MetaMethyLib Usage Manual

Disclaimer: MetaMethyLib is currently available as an alpha release (work in progress). The current version is still under development.

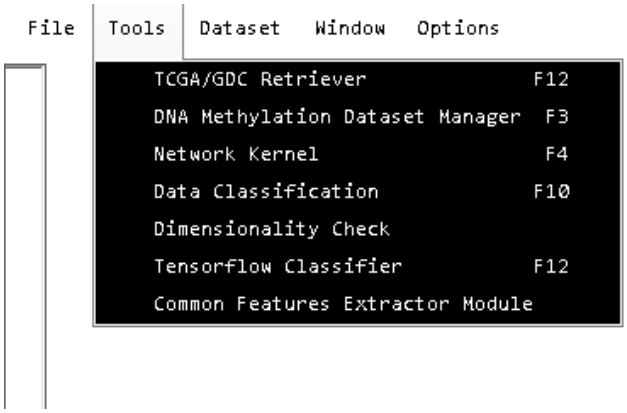
Main features:

1) Download DNA methylation data from TCGA portal.
2) Process the downloaded data to build the dataset.
3) Perform the metaheuristic feature selection of DNA methylation data (1st stage)
4) Perform the DNN pan-cancer classification (2nd stage).

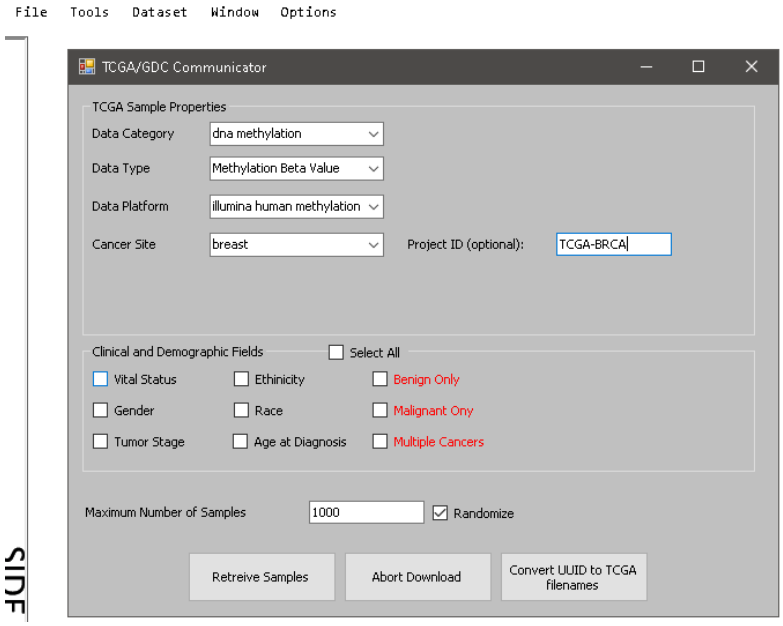
Data collection from TCGA portal

To download the DNA methylation data from the TCGA data portal, follow these steps:

a. Launch the application and press F12 or navigate to **Tools->"TCGA/GDC Retriever"**



b. Input the required information and then click on **"Retrieve Samples"** button.



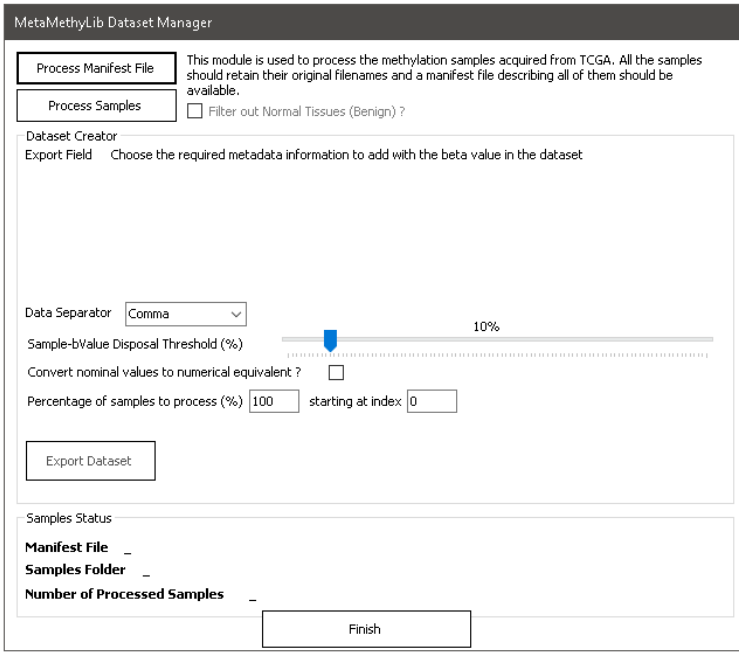
Select a folder where the DNA methylation samples, along with the .cvs manifest file, will be saved.

Notes:

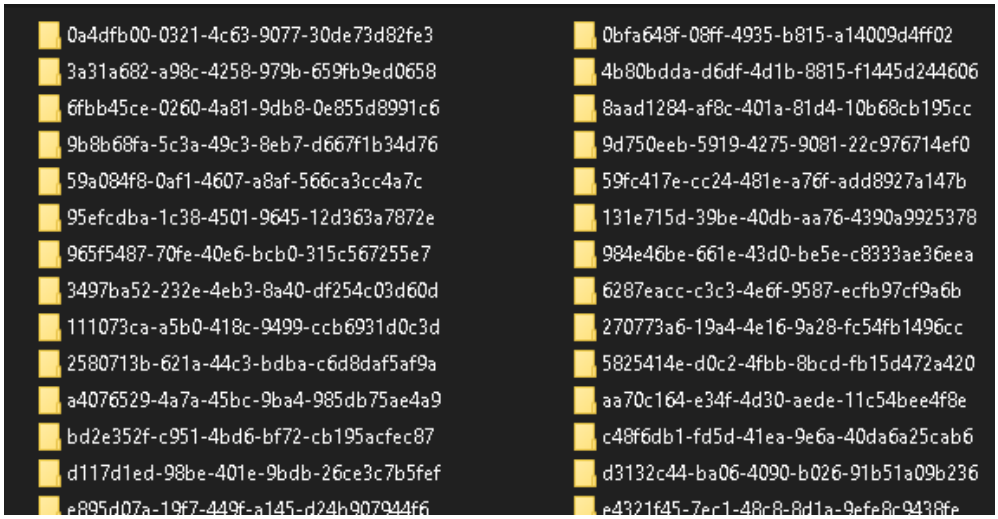
1) The optional project ID is to specify the exact TCGA project name if the current cancer site contains more than one project. You can select optional clinical field (e.g. vital status, gender, race, etc.) to be downloaded if available on the TCGA portal.
2) You can select the "Benign Only" option to download benign samples only, or "Malignant Only" to download malignant samples only. By default, the application will download random mixed samples (if available).
3) The "Convert UUID to TCGA filenames" is currently only required if the downloaded samples are based on the "Illumina 450k methylation platform". You can use this to convert the filenames from UUID to TCGA ID, so that you can use the downloaded samples with the application.

Building the data set

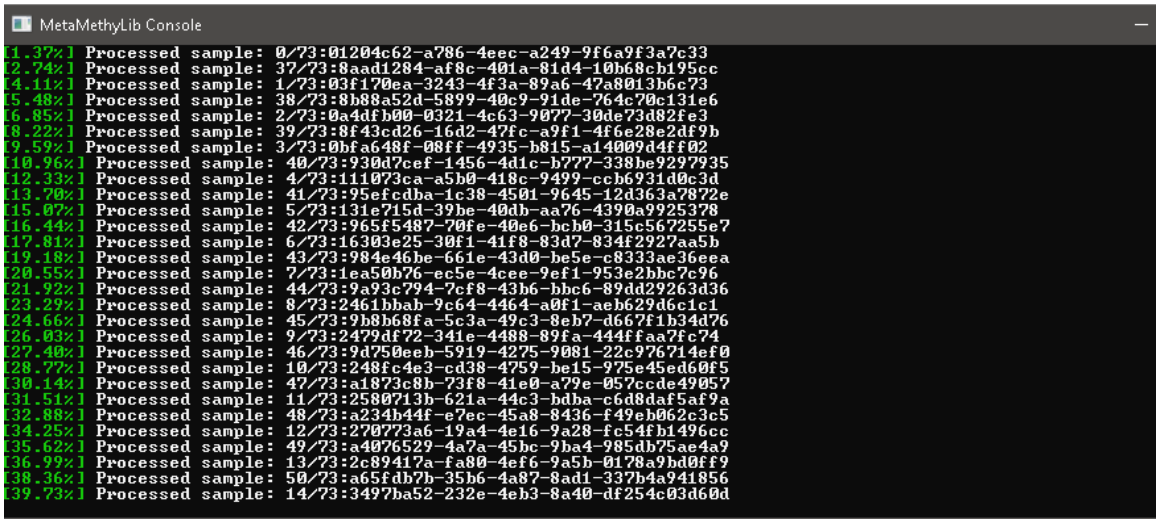
To build the unified data set that will be used in the 1st stage, press F3 or navigate to **Tools->"DNA Methylation dataset manger"**. Then follow these steps:



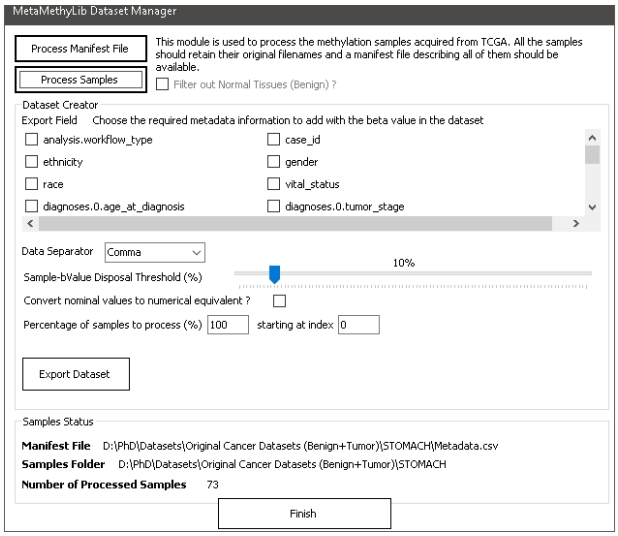
a. Click on the **"Process Manifest File"** button and select the created CSV manifest file. Make sure you untick the **"Filter out Normal tissues"** checkbox if you need to process all entries (malignant + benign).
b. Click on the **"Process Samples"** button and select the folder where the downloaded samples are located (illustrated in the figure below).



c. The application will start processing the data



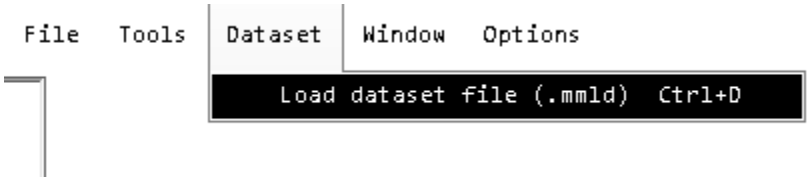
d. After the data is successfully processed, the list of available clinical information will be displayed (if available). Select any clinical information that you want to include in the data set file



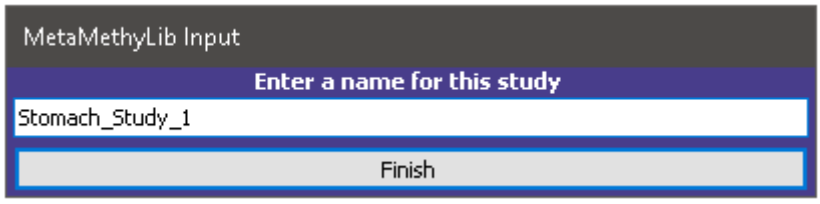
e. Select the cut-off threshold of missing beta values from the original TCGA DNA methylation samples. All the samples that contain missing beta values higher than the specified threshold will not be included in the final data set. You can also specify the percentage of samples that you need to include in the final data set (e.g. 70%, 30%).
f. The application will create the data set file (.mmld) that will be used later, in the metaheuristic feature selection stage.

Implementing the 1st Stage (Metaheuristic Feature Selection)

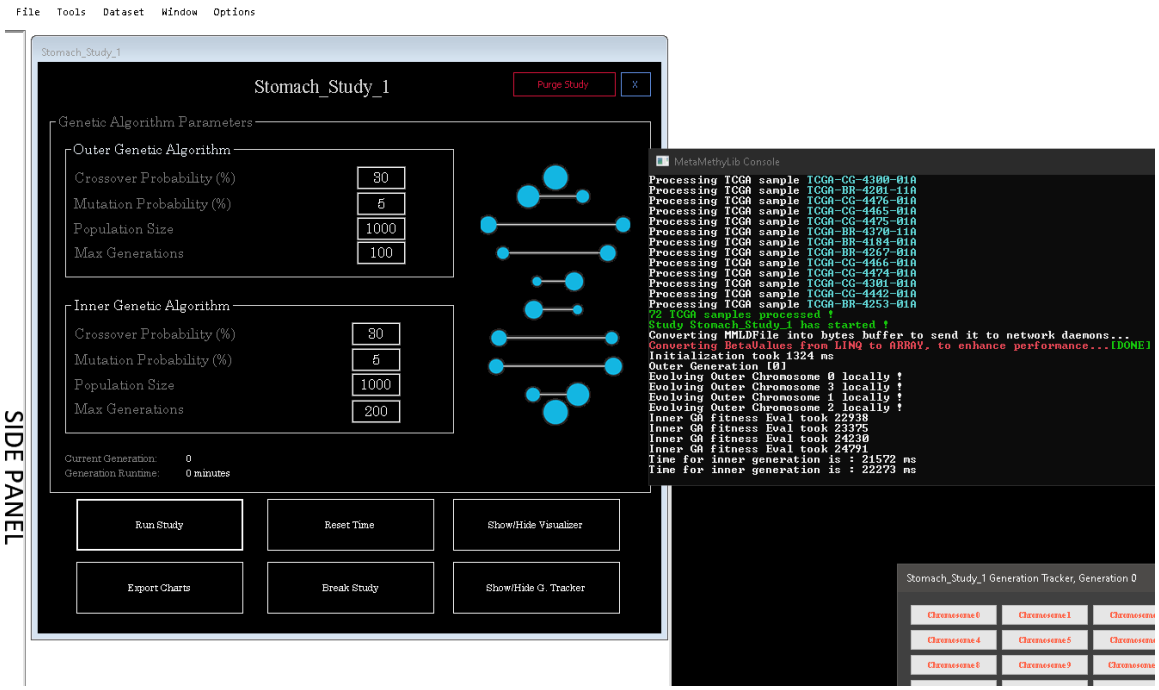
a. First, you need to load the generated data set file (.mmld). You can press "Ctrl+D", or navigate to **Dataset->"Load dataset file (.mmld)"**



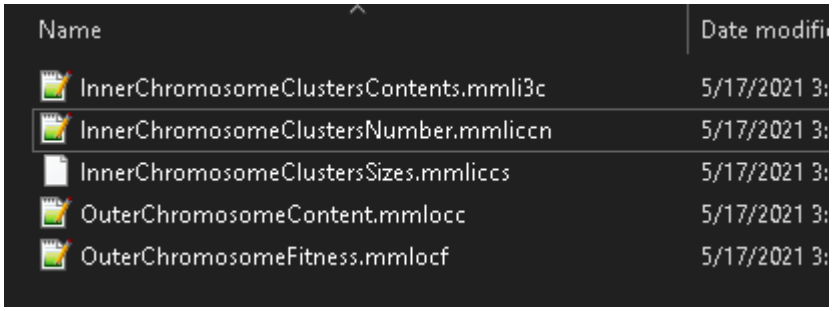
b. Then, you need to start a new study. To do so, press F5 or navigate to **File->New->Study.** Enter the desired study name and click Finish.



c. Enter the data for the nested genetic algorithms(s) and click on the **"Run Study"** button. (note: the animation is just to give confidence that the convergence is not locked in a non-responsive state).



d. The first stage will start its convergence process as specified. It is important to note that you should “save” the study first before running it, to make sure that the auto-save feature will work. This feature exports the current progress to the workspace file after each generation. This file can be loaded anytime, and the application can resume the convergence process from where it stopped. In order to save the workspace (including all the active study sheets), navigate to **File->”Save workspace”** or process “Ctrl+S”.
e. The following files will be created and updated throughout the convergence process.



Notes: by default, the application only uses local convergence on the host PC. If you need to use the network convergence feature. You need to launch the application in daemon mode and preferably, pass the required data set. To launch the application on network daemon mode**,** you need to run it from a command prompt (CMD) or by creating a shortcut and adding the following argument

```
-ParallelDaemon
```

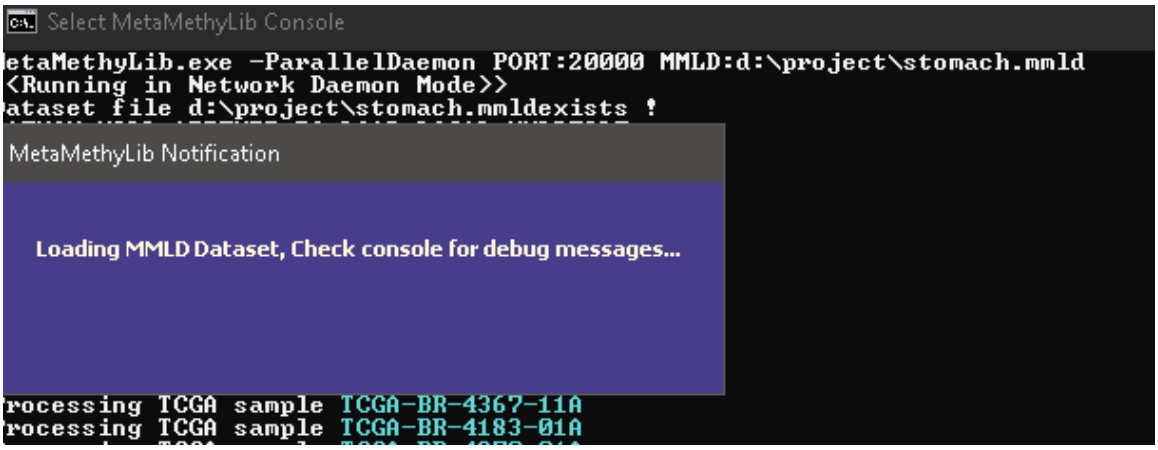
The network daemon will start listening on the default IPv4 address of the local host. You can specify the listening port and the data set file by adding the following arguments, respectively:

```
PORT:xxxx MMLD:yyyy
```

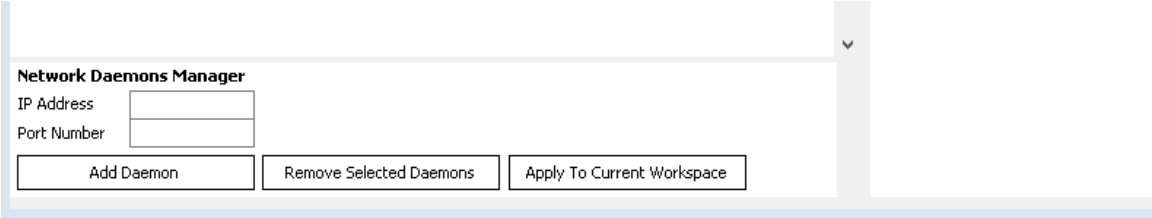
Where ‘xxxx’ is the port number and ‘yyyy’ is the full path of the .mmld data set file.

For example, to run the application as network daemon on port 20000 and data set file located in “d:\project\stomach.mmld”, you should use the following command:

```
MetaMethyLib.exe -ParallelDaemon PORT:20000 MMLD:d:\project\stomach.mmld
```



To connect the host application to the running daemon(s), navigate to **Tools->"Network Kernel"**, or press F4.



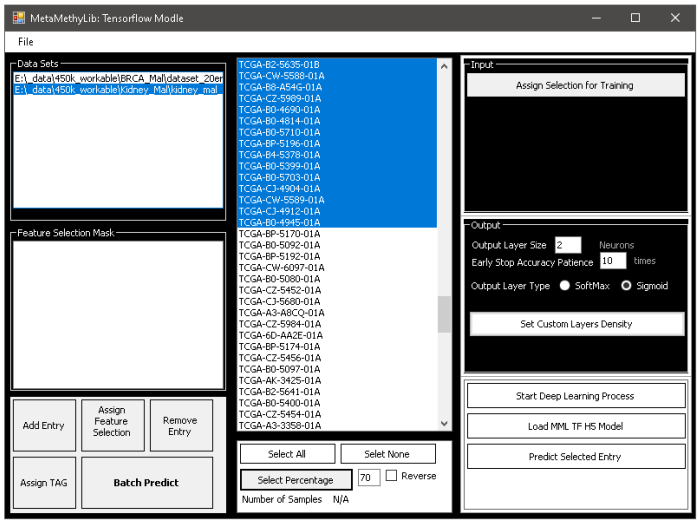
Write the IP address and port number of the daemon(s) and click the **"Add Daemon"** button and then click on the **"Apply to Current Workspace"** button.

Implementing the 2nd Stage (Classification)

For this stage, you need to make sure that all required pre-requisites are installed:

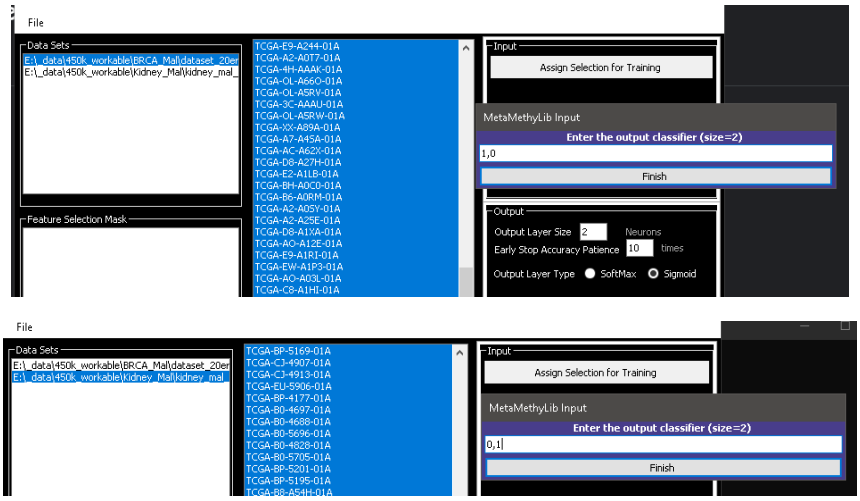
1. **Microsoft Visual C++ 2015-2019 Redistributable x86-x64** (download from https://docs.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170)
2. **Python 3.8** (download from https://www.python.org/ftp/python/3.8.0/python-3.8.0-amd64.exe)
3. **Numpy** for python (pip) (install by running `'pip install numpy'` from any command prompt)
4. **Tensorflow** for python (pip) (install by running `'pip install tensorflow` from any command prompt)
5. **Nvidia CUDA 11.2 x64** (download from https://developer.nvidia.com/cuda-11.2.0-download-archive)
6. **cuDNN 11.2 x64** (download from https://developer.nvidia.com/compute/machine-learning/cudnn/secure/8.1.0.77/11.2_20210127/cudnn-11.2-windows-x64-v8.1.0.77.zip)

To start the classification process, navigate to **Tools->"Tensorflow Classificer"** or press F12.



a. First, add the required data set for each cancer type. You can do so by clicking the **"Add entry"** button. The data set(s) are the same ones generated during the 1st stage (.mmld file).

b. Manually select the samples that you need to include in the training of the DNN, or alternatively, enter the required percentage (e.g. 70%, 30%, 20% etc.) in the percentage textbox and then click the **"Select Percentage"** button.
c. If you need to assign different names to the training labels (instead of using the filename for each added cancer type), you can click the **"Assign TAG"** button for each selected entry.
d. At this point, you can assign the generated features from the 1st stage to the entries by clicking on the **"Assign Feature Selection"** button. The extracted features from the 1st stage are found in the ".mmlocc" file. (To extract the common features from all the studies, you can use the common features extractor module by navigating to **Tools->"Common Features Extractor Module"**).
e. Now, to define the output layer's classes, select each entry, and click the **"Assign Selection for training"** button. Enter the One hot encoded data, separated by commas. You need to repeat this process for each one of the selected entries.



By default, the output layer size is estimated based on the added entries. You can explicitly change this number by modifying the value in **"Output Layer Size"** textbox. (e.g. binary classification).

f. You can set an early termination threshold to stop the model from over fitting the data. You can also specify whether the output layer is based on a "sigmoid" or a "softmax" activation function.
g. Finally, click on the **"Start Deep Learning Process"** button to start training the model using Keras Tensorflow.
h. After the model has been successfully trained, the application will prompt you to save it on your local storage device. You can cancel this step if you do not want to save the trained model.
i. To predict the result of any sample, first click on the entry in the "data sets" itembox and then select the desired sample by clicking on it. Afterwards, click on the **"Predict Selected Entry"** button.
j. For convenience, you can also batch predict all the samples in the selected entry, by first selecting the desired entry, followed by clicking on the **"Batch Predict"** button. In this

case, after the batch prediction is complete, you will be prompted to save the results on your local storage device.