Cornell University

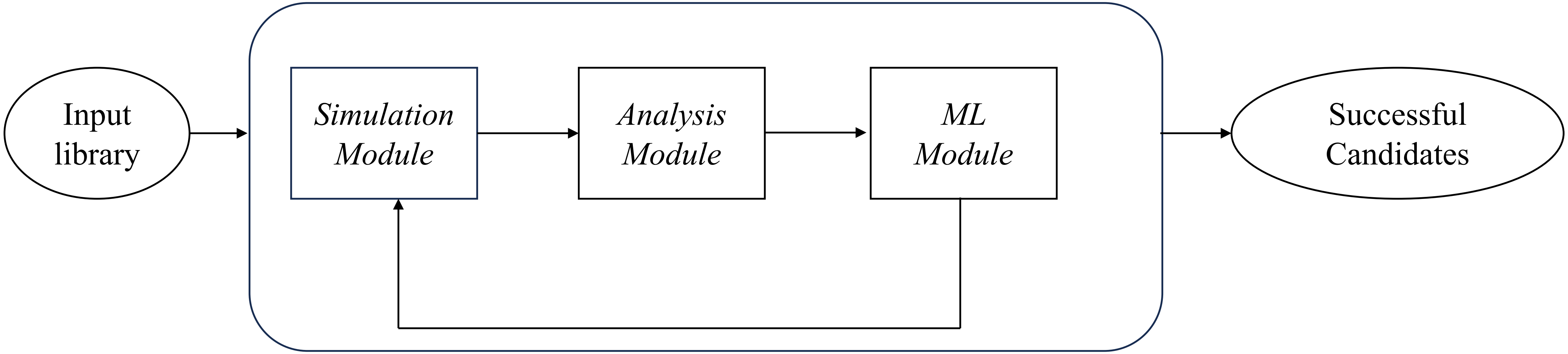
Tutorial on automated dataflow package named Model Parameter Targeted Search (MPTS)

Vikram Thapar
Prof. Escobedo Group

Glossary

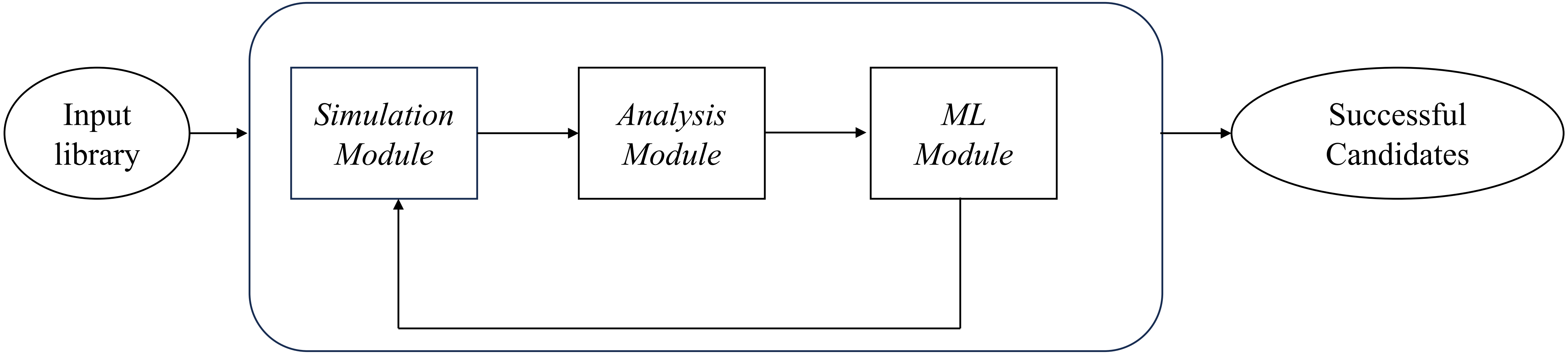
Introduction

- A design library of candidates in the user defined K-dimensional model parameter space $\{p_1,p_2,\ldots p_K\}$.
- Ground truth value of output of the model for any given candidate is obtained by simulating the model. (y_i)
- MPTS is python-based package written to apply a framework which searches through library to find candidates that have the highest y_i with the goal of running minimum number of simulations.



MPTS Flow Chart

Introduction



An iterative framework composed of three user written modules named *Simulation Module*, *Analysis Module*, *ML module*.

1. Simulations of the current proposed batch are performed in the *Simulation Module*.

2. Once the simulations are completed, the data from simulations are fed into the *Analysis Module* which calculate the ground truth output value (y_i).

3. The dataset of parameters values as input and output as y_i is added to the combined dataset of previous proposed batches to build the training dataset.

4. The training dataset is fed into *ML module* to build the surrogate model.

5. The optimization strategy in *ML module* uses the surrogate model to propose the next batch of candidates.

6. Steps 1-5 are repeated till the maximum number of framework iterations are performed.

How to install

- Clone the repository

  1. git clone https://github.com/vt87/MPTS.git

- Add the following in ~/.bashrc

  2. export PATH=$PATH:{***PATH***}

  PATH is where git directory is cloned.

- Source the bashrc using the following command

  3. source ~/.bashrc

- *Example*

```
cd ~/test

git clone https://github.com/vt87/MPTS.git

export PATH=$PATH:~/test

source ~/.bashrc
```

How to run

```
MPTSrun.py –y input.yaml
```

input.yaml : Input file of yaml format. A common format used to make input files.

Refer the following for a short tutorial on yaml formats.
https://www.cloudbees.com/blog/yaml-tutorial-everything-you-need-get-started

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # # directory where module input files are stored
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

Python dictionary of name mpts is initialized *(DO NOT CHANGE)*

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

Name of Input library file containing the required data.

Format in which data should be written is discussed in slides 31-32.

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

Directory where library file is located.

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

Directory where output of MPTS is saved.
Output of MPTS is discussed in slide 34.

input.yaml

Total number of iterations for which main function of MPTS is executed

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee       # (example) optimization technique in ml module input file
```

input.yaml

Sleep time in seconds between adjacent iterations of MPTS.
The purpose of this variable is to pause the MPTS while simulation data is being generated.

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

input.yaml

Name of library file containing the data for the initial batch.
Should be located in inp_path.
Format in which data is written is like the input library file.
More is discussed in slide 33.

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input director
 out_path : ./out          # output director
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

Number of next best candidates to be proposed by machine learning.

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

Number of framework iterations.
It is the number of times next best batches are proposed.

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

Directory where user defined module files are located.

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files are stored
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

Directory where input files of user defined modules are located.

input.yaml

```
mpts:                      # mpts dictionary
 libname : lib.txt          # input library file
 inp_path : ./inp           # input directory
 out_path : ./out           # output directory
 niter : 100                # number of iterations for which main of MPTS is executed
 sleeptime : 60             # sleeptime between MPTS iterations.
 initfile : initlib.txt     # library file for initial batch
 nextsize : 20              # next best candidate batch size
 mliter : 5                # number of framework iterations.
 module_path : ./mods       # directory where module files are stored
 module_inppath : ./minps   # directory where module input files directory
 module_sim :               # simulation module dictionary
  name : sim                 # simulation module name
  inpname : siminp.yaml      # simulation module input file
  vars :                     # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims          # (example) simulation data storage directory in simulation module input file
 module_anlys :             # analysis module dictionary
  name : anlys               # analysis module name
  inpname : anlysinp.yaml    #  analysis module input file
  vars :                     # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6                # (example) optype  in analysis module input file
 module_ml :                # ml module dictionary
  name : ml                  # ml module name directory
  inpname : mlinp.yaml       # ml module input file name
   vars :                    # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee       # (example) optimization technique in ml module input file
```

Python dictionary of name module_sim is initialized *(DO NOT CHANGE)*

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files are stored
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

INSIDE THE module_sim DICTIONARY:
Name of the simulation module.
<u>sim.py should exist in $module_path</u>

How to write a simulation module is described in slides 37-39.

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between M
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

INSIDE THE module_sim DICTIONARY:
Input file to the simulation module.
siminp.yaml should exist in $module_inppath

This contains the information about variables required to run model simulations. E.g., temperature and pressure.

How to write a simulation input module file is described in slide 36.

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

INSIDE THE module_sim DICTIONARY:

An optional dictionary of name vars.

The purpose of this dictionary is to specify additional inputs to the simulation module.

Example : sim_path : Directory where data from simulations are stored.

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

Python dictionary of name module_anlys is initialized *(DO NOT CHANGE)*

input.yaml

```
mpts:                         # mpts dictionary
 libname : lib.txt             # input library file
 inp_path : ./inp              # input directory
 out_path : ./out              # output directory
 niter : 100                   # number of iterations for which main of MPTS is executed
 sleeptime : 60                # sleeptime between MPTS iterations.
 initfile : initlib.txt        # library file for initial batch
 nextsize : 20                 # next best candidate batch size
 mliter : 5                   # number of framework iterations.
 module_path : ./mods          # directory where module files are stored
 module_inppath : ./minps      # directory where module input files are stored
 module_sim :                  # simulation module dictionary
  name : sim                    # simulation module name
  inpname : siminp.yaml         # simulation module input file
  vars :                        # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims             # (example) simulation data storage directory in simulation module input file
 module_anlys :                # analysis module dictionary
  name : anlys                  # analysis module name
  inpname : anlysinp.yaml       #  analysis module input file
  vars :                        # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6                   # (example) optype  in analysis module input file
 module_ml :                   # ml module dictionary
  name : ml                     # ml module name directory
  inpname : mlinp.yaml          # ml module input file name
   vars :                       # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee          # (example) optimization technique in ml module input file
```

INSIDE THE module_anlys DICTIONARY:
Name of the analysis module.
<u>anlys.py should exist in $module_path</u>

How to write an analysis module is described in slides 41-43.

input.yaml

```
mpts:                      # mpts dictionary
 libname : lib.txt          # input library file
 inp_path : ./inp           # input directory
 out_path : ./out           # output directory
 niter : 100                # number of iterations for which main of MPTS is executed
 sleeptime : 60             # sleeptime between MPTS iterations.
 initfile : initlib.txt     # library file for initial batch
 nextsize : 20              # next best candidate batch size
 mliter : 5                # number of framework iterations.
 module_path : ./mods       # directory where module files
 module_inppath : ./minps   # directory where module input files  directory
 module_sim :               # simulation module dictionary
  name : sim                 # simulation module name
  inpname : siminp.yaml      # simulation module input file
  vars :                     # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims          # (example) simulation data storage directory in simulation module input file
 module_anlys :             # analysis module dictionary
  name : anlys               # analysis module name
  inpname : anlysinp.yaml    #  analysis module input file
  vars :                     # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6                # (example) optype  in analysis module input file
 module_ml :                # ml module dictionary
  name : ml                  # ml module name directory
  inpname : mlinp.yaml       # ml module input file name
   vars :                    # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee       # (example) optimization technique in ml module input file
```

INSIDE THE module_anlys DICTIONARY:
Input file to the analysis module.
anlysinp.yaml should exist in $module_inppath

This contains the information about variables required to analyze the data from simulations to obtain y_i. E.g., op type (structure factor, greatest cluster).

How to write an analysis input module file is described in slide 40.

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

INSIDE THE module_anlys DICTIONARY:

An optional dictionary of name vars.

The purpose of this dictionary is to specify additional inputs to the analysis module.

Example : optype : Estimate q6 based order parameter.

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

Python dictionary of name module_ml is initialized *(DO NOT CHANGE)*

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

INSIDE THE module_ml DICTIONARY:
Name of the machine learning module.
ml.py should exist in $module_path

How to write an ml module is described in slides 45-47.

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

INSIDE THE module_ml DICTIONARY:
Input file to the ml module.
<u>mlinp.yaml should exist in $module_inppath</u>

This contains the input to machine learning approach that is applied to propose the next best batch.

How to write a ml module input file is described in slides 44.

input.yaml

```
mpts:                     # mpts dictionary
 libname : lib.txt         # input library file
 inp_path : ./inp          # input directory
 out_path : ./out          # output directory
 niter : 100               # number of iterations for which main of MPTS is executed
 sleeptime : 60            # sleeptime between MPTS iterations.
 initfile : initlib.txt    # library file for initial batch
 nextsize : 20             # next best candidate batch size
 mliter : 5               # number of framework iterations.
 module_path : ./mods      # directory where module files are stored
 module_inppath : ./minps  # directory where module input files  directory
 module_sim :              # simulation module dictionary
  name : sim                # simulation module name
  inpname : siminp.yaml     # simulation module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in simulation module input file
   sim_path : ./sims         # (example) simulation data storage directory in simulation module input file
 module_anlys :            # analysis module dictionary
  name : anlys              # analysis module name
  inpname : anlysinp.yaml   #  analysis module input file
  vars :                    # OPTIONAL dictionary to specify the arguments in analysis module input file
   optype : q6               # (example) optype  in analysis module input file
 module_ml :               # ml module dictionary
  name : ml                 # ml module name directory
  inpname : mlinp.yaml      # ml module input file name
   vars :                   # OPTIONAL dictionary to specify the arguments in ml module input file
    opt_technique : bee      # (example) optimization technique in ml module input file
```

INSIDE THE module_ml DICTIONARY:

An optional dictionary of name vars.

The purpose of this dictionary is to specify additional inputs to the ml module.

Example : opt_technique : Bayesian optimization technique

Input Library File:
How to write

FILE DETAILS

- Input library file contains the input data and output data for every candidate in the design library.
- It is of text file format that uses single space to separate values.
- The first line of the file contains the variable names.
- Following lines contain the variable values, where each line represents a data of a single candidate.

REQUIREMENTS

- Variable “cand_name” must be specified. It is candidate name. This variable is a unique name given to every candidate and is used for not repeating the simulations for a given candidate.
- Variable “Output” must be specified. The value of that variable for every candidate should be set to NA. An error will be thrown if the value other than NA found for any candidate.
- Variables that begin with string “desc_” are the descriptors. A vector of descriptors are used as an input to the surrogate machine learning model. An error is thrown if there is no item found with “desc_”.
- Variable “id” must be specified. It is an integer id of a candidate which is essentially the index of candidate in the input library. The index value starts from 0.

Input Library File:
How to write

GOOD EXAMPLE

```
cand_name desc_eps desc_sigma Output id
eps0.4sig1.0 0.4 1.0 NA 0
eps0.5sig1.1 0.5 1.1 NA 1
eps0.6sig1.2 0.6 1.2 NA 2
eps0.7sig1.3 0.7 1.3 NA 3
eps0.8sig1.4 0.8 1.4 NA 4
eps0.9sig1.5 0.9 1.5 NA 5
eps1.0sig1.6 1.0 1.6 NA 6
eps1.1sig1.7 1.1 1.7 NA 7
```

- cand_name is specified
- Atleast two variables contain string starting with desc_
- All output values are NA
- id is specified with correct index.

BAD EXAMPLE

```
cand_name desc_eps desc_sigma Output id
eps0.4sig1.0 0.4 1.0 NA 2
eps0.5sig1.1 0.5 1.1 NA 1
eps0.6sig1.2 0.6 1.2 NA 2
eps0.7sig1.3 0.7 1.3 NA 5
eps0.8sig1.4 0.8 1.4 NA 4
eps0.9sig1.5 0.9 1.5 NA 5
eps1.0sig1.6 1.0 1.6 NA 3
eps1.1sig1.7 1.1 1.7 NA 7
```

- Incorrect ids
- Id should simply be, 0,1,2,…M-1, where M is the number of candidates in library

BAD EXAMPLE

```
cand_name Output id
eps0.4sig1.0 NA 0
eps0.5sig1.1 NA 1
eps0.6sig1.2 NA 2
eps0.7sig1.3 NA 3
eps0.8sig1.4 NA 4
eps0.9sig1.5 NA 5
eps1.0sig1.6 NA 6
eps1.1sig1.7 NA 7
```

- No variables starting with desc_
- No descriptors in the library

BAD EXAMPLE

```
cand_name desc_eps desc_sigma Output id
eps0.4sig1.0 0.4 1.0 NA 0
eps0.5sig1.1 0.5 1.1 NA 1
eps0.6sig1.2 0.6 1.2 2.0 2
eps0.7sig1.3 0.7 1.3 NA 3
eps0.8sig1.4 0.8 1.4 NA 4
eps0.9sig1.5 0.9 1.5 4.0 5
eps1.0sig1.6 1.0 1.6 NA 6
eps1.1sig1.7 1.1 1.7 NA 7
```

- All Output values should be NA

Initial batch Library File: How to write

FILE DETAILS

- Initial batch library file contains the input data and output data for initial batch of candidates in the design library.
- It is of the same format and has the same requirements as input library file (see Slide 32-33).
- Only difference is in under the variable name id. In this file, id is the integer id of a candidate in the input library.

EXAMPLE,

Consider my input library is lib.txt (shown on the right) and I want to use 2 and 4th candidate in my initial batch.

Input library : lib.txt

```
cand_name desc_eps desc_sigma Output id
eps0.4sig1.0 0.4 1.0 NA 0
eps0.5sig1.1 0.5 1.1 NA 1
eps0.6sig1.2 0.6 1.2 NA 2
eps0.7sig1.3 0.7 1.3 NA 3
eps0.8sig1.4 0.8 1.4 NA 4
eps0.9sig1.5 0.9 1.5 NA 5
eps1.0sig1.6 1.0 1.6 NA 6
eps1.1sig1.7 1.1 1.7 NA 7
```

My initial batch library (initlib.txt) should look like the one on the right.

Input library : initlib.txt

```
cand_name desc_eps desc_sigma Output id
eps0.6sig1.2 0.6 1.2 NA 2
eps0.8sig1.4 0.8 1.4 NA 4
```

Note that the ids here are not 0,1 but the ids of input library.

Output of MPTS

- Output of MPTS is located in directory named $out_path. (see Slide 12).
- It contains two types of files for every framework iteration named "prpsl_$*i*.txt" and "train_$*i*.txt".
- Here, *i* is iteration number of the framework.
- prpsl_$*i*.txt contains the data for proposed next best batch of candidates for an iteration, *i*.
- train_$*i*.txt contains the concatenation of data in all the proposal files upto an iteration, *i*.
- *Note that prpsl_0.txt is the user-defined initial batch file.*

EXAMPLE :

If the two ML iterations are finished successfully and simulations are in progress for the third iteration, prpsl_0.txt, train_0.txt , prpsl_1.txt, train_1.txt and prpsl_2.txt are generated in output directory.

prpsl_0.txt contains same candidates as initial batch file. This batch file contains filled Output column for every candidate as simulations are completed and analysis has been done.
train_0.txt is same as prpsl_0.txt

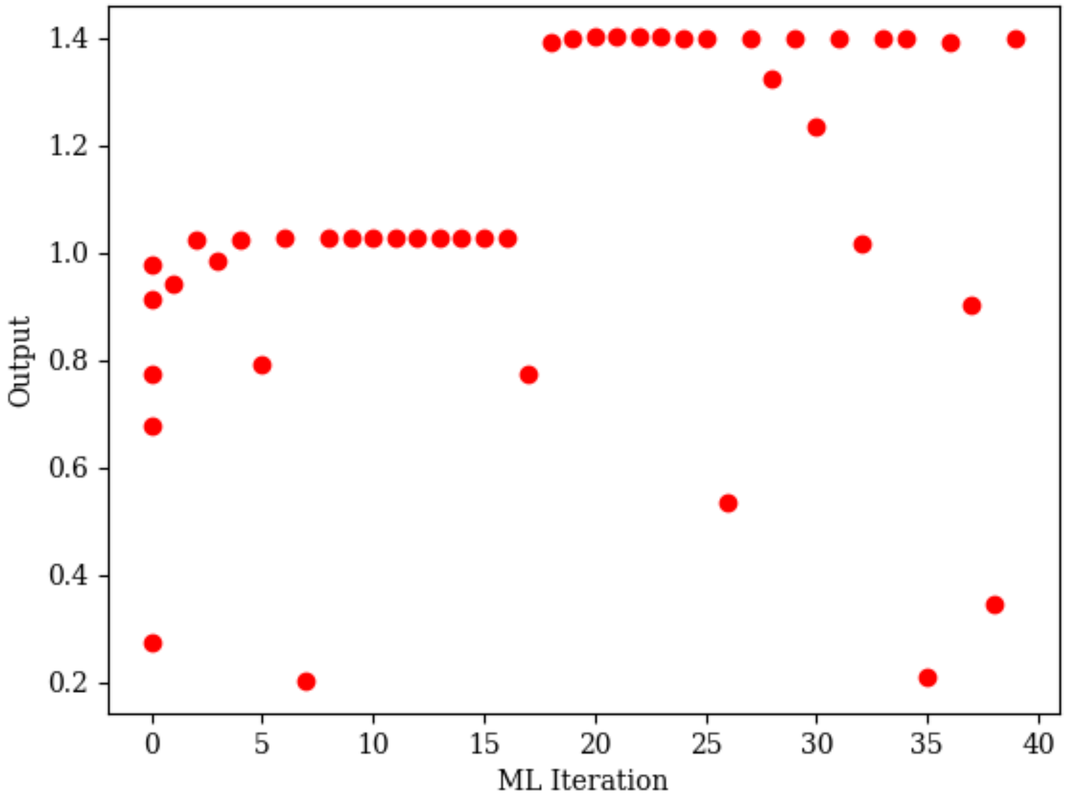
prpsl_1.txt is the batch file generated after running ML iteration 1. This batch file contains filled Output column for every candidate as simulations are completed and analysis has been done
train_1.txt is the concatenation of prpsl_0.txt and prpsl_1.txt.

prpsl_2.txt is the batch file generated after running ML iteration 2. This batch file still contain NA as Output value as simulations are in progress..

How to quickly visualize the progress of MPTS

MPTSanlys.py –y input.yaml

Following plot along with the raw data will be generated in the directory $out_path/Anlys.



Output values of all candidates in a proposed batch for a given ML iteration number.

Simulation module input file
How to write

FILE DETAILS

- A simulation module input file is an input to the simulation module.
- It is of .yaml format.
- It contain the conditions under which simulations are performed (e.g. temperature and pressure).

FILE REQUIREMENTS

- It is of .yaml format
- It must start with initializing dictionary named sim
- It must contain sim_path. This is a path where data from simulations are saved.

EXAMPLE : siminp.yaml

```
sim:
 sim_path : ./simstmp
```

- It may or may not contain additional variables depending on the requirement of simulation module.

Simulation module
How to write

- A simulation module is simply a python code that runs simulations for a given batch of candidates.

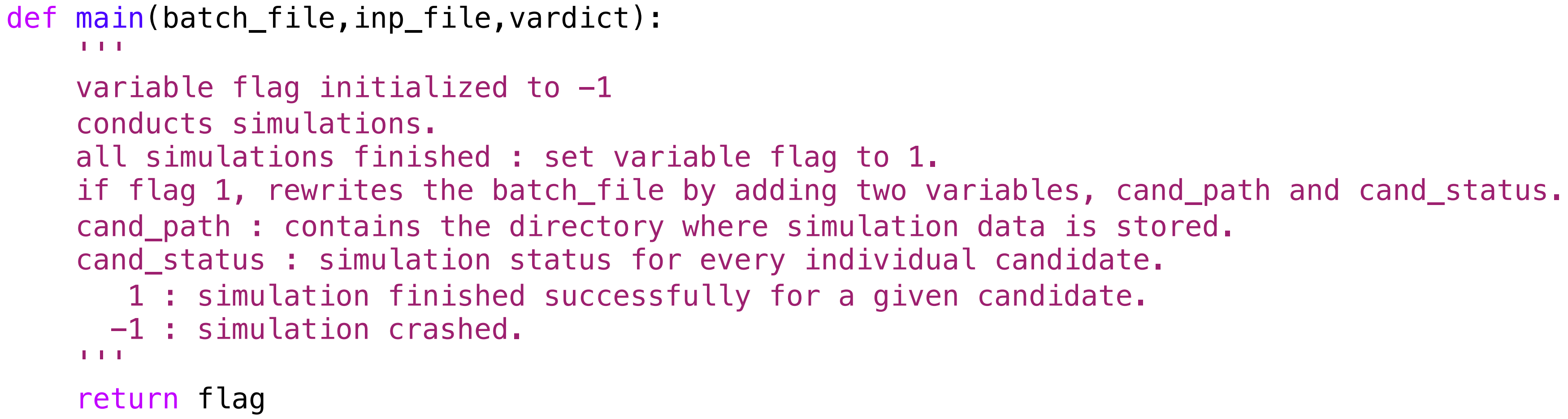
```
def main(batch_file,inp_file,vardict):
```

INPUT

- It contains a main function which has 3 input arguments.
- First input argument is a batch_file. This is a given batch library file. For the first iteration, it is the initial batch library file. Format of this file is discussed in Slides 32-34.
- Second input argument is an inp_file. This is an input file to the simulation module. It is of .yaml format. Details of this file are provided in Slide 36.
- Third input argument is a dictionary named vars. This is an additional input dictionary as specified in input.yaml (see Slide 23).

Simulation module
How to write

- A simulation module is simply a python code that runs simulations for a given batch of candidates.



```
def main(batch_file,inp_file,vardict):
    '''
    variable flag initialized to -1
    conducts simulations.
    all simulations finished : set variable flag to 1.
    if flag 1, rewrites the batch_file by adding two variables, cand_path and cand_status.
    cand_path : contains the directory where simulation data is stored.
    cand_status : simulation status for every individual candidate.
       1 : simulation finished successfully for a given candidate.
      -1 : simulation crashed.
    '''
    return flag
```

EXECUTION AND OUTPUT

- It reads batch_file, inp_file.
- It creates a unique directory named $sim_path/$cand_name for every candidate.
- It goes inside the directory and runs simulations for every candidate ONLY ONCE (NO RESUBMITTING OF JOBS OF SAME CANDIDATE).
- Any data that will be generated by simulations will be stored inside this directory.
- MPTS calls the main of simulation module periodically and checks whether simulations are finished.
- Once simulations are finished for the entire batch, it returns the flag variable of value 1.
- If simulations are still underway, a flag of -1 is returned.
- If flag 1, it rewrites the batch file by adding two variables. (details mentioned above)

Simulation module
Example

INPUT

batch_file

```
cand_name desc_eps desc_sigma Output id
eps0.6sig1.2 0.6 1.2 NA 2
eps0.8sig1.4 0.8 1.4 NA 4
```

inp_file

```
sim:
 sim_path : ./simstmp
```

vardict

```
{'sim_path': './sims'}
```

OUTPUT WHEN SIMULATIONS ARE NOT FINISHED.

No rewriting of batch_file. Returns flag value as -1

OUTPUT WHEN SIMULATIONS ARE FINISHED. Flag as 1.

batch_file is rewritten.

```
cand_name desc_eps desc_sigma Output id cand_status cand_path
eps0.6sig1.2 0.6 1.2 NA 2 −1 ./sims/eps0.6sig1.2
eps0.8sig1.4 0.8 1.4 NA 4  1 ./sims/eps0.8sig1.4
```

Cand status of -1 means simulation is crashed.
Cand status of 1 means simulation ran successfully.

IMPORTANT NOTE : cand_path here is ./sims/.. . It is not ./simstmp.
It is because vardict and sim both have sim_path as variables. In that case variable values in vardict are used.

Analysis module input file
How to write

FILE DETAILS

- An analysis module input file is an input to the analysis module.
- It is of .yaml format.

FILE REQUIREMENTS

- It is of .yaml format
- It must start with initializing dictionary named anlys.

EXAMPLE : anlysinp.yaml

```
anlys: {}
```

Empty anlys dictionary

- It may or may not contain any variables depending on the requirement of analysis module.

Analysis module
How to write

- An analysis module is simply a python code that analyzes the data obtained from running simulations of a given batch of candidates.
- Note that this module is only run when all simulations are finished for a given batch.

```
def main(batch_file,inp_file,vardict):
```

INPUT

- It contains a main function which has 3 input arguments.
- First input argument is a batch_file. This is a batch library file written by simulation module which contains the variables cand_path and cand_status. (check slides 35-38).
- Second input argument is an inp_file. This is an input file to the analysis module. It is of .yaml format. Details of this file are provided in Slide 39.
- Third input argument is a dictionary named vars. This is an additional input dictionary as specified in input.yaml (see Slide 27).

Analysis module
How to write

- A simulation module is simply a python code that runs simulations for a given batch of candidates.

```
def main(batch_file,inp_file,vardict):
    '''
    analyzes simulations.
    rewrites the batch_file. Fills the Output column for every candidate.
    If for any candidate, cand_status is -1 (crashed simulation), string FAIL is written.
    '''
```

EXECUTION AND OUTPUT

- It reads batch_file, inp_file.
- Analyzes the data and obtain the output value for every candidate in a given batch.
- It rewrites the batch_file by filling the Output column for every candidate in a given batch.
- If for any candidate, cand status is -1 (crashed simulation), string FAIL is written.

Analysis module
Example

INPUT

batch_file

```
cand_name desc_eps desc_sigma Output id cand_status cand_path
eps0.6sig1.2 0.6 1.2 NA 2 -1 ./sims/eps0.6sig1.2
eps0.8sig1.4 0.8 1.4 NA 4  1 ./sims/eps0.8sig1.4
```

inp_file

```
anlys: {}
```

vardict

```
{}
```

OUTPUT ON ANALYZING THE SIMULATIONS

batch_file is rewritten.

```
cand_name desc_eps desc_sigma Output id cand_status cand_path
eps0.6sig1.2 0.6 1.2 FAIL 2 -1 ./sims/eps0.6sig1.2
eps0.8sig1.4 0.8 1.4 2.3 4  1 ./sims/eps0.8sig1.4
```

If cand status is -1, FAIL is written.

If cand status is 1, a numerical value obtained by analysis module is written.

ML module input file
How to write

FILE DETAILS

- A machine learning module input file is an input to the machine learning module
- It is of .yaml format.

FILE REQUIREMENTS

- It is of .yaml format
- It must start with initializing dictionary named ml.

EXAMPLE : mlinp.yaml

```
ml: {}
```

Empty ml dictionary

- It may or may not contain any variables depending on the requirement of ml module.

ML module
How to write

- A simulation module is simply a python code that runs simulations for a given batch of candidates.

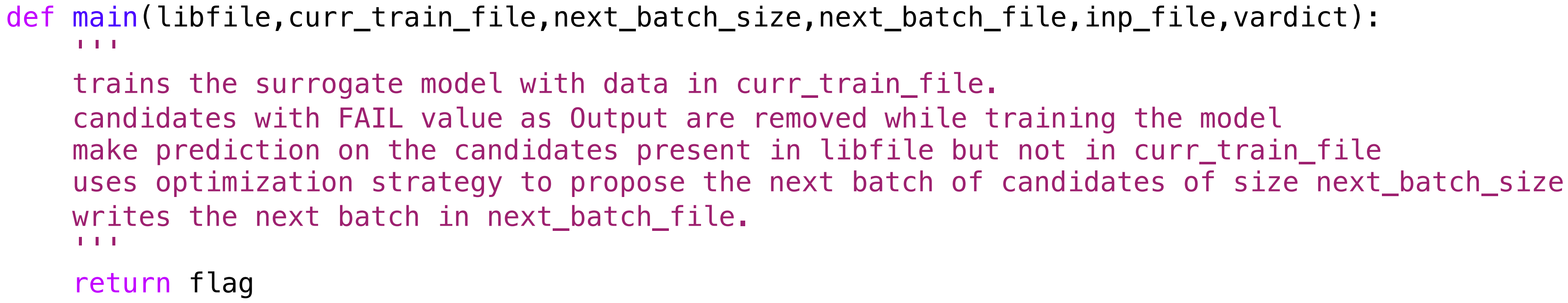
```
def main(libfile,curr_train_file,next_batch_size,next_batch_file,inp_file,vardict):
```

INPUT

- It contains a main function which has 5 input arguments.
- First input argument is libfile. This is the input library file containing data for every candidate in the library (see Slides 32-33). Note that this library contain NA values as Output.
- Second input argument is a curr_train_file. This is a train file containing the data gathered from all previous iterations for which Output variable is estimated.
- Third input argument is a next_batch_file. This is a batch file which will contain the data for the next best batch.
- Third input argument is a next_batch_size. This is a size of next batch.
- Fourth input argument is an inp_file. This is an input file to the ML module. It is of .yaml format. Details of this file are provided in Slide 39.
- Fifth input argument is a dictionary named vars. This is an additional input dictionary as specified in input.yaml (see Slide 31).

ML module
How to write

- A simulation module is simply a python code that runs simulations for a given batch of candidates.



```python
def main(libfile,curr_train_file,next_batch_size,next_batch_file,inp_file,vardict):
    '''
    trains the surrogate model with data in curr_train_file.
    candidates with FAIL value as Output are removed while training the model
    make prediction on the candidates present in libfile but not in curr_train_file
    uses optimization strategy to propose the next batch of candidates of size next_batch_size
    writes the next batch in next_batch_file.
    '''
    return flag
```

EXECUTION AND OUTPUT

- It reads libfile, curr_train_file and inp_file.
- It trains the surrogate model with data in curr_train_file.
- Candidates with FAIL value as Output are removed while training the model.
- It then makes prediction on the candidates present in libfile but not in curr_train_file.
- It uses optimization strategy to propose the next batch of candidates of size next_batch_size.
- It writes the next batch in next_batch_file.

ML module
Example

INPUT

libfile

```
cand_name desc_eps desc_sigma Output id
eps0.4sig1.0 0.4 1.0 NA 0
eps0.5sig1.1 0.5 1.1 NA 1
eps0.6sig1.2 0.6 1.2 NA 2
eps0.7sig1.3 0.7 1.3 NA 3
eps0.8sig1.4 0.8 1.4 NA 4
eps0.9sig1.5 0.9 1.5 NA 5
eps1.0sig1.6 1.0 1.6 NA 6
eps1.1sig1.7 1.1 1.7 NA 7
```

curr_train_file

```
cand_name desc_eps desc_sigma Output id cand_status cand_path
eps0.6sig1.2 0.6 1.2 FAIL 2 -1 ./sims/eps0.6sig1.2
eps0.8sig1.4 0.8 1.4 2.3 4  1 ./sims/eps0.8sig1.4
eps1.0sig1.6 1.0 1.6 1.1 6 1 ./sims/eps1.0sig1.6
eps1.1sig1.7 1.1 1.7 3.7 7 1 ./sims/eps1.1sig1.7
```

next_batch_size : 2

inp_file

```
ml: {}
```

vardict

```
{}
```

OUTPUT

Writing next_batch_file

```
cand_name desc_eps desc_sigma Output id
eps0.4sig1.0 0.4 1.0 NA 0
eps0.5sig1.1 0.5 1.1 NA 1
```

2 candidates are proposed using ML module.

While training a surrogate model, candidate with output FAIL is removed from training.