: Install sterope natively on your computer

The recommended approach is to use system tools, or install them if necessary. To install python packages, you could use pip, or download the package from [python package index](#).

1. **Install with system tools**

With pip, you simple need to execute and sterope will be installed on `$HOME/.local/lib/python3.6/site-packages` folder or similar.

```
pip3 install sterope --user
```

If you have system rights, you could install sterope for all users with

```
sudo -H pip3 install sterope
```

2. **Download from python package index**

Alternatively, you could download the package (useful when pip fails to download the package) and then install with pip. For instance:

```
wget https://files.pythonhosted.org/packages/01/db/
↳582f9d3c34c840117e8fd19c9de4d9745508718765e48a808427005f16dd/sterope-1.5-py3-
↳none-any.whl
pip3 install sterope-1.5-py3-none-any.whl --user
```

Note: Why Python3?: Pleione is intended to be used with python3, despite the lack of incompatible functions with python2, because the latter won't receive further development past 2020.

Note: pip, Python and Anaconda: Be aware which pip you invoke. You could install pip3 with `sudo apt-get install python3-pip` if you have system rights, or install python3 from source, and adding `<python3 path>/bin/pip3` to the path, or linking it in a directory like `$HOME/bin` which is commonly added to the path at login. Also be aware that, if you installed Anaconda, pip could be linked to the Anaconda specific version of pip, which will install pleione into Anaconda's installation folder. Type `which pip` or `which pip3` to find out the source of pip, and type `python -m site` or `python3 -m site` to find out where is more likely pleione will be installed.

1.2 Option 2: Clone the Github repository

1. Clone with git

The source code is uploaded and maintained through Github at <https://github.com/glucksfall/sterope>. Therefore, you could clone the repository locally, and then add the folder to the PYTHONPATH. Beware that you should install the *salib*, *pandas*, *dask*, and *dask-jobqueue* packages by any means.

```
git clone https://github.com/glucksfall/sterope /opt
echo export PYTHONPATH="\$PYTHONPATH:/opt/sterope" >> $HOME/.profile
```

Note: Adding the path to `$HOME/.profile` allows python to find the package installation folder after each user login. Similarly, adding the path to `$HOME/.bashrc` allows python to find the package after each terminal invocation. Other options include setting the PYTHONPATH environmental variable in a sh file (see the example folder) or invoke `python3 setup.py clean build install` to install Pleione as it was downloaded from the PyPI server.

Global Sensitivity Analysis

1. Prepare the model

Sterope finds which variables will be analyzed using the symbol `//` (doble slash, as C/C++) followed by:

- Type of bound: `range` or `factor`
- Bounds for each parameter: `[min max]`. In the case of `factor`, the original parameter value is multiply by $(1 - \text{min})$ and $(1 + \text{max})$, therefor, `min` and `max` should by between zero and one.

For instace, the Thomas' model was configured as:

```
%var: 'KD1__FREE__' 1.000000e+00 // range[0.01 100]
%var: 'km1__FREE__' 1.000000e+00 // range[0.01 100]
%var: 'K2RT__FREE__' 1.000000e+00 // range[0.01 100]
%var: 'km2__FREE__' 1.000000e+00 // range[0.01 100]
%var: 'kphos__FREE__' 1.000000e+00 // range[0.01 100]
%var: 'kdephos__FREE__' 1.000000e+00 // range[0.01 100]
```

or the following configuration if the model is written in syntax 3:

```
%var: 'KD1__FREE__' 1.000000e+00 # range[0.01 100]
%var: 'km1__FREE__' 1.000000e+00 # range[0.01 100]
%var: 'K2RT__FREE__' 1.000000e+00 # range[0.01 100]
%var: 'km2__FREE__' 1.000000e+00 # range[0.01 100]
%var: 'kphos__FREE__' 1.000000e+00 # range[0.01 100]
%var: 'kdephos__FREE__' 1.000000e+00 # range[0.01 100]
```

Note: Sterope is compatible with KaSim4 only. The software could simulate *kappa* models written in syntax 3 (See below option syntax).

2. Configure Sterope

Write a `script.sh` file with the following, configured with a number of processes that will not interfere with other user's plans to lunch jobs in a cluster. Also and only if the job is launched outside

SLURM (see below), the number of processes could not be greater than the number of cores the node (your laptop, desktop, or server) has.

```
#!/bin/sh

#SBATCH --no-requeue
#SBATCH --partition=cpu

#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1

#SBATCH --job-name=sterope
#SBATCH --output=stdout.txt
#SBATCH --error=stderr.txt

export PYTHONPATH="$PYTHONPATH:$HOME/opt/sterope"

MODEL=pysbmodel-example6-kasim.kappa
FINAL=660
STEPS=10 # KaSim4 interprets as the period, not how many points to report!

NUM_LEVELS=1000

python3 -m sterope.kasim --model=$MODEL --final=$FINAL --tmin=600 --steps=$STEPS \
--grid=$NUM_LEVELS --kasim=kasim4 --syntax 3 --method sobol --nprocs 250 --memory_
↪2000MB
```

3. Execute Sterope

```
sbatch script.sh
```

OR

```
nohup sh script.sh &
```

Note: The Dynamic Influence Network changes over the course of a simulation. To report time window smaller than the simulation time, set `--tmin float` as the starting point and `--tmax float` as the finish point. The `--tmin` default is zero and `--tmax` default is `--final`.

Note: **Sensitivity methods** type `python3 -m sterope.kasim --help` to find out the available methods. Sobol is the default and we implemented all the available in the SALib python package.

Note: `--memory` is not necessary when running sterope without SLURM. Also, the `#SBATCH` directives are omitted when executing the script with `sh` so they are safe to leave.

CHAPTER 3

Compiling Python3 from source

If you don't have admin access to the cluster configuration, you could compile and install python3 from source following these instructions:

```
wget https://www.python.org/ftp/python/3.6.5/Python-3.6.5.tgz \
-O ~/opt/ubuntu-software/Python-3.6.5.tgz
if [ -d ~/opt/Python-3.6.5 ]; then rm -rf ~/opt/Python-3.6.5; fi
tar xvfz ~/opt/ubuntu-software/Python-3.6.5.tgz -C ~/opt
cd ~/opt/Python-3.6.5
if [ -f Makefile ]; then make clean; fi
if [ -d $(HOME)/opt/python-3.6.5 ]; then rm -rf $(HOME)/opt/python-3.6.5; fi
./configure --prefix=$(HOME)/opt/python-3.6.5
make
make install
```

Note: Don't copy an installation folder from another machine since there may be libraries incompatibilities. Instead, the code will download, configure, compile, and install. To make accesible from anywhere, you could add an alias into ~/.bashrc or a symbolic in your \$HOME/bin folder for \$HOME/opt/python-3.6.5/bin/python3 and pip3.

To install numpy and pandas use the following instructions, in order since some pandas dependencies has also dependencies:

```
wget https://files.pythonhosted.org/packages/71/90/
↳ ca61e203e0080a8cef7ac21eca199829fa8d997f7c4da3e985b49d0a107d/numpy-1.14.3-cp36-
↳ cp36m-manylinux1_x86_64.whl
wget https://files.pythonhosted.org/packages/dc/83/
↳ 15f7833b70d3e067ca91467ca245bae0f6fe56ddc7451aa0dc5606b120f2/pytz-2018.4-py2.py3-
↳ none-any.whl
wget https://files.pythonhosted.org/packages/67/4b/
↳ 141a581104b1f6397bfa78ac9d43d8ad29a7ca43ea90a2d863fe3056e86a/six-1.11.0-py2.py3-
↳ none-any.whl
wget https://files.pythonhosted.org/packages/cf/f5/
↳ af2b09c957ace60defae112b669c45e8e97e32f94aa8b56da4c6d1682825/python_dateutil-2.7.3-
↳ py2.py3-none-any.whl
```

(continues on next page)

(continued from previous page)

```
wget https://files.pythonhosted.org/packages/69/ec/
↪8ff0800b8594691759b78a42ccd616f81e7099ee47b167eb9bbd502c02b9/pandas-0.23.0-cp36-
↪cp36m-manylinux1_x86_64.whl

pip3 install numpy-1.14.3-cp36-cp36m-manylinux1_x86_64.whl
pip3 install pytz-2018.4-py2.py3-none-any.whl
pip3 install six-1.11.0-py2.py3-none-any.whl
pip3 install python_dateutil-2.7.3-py2.py3-none-any.whl
pip3 install pandas-0.23.0-cp36-cp36m-manylinux1_x86_64.whl
```

If you have admin access (and willing to compile python3 from source) you could install the following dependencies:

```
apt-get install libssl-dev zlib1g-dev libncurses5-dev \
libncursesw5-dev libreadline-dev libsqlite3-dev libgdbm-dev \
libdb5.3-dev libbz2-dev libexpat1-dev liblzma-dev tk-dev
```

Compiling python3 with all dependencies would make installation of packages easier. Just follow the instructions:

```
pip3 install pandas
```

Note: Installing pandas with pip will install numpy as its dependency.

Note: Be sure you are calling pip3 after creating an alias or a symbolic link. Without admin credentials, pip3 would fail to install pandas.

CHAPTER 4

Installing SLURM in your machine

To install SLURM you need admin access to the machine. Please follow this instructions to start up running the workload manager, in the controller as well in the controlled machines.

```
sudo apt-get -y install slurm-wlm
sudo nano /etc/slurm-llnl/slurm.conf

sudo chown -R slurm:slurm /var/run/slurm-llnl/
sudo chown -R slurm:slurm /var/lib/slurm-llnl/
sudo chown -R slurm:slurm /var/log/slurm-llnl/
sudo mkdir /var/spool/slurmd
sudo chown -R slurm:slurm /var/spool/slurmd

sudo systemctl start slurmd
```

Replace `$HOST_NAME` with your machine name that is going to act as the controller. If you have multiple machines, this configuration file must be identical and in all machines in the queue. Be aware that Dask requires configuration of memory to be able to configure a SLURMCluster (More info in <https://jobqueue.dask.org/en/latest/examples.html>)

```
### slurm.conf - Slurm config file.

#ClusterName=$HOST_NAME
ControlMachine=$HOST_NAME
SlurmUser=slurm
AuthType=auth/munge

SlurmctldPidFile=/var/run/slurm-llnl/slurmctld.pid
SlurmdPidFile=/var/run/slurm-llnl/slurmd.pid
SlurmdSpoolDir=/var/lib/slurm-llnl/slurmd
StateSaveLocation=/var/lib/slurm-llnl/slurmctld

SwitchType=switch/none
ProctrackType=proctrack/pgid
TaskPlugin=task/none
```

(continues on next page)

(continued from previous page)

```
MpiDefault=none
MaxJobCount=100000
MaxArraySize=64000

# TIMERS
SlurmdTimeout=300
InactiveLimit=0
MinJobAge=300
KillWait=30
Waittime=0

# SCHEDULING
DefMemPerCPU=18400
SchedulerType=sched/backfill
SelectType=select/cons_res
SelectTypeParameters=CR_CPU_Memory
FastSchedule=1

# LOGGING
SlurmctldDebug=3
SlurmctldLogFile=/var/log/slurm-llnl/slurmctld.log
SlurmdDebug=3
SlurmdLogFile=/var/log/slurm-llnl/slurmd.log

# COMPUTE NODES

# Here you add the machine hardware configurations
NodeName=$HOST_NAME Procs=56 TreadsPerCore=1 State=idle RealMemory=1031755

# Here you add the machine(s) to a Partition
PartitionName=MyCluster Nodes=$HOST_NAME Default=yes MaxTime=INFINITE State=up
```

Note: Please refer to [SLURM](#) for advance configuration like limiting time, CPUs and RAM for users or groups, to balance load in your cluster.

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`