

Equilibrium Molecule Classifier Instructions

This user manual is designed for a user to implement a machine learning model to predict the equilibrium state of a given organic molecule. This user manual is divided into different sections based on the utilities provided in this software program: to predict molecular equilibrium using the pre-designed model in this program; or to develop and test one's own model in predicting molecular equilibrium.

Table of project documents:

Legend

Client predictions Optional use Developer use Project reports

Document Name	Purpose
Client Instructions	This PDF with instructions for using the project.
xyz_model.py	A Python program to load the model and run predictions.
xyz_model.keras	The trained and saved deep learning model. This needs to be kept in the same folder as the xyz_model.py file.
negative.csv and positive.csv	Example input csv files for running through xyz_model.py program.
comma_to_space.py	Script to convert csv files to txt files for observation in a molecule viewer.
xyz_model_analysis.ipynb	A Jupyter Notebook for detailed model performance analysis.
README.md	A Markdown file for future development teams to learn the project.
xyz_training_model.ipynb	A Jupyter Notebook for model training and evaluation.
qm9_create_tfdataset.ipynb	A Jupyter Notebook to pre-process data before training the model.
log_extractor.pl	Script to extract coordinates from all log files in a directory.
energy_calculations.py	Script to calculate the HOMO-LUMO energy gap.
displace_coordinates.py	Script to generate nonequilibrium molecules.
log_script_single.pl	The script provided to us by Amir Karton to extract coordinates from QM9 log files.
qm9_tensorflow_datasets	A folder containing pre-processed training and testing datasets for the model.
Project Presentation	The PowerPoint presentation covering this project.
Final Project Report	The final project report with useful evaluation metrics.

Installations/System Requirements

If you intend on running the Python program for predictions from your Command Line Interface, you will need Python installed (versions 3.11 onwards), then when the program starts, it will check if there are any missing dependencies and return a list to you. If your machine has `pip` installed (a package installer for Python), it will confirm if you want the program to attempt to install all missing dependencies.

If running the workflow for model development using the other scripts and/or Jupyter notebooks, please ensure that you have the following packages installed on your machine:

- [Matplotlib](#) any version.
- [Pandas](#) versions 2.2.3 onwards.
- [Pyscf](#) any version.
- [Python](#) versions 3.11 onwards.
- [Scikit-learn](#) any version.
- [Tensorflow](#) versions 2.16 onwards.

Please refer to the following links attached to these packages to access instructions on how to install them on your local machine.

Project Usage for Predictions:

Please follow these steps to use the model to predict whether a given set of atomic coordinates represents a molecule in equilibrium:

Input Data Preparation

1. Prepare a csv file containing the molecules of interest that you want to predict, following the structure below:
 - atom type (H, C, N, O, F)
 - Spatial (X, Y, Z) coordinates.

This can be completed in a text editor and saved as a csv file if needed.

```

C,2.152789,1.401974,0.597515
C,1.998043,0.223677,-0.359047
O,3.149622,-0.608604,-0.386871
C,0.782749,-0.648626,-0.041220
O,-0.396569,0.166622,-0.190805
C,-1.564498,-0.425885,0.068045
N,-1.683472,-1.634127,0.433307
C,-2.672251,0.496153,-0.122281
N,-3.596210,1.177815,-0.253258
H,3.031185,1.993769,0.327938
H,1.270360,2.046388,0.569160
H,2.277629,1.048408,1.629434
H,1.889483,0.597758,-1.383445
H,3.390302,-0.814685,0.523311
H,0.728477,-1.499685,-0.725138
H,0.822759,-1.037900,0.983223
H,-2.657830,-1.887772,0.578517

```

Alternatively, if you were looking to extract the spatial coordinates from the QM9 dataset, then run either the *log_single_script.pl* to extract the coordinates from a single file or the *log_extractor.pl* Perl script to process an entire folder of log files by executing the following command on bash:

```
perl log_extractor.pl input_directory output_directory
```

Where:

- *input_directory* is the directory pointing to the QM9 log files of the molecules of interest.
 - *output_directory* is the directory where you'd like the spatial coordinates to be saved.
2. (Optional) If you would like to displace the coordinates, you could do this manually, or there is a *displace_coordinates.py* script that was used for displacing the training and testing datasets during the model's development.
 3. (Optional) You also have the *comma_to_space.py* script as an optional utility to be run against a single csv file and produce a plain text file with the commas replaced with spaces, this will allow for quick and easy visualisation in a molecular viewer program such as this [example](#) provided to us.

Running the model

1. To use the model to make a prediction for your molecule of interest, run the *xyz_model.py* python script by executing the following command in your

terminal:

```
python xyz_model.py coordinate_file.csv
```

2. The script will first check if your machine has the required packages installed; if not, it will check if you have pip installed. If you do have pip installed, it will prompt you to confirm if it can download the required packages (This can be permanently disabled by commenting out the indicated line in the file).
 - a. If successful, you will need to re-run the script for them to take effect.
 - b. If unsuccessful or you do not have pip installed, it will output to the console the missing packages for you to manually download.
3. Next, it will prompt whether you wish to run in debug mode. Debug mode contains a more verbose output to the console (This can be permanently disabled by commenting out the indicated line in the file).
4. It will then take a moment to load the model and print to the console its classification of the molecule and its confidence as a percentage.

Note on using Turing or Bourbaki: this program has been tested and confirmed to work on several machines. The exception is currently that Turing is using Python 3.13 which doesn't support Tensorflow. However, it does run on Bourbaki, with this, since Bourbaki is using pip3 not pip you will need to manually install the missing dependencies. Given the nature of Bourbaki, it is probably best to leave the installs to the user rather than automate them. For example: `pip3 install pyscf --user`

Project Usage for Further Model Development:

Input Data Preparation

Option 1:

If you would like to use the same training/testing data that the model was initially trained against, then please skip to the '**Training the Model**' section that is on the very last page (page 8) of this manual.

The datasets required for this section is included in the project folder under `qm9_tensorflow_datasets`.

Option 2:

If you would like to train our model using your own training/testing data, the please proceed to follow the steps below:

1. prepare a folder of csv files containing the molecules of interest that you want to train and test the model with, following the structure below:
 - a. atom type (H, C, N, O, F)
 - b. Spatial (X, Y, Z) coordinates.

These can be completed in a text editor and saved as csv files if needed.

Please note: the model is currently restricted to the above atom types and a maximum of 26 atoms per molecule.

```
C,2.152789,1.401974,0.597515
C,1.998043,0.223677,-0.359047
O,3.149622,-0.608604,-0.386871
C,0.782749,-0.648626,-0.041220
O,-0.396569,0.166622,-0.190805
C,-1.564498,-0.425885,0.068045
N,-1.683472,-1.634127,0.433307
C,-2.672251,0.496153,-0.122281
N,-3.596210,1.177815,-0.253258
H,3.031185,1.993769,0.327938
H,1.270360,2.046388,0.569160
H,2.277629,1.048408,1.629434
H,1.889483,0.597758,-1.383445
H,3.390302,-0.814685,0.523311
H,0.728477,-1.499685,-0.725138
H,0.822759,-1.037900,0.983223
H,-2.657830,-1.887772,0.578517
...
```

Alternatively, if you were looking to extract the spatial coordinates from the QM9 dataset, then run the *log_extractor.pl* Perl script to process an entire folder of log files by executing the following command on bash:

```
perl log_extractor.pl input_directory output_directory
```

where:

input_directory is the directory pointing to the QM9 log files of the molecules of interest.

output_directory is the directory where you'd like the spatial coordinates to be saved.

2. Next, you will need to create a set of 'displaced' coordinates. Ensure you keep the valid and invalid coordinate files in separate folders at this point. You could do this manually, or there is a `displace_coordinates.py` script that was used for displacing the training/testing datasets during the model's initial development. Use the file by running the following command in your terminal:

```
python displace_coordinates.py input_dir output_dir number_displaced
```

Where:

- a. *input_dir* is the directory containing your original csv file of the molecules of interest.
 - b. *output_dir* is the directory of interest where you'd like the files saved.
 - c. *number_displaced* is any number between 3 and 5, which will be the number of angstroms in which the atoms in your molecule will be displaced.
3. (Optional) Visualise your molecules of interest by running the `comma_to_space.py` Python file against a folder of csv files. It will convert all the csv files to txt and replace the commas with spaces. This will allow for quick and easy visualisation in a molecular viewer program such as the [example](#) provided to us. Run it by executing the following command in your terminal:

```
python comma_to_space.py input_directory output_directory
```

4. Calculate the HOMO-LUMO energy gaps between all the molecules from these CSV files by running the `energy_calculations.py` python file by running the following command in your terminal:

```
python energy_calculations.py molecule_dir
```

Where:

- a. *molecule_directory* represents the directory pointing to where the csv file of your original molecules of interest is stored. The energy gap will be prepended to the first line of the file before the atoms.
5. Create another csv file containing the 'displaced' version of the molecules of interest by running the `displace_coordinates.py` python file by running the

following command on bash:

```
python displace_coordinates.py input_dir output_dir number_displaced
```

Where:

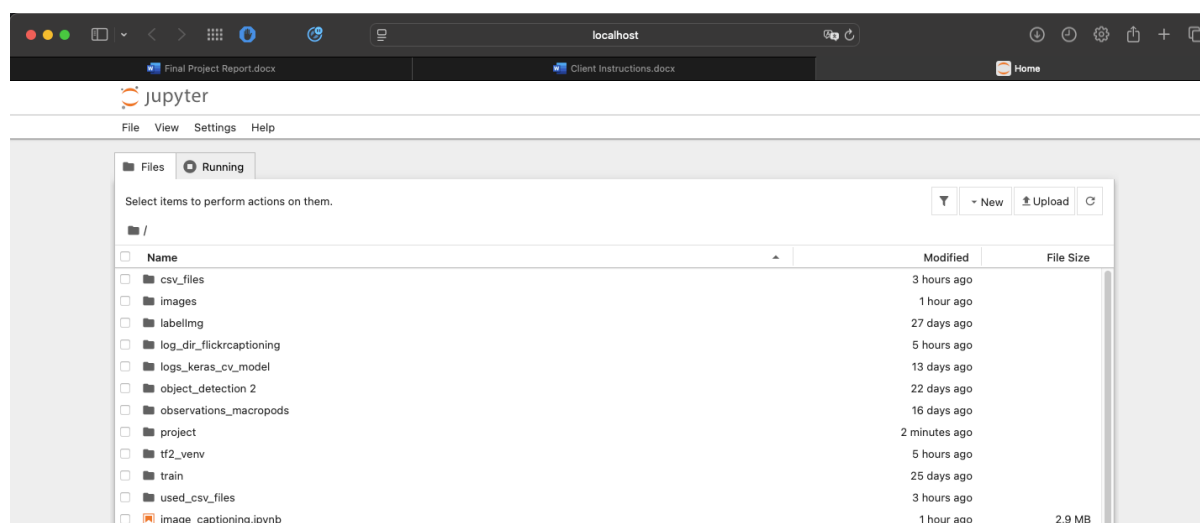
- input_dir* is the directory containing your original csv file of the molecules of interest.
- output_dir* is the directory of interest where you'd like the files saved.
- number_displaced* is any number between 3 and 5, which will be the number of angstroms in which the atoms in your molecule will be displaced.

The following sections require you to run Jupyter notebooks from your local machine. Please follow the steps below to access Jupyter from your local device:

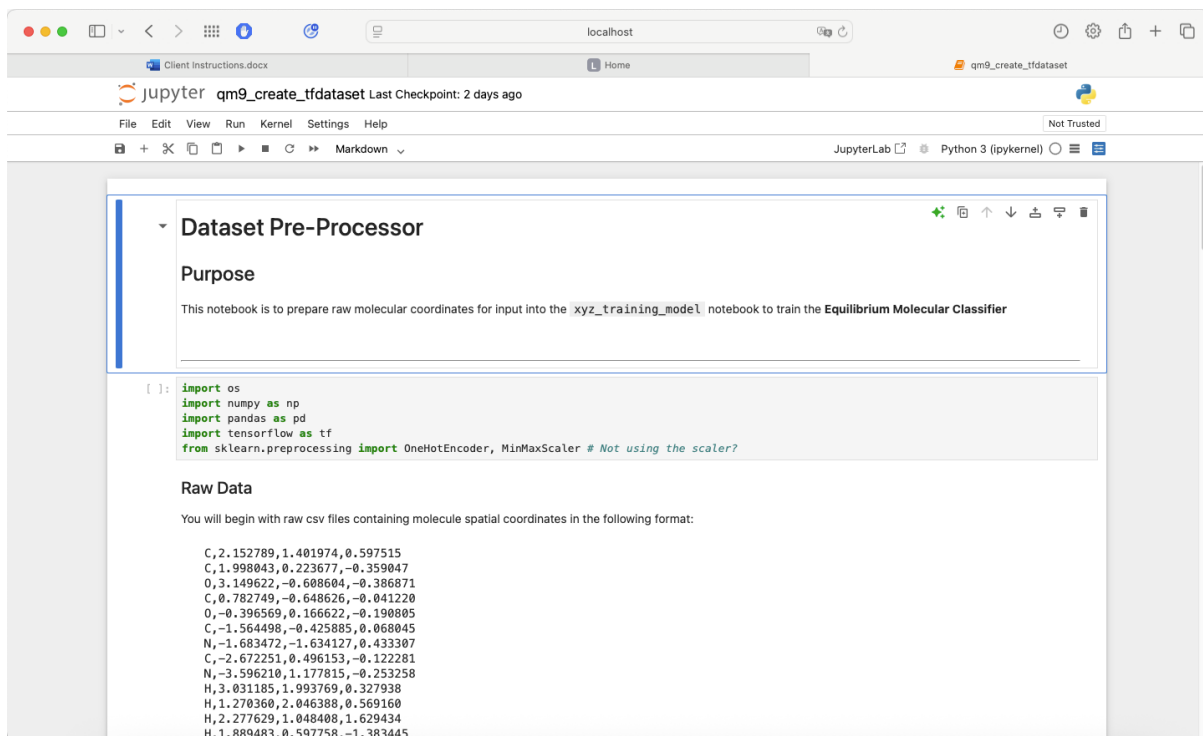
1. Run the following command in your terminal window:


```
jupyter notebook
```

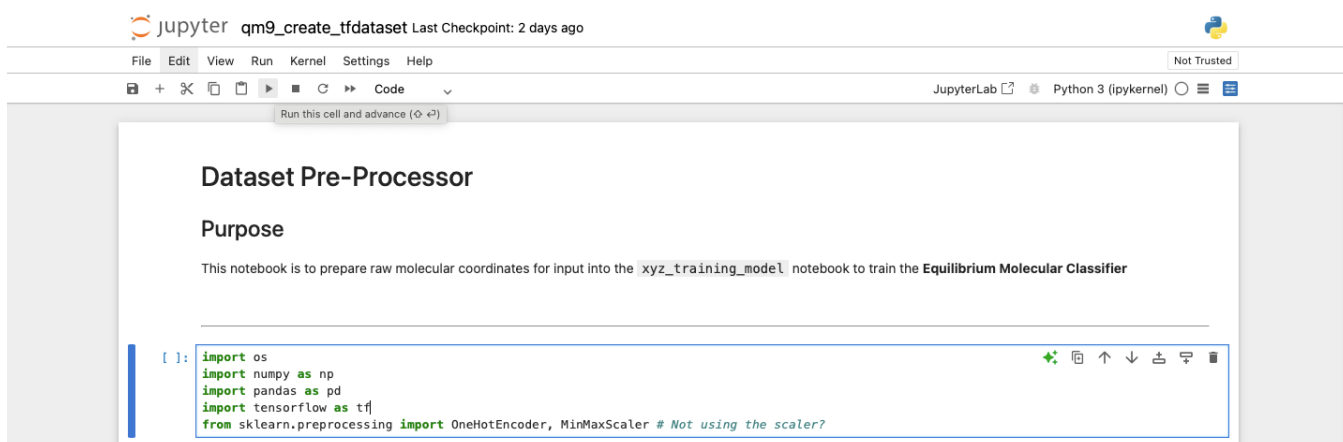
2. This command will start a local server hosted from your personal device and open Jupyter in your default browser.



The Jupyter notebook will open in a new tab - this is what it looks like when opened in your browser:



- Cells containing executable code are highlighted in grey boxes. To run the code in a given cell, click on the cell you want to execute and then click on the play  symbol at the top of the notebook to run the cell.



- Once you have finished using Jupyter, you can shut down the local server by navigating back to your terminal and pressing Ctrl+C on your keyboard. You will then receive the following message:

```
[I 2025-05-18 21:08:45.992 ServerApp] Serving notebooks from local directory: /Users/joeyhuang/Desktop/Master of Data Science/2025 Term 1/COSC551/cosc551_assignments
0 active kernels
Jupyter Server 2.15.0 is running at:
http://localhost:8890/tree?token=983ec8359ba926ddc073fc2f0a1ba034c534c50aeb2c5562
http://127.0.0.1:8890/tree?token=983ec8359ba926ddc073fc2f0a1ba034c534c50aeb2c5562
Shut down this Jupyter server (y/[n])? [I 2025-05-18 21:08:50.997 ServerApp] No answer for 5s:
```

Type 'y' in the command prompt and press enter to terminate the server.

Pre-Processing the Datasets

Open and run the *qm9_create_dataset.ipynb* notebook in the Jupyter browser, carefully following the instructions in the notebook to successfully curate the datasets above to train the neural network model.

Training the Model

Open and run the *xyz_training_model.ipynb* in the Jupyter browser, carefully following its instructions to successfully build and train the inference model you will use to make predictions on your molecules of interest.

(Optional) Analysing Model Performance

Open and run the *xyz_model_analysis.ipynb* in the Jupyter browser, carefully following its instructions regarding file paths to view an in-depth statistical analysis on our original model and test sets. This notebook can be used with other models and data but carefully read the internal documentation before doing so or basing interpretations on the contents in this case.