SLM GUI Manual

1. Introduction:

the SLM algorithm aims to detect and analyze meaningful figure of merit in time-series data. The attached modification of the algorithm comes with a user interface application, allowing any user to adjust the algorithm to his needs without any prior knowledge regarding the algorithm or programming in general. The GUI is designed to be simple and easy to use, handling various data types, and adjusting computational and functional options of the algorithm, such as uploading options, down-sample factor, and number of targets, to the user needs. In this manual, all the information needed to operate the GUI is provided.

2. System Requirements:

2.1. Python Version:

2.1.1. Python 3.8 or higher is required.

2.2. Python Libraries:

2.2.1. Ensure the following Python libraries are installed:

- NumPy
- SciPy
- Pandas
- Matplotlib
- Rbeast
- Scikit-learn (Sklearn)
- time
- datetime
- os
- logging
- Kivy

3. Getting Started:

3.1. Access the GitHub Respiratory at:

https://github.com/OkTAU16/SA_UI/tree/main

3.2. Opening the GUI:

3.2.1. Via Python file – “SLM_UI.py”

3.2.2. Via .exe file – “SLM_UI.exe” which is located under the ‘dist’ folder

3.3. When opened, the introduction screen will appear

3.4. The GUI includes three screens:

3.4.1. Introduction screen – provides basic information about data formatting and the algorithm.

3.4.2. Main screen – actively modified by the user to use the algorithm.

3.4.3. Results screen.

4. Navigating the GUI:

4.1. Navigation between the screens can be done using the ‘next’ and ‘back’ at the bottom of each screen.

4.2. **Introduction screen:**

4.2.1. Includes one section – text of basic information.

4.2.2. Basic control:

4.2.2.1. Button - Click on it to navigate to the next screen

4.2.3. **Main screen:**

4.2.4. Sections:

4.2.4.1. Dropping section – allows to drag and drop file/folder to the algorithm.

4.2.4.2. Fine-tuning section – control the hyperparameters of the algorithm.

4.2.4.3. Data & warning section.

4.2.5. Basic control:

4.2.5.1. Drag & drop – drag data to the marked area

4.2.5.2. Radio button – choose between two options.

4.2.5.3. List button - choose between a set of options.

4.2.5.4. Text box - Type in these to enter data.

4.2.5.5. Buttons - Click on these to perform actions.

5. data Formatting:

5.1. Data must be uploaded as a **CSV, mat, or Excel file.**

5.2. data can be uploaded in a single file or a folder.

5.3. Each file must be organized as follows:

5.4. First column – **time series** (optional): include the time stamps of each sample. Must be included for irregularly spaced time series data.

5.5. Second column – **total energy:** the observable variable in the experiment.

5.6. **Distance** columns – a figure of distance/loss from the final target. When a final target is reached, the distance must be zero (by default) or the threshold defined by the user. **Each target must have a separate distance column**.

5.7. **Note –** if Distance columns are not provided, the threshold will be defined for **total energy. This case is possible in single-target experiments only** and does not support multiple-target experiments**.** In experiments with multiple targets please give a distance column for each target

6. **Output:**

6.1. Three evaluation graphs

6.2. The stochastic landscape, generated by the algorithm.

7. UI's workflow:

7.1. Drag & drop the data in the marked area. Make sure the data is well formatted and that no warning message appears on the screen.

7.2. If the data is set in a folder, make sure all the files are detected.

7.3. Use the radio button to define the need for a time-series, down-sample, and cross-validation. If down-sample and/or cross-validation is needed, please fill in the text box and press the "✓" button.

7.4. Use the radio button to define the existence of distance columns and fill the text box with the wanted threshold for the distance/energy columns accordingly. **Reminder: lack of distance column is possible only in single-target-experiments**

7.5. Choose the number of particle clusters using the list button.

7.6. Fill in the target number and output path field and press the "✓" button. Notice that all fields' backgrounds must turn green before you continue.

7.7. Press the "test" button and check for approval/error messages.

7.8. Press the "submit" button to start the algorithm. If .mat files were used, please fill in the needed information in the pop-up window.

7.9. While the algorithm is running, a loading pop-up window will show. **Do not close the window**.

7.10. When the "complete" pop-up is showing, you may continue to the results screen.

7.11. Click on the graphs buttons the see them in a pop-up window. All graphs. Along the stochastic landscape are saved in the output path provided.

8. Example data:

8.1. The example data includes .mat files and is in the GitHub respiratory, in the 'example_data' folder: https://github.com/OkTAU16/SA_UI/tree/main/example_data

8.2. To run the example, please download the folder to your local device.

8.3. Drag & drop the data in the marked area.

8.4. Time series and down-sampling are not needed for this. Therefore, select the 'No' (red) radio button for both.

8.5. Set the Cross-Validation factor to 10 by choosing the 'Yes' (green) radio button under the 'CV' title, typing '10' in the text box, and pressing the "✓" button.

8.6. set the number of clusters to 3 by pressing the list button.

8.7. Set the number of targets to 2.

8.8. enter the output directory path of your choice.

8.9. Press the 'test' button to make sure all fields are filed correctly.

8.10. Press the 'submit' button.

8.11. The variable name for the .mat files is 'energy_distance'. Type it in the pop-up and press 'submit'.

9. Troubleshooting:

9.1. **GUI won't open:**

9.1.1. Make sure Python 3.8 or higher is installed.

9.1.2. Make sure 'kivy' library is installed.

9.2. **GUI crushes:**

9.2.1. Make sure all required libraries are installed.

9.2.2. Check data formatting.

9.2.3. make sure the energy/distance threshold is set correctly.

9.2.4. For multiple-target experiments – make sure that a distance vector is added for each target.

9.3. **NaN result:**

9.3.1. Make sure enough data is uploaded. Lack of data may lead to a major NaN percentage in the stochastic landscape

9.4. **Unable to "submit" the data:**

9.4.1. make sure all options are chosen

9.4.2. Make sure all text boxes are submitted and have a green background.

9.4.3. Check warning messages for further information.

9.5. **stuck main screen:**

9.5.1. Increase the **down-sample factor.**

10. **Contact Information:**

10.1. This code was written by Omri Kovarsky and Idan Frenkel.

10.2. Need Help? Contact us at omrikovarsky@mail.tau.ac.il for assistance.