KURF project: Creation of polymer databases and training a Graph Neural Networks (GNN) for materials discovery.

Motivation:

This will be a tentative roadmap to achieve the main two objectives of the project. This document is going to provide a guidance in terms of the timelines and expected goals to be achieved correspondingly.

Objectives:

1. Creation of a database containing different properties computed using ab-initio methods.

2. Development of a simple GNN using the previous curated database aiming to discover new materials.

Workplan:

The project is going to be divided into two main packages. Firstly, creation and curation of a database containing 38459 polymers. The database should contain different kinds of relevant information, including but not restricted to RDKit molecular properties and ***ab-initio*** electronic properties. Main objective of this task will be portability and accessibility. This includes to create a modular object that can be employed by users wishing to construct databases from scratch employing specialised software. This will concatenate already developed software with new tools for creating databases on the fly from a series of folders.

Secondly, by employing this database, a simple GNN model will be trained using SMILES to recognise different molecules through graphs. Main objective of this part of the project is reproducibility and correct development of a working GNN through correct tokenization of the molecules. Extensive python-library research is expected for optimizing the tokenization of the molecules and the development of the GNN model. Creation of tutorials for usage of the database and training of the GNN is a main objective in this part of the project.

An ambitious goal and based on the success rate of the two previous tasks, a first attempt to train a simple Message Passing Graph Neural Network (MP-GNN) will be performed. We are interested to generate a correct workflow employing for exploiting all the available information of the created database.

Tasks per week:

1st week: Database creation

→ Testing the code ***orca_elec_prop.py*** in the small database polymer provided in Lorenz-Lab-KCL GitHub repository.

→ Developing a code to collect and store the information from the folders. Firstly single-threaded and subsequently in parallel. The parallel implementation should check the following libraries:

→ concurrent-futures
→ dask-mpi/bag
→ mpi-pool-executor

→ Providing an analytic description of the database as it is done in this repository:

https://github.com/rnepal2/Solubility-Prediction-with-Graph-Neural-Networks/blob/main/analytics.ipynb

2nd and 3rd week:

→ Supplying the raw-database to be organised using the ***orca_elec_prop.py*** and new developed code for retrieving and organising the information.

→ Function to interpret molecular information as a Graph using nodes and edges. Possibility to decorate it for further information compilation.

→ Creation of a stand-alone module in Python for subsequent integration with PySoftK. Documentation and creation of tests will be carried out.

4th , 5th , and 6th week:

→ Continuation of the previous tasks due to unexpected complications.

→ Selection of a small part of the database for training a “simple”-GNN. A benchmark concerning the size of the population vs training time.

→ Extension of the GNN to use more information for training a MP-GNN.

Resources to browse:

Introduction to GNN and its training:

https://dmol.pub/dl/introduction.html

https://www.blopig.com/blog/2022/02/how-to-turn-a-smiles-string-into-a-molecular-graph-for-pytorch-geometric/

https://uvadlc-notebooks.readthedocs.io/en/latest/tutorial_notebooks/JAX/tutorial7/GNN_overview.html

https://towardsdatascience.com/practical-graph-neural-networks-for-molecular-machine-learning-5e6dee7dc003

https://medium.com/@tejpal.abhyuday/application-of-gnn-for-calculating-the-solubility-of-molecule-graph-level-prediction-8bac5fabf600

Solubility example:
https://github.com/rnepal2/Solubility-Prediction-with-Graph-Neural-Networks

Explanation and jupyter notebook:

https://iwatobipen.wordpress.com/2019/04/05/make-graph-convolution-model-with-geometric-deep-learning-extension-library-for-pytorch-rdkit-chemoinformatics-pytorch/

https://nbviewer.org/github/iwatobipen/playground/blob/master/gcn.ipynb

HOMO-LUMO Gap example:

https://qiita.com/maskot1977/items/ede0e0c97ad0d0bce29d