RDKit to convert a dataset of molecular structures from SMILES to 3D conformations.

Name: Rohan Joydhar
Email: rohan.joydhar@accenture.com

Introduction

This report outlines the methodologies, logic, libraries, and functions utilised in the provided Jupyter Notebook to manipulate chemical data using RDKit and pandas. The code focuses on converting SMILES strings to 3D conformers and involves data manipulation and visualisation.

RDKit

RDKit is a collection of cheminformatics and machine learning tools. It allows for the manipulation of chemical structures and the performance of various computational chemistry tasks. Here are some of the key features and uses of RDKit:

- **Chemical Structure Representation:** RDKit can read and write various chemical file formats, including SMILES, SDF, Mol, and more.
- **Molecular Descriptors:** It can compute various molecular descriptors and properties, such as molecular weight, logP, and topological polar surface area.
- **Substructure Searching:** RDKit supports substructure searching and matching, allowing for the identification of specific patterns within larger molecules.
- **Molecular Fingerprints:** It can generate molecular fingerprints, which are useful for similarity searching and clustering.
- **Conformational Analysis:** RDKit can generate 3D conformations of molecules, perform conformer searches, and optimise conformers using force fields.
- **Chemical Reactions:** RDKit can represent and apply chemical reactions, enabling virtual synthesis and reaction prediction.

- **Integration with Machine Learning:** RDKit can be used in combination with machine learning libraries like TensorFlow and PyTorch to develop predictive models for various chemical properties and activities.

Libraries Used

- **pandas:** For data manipulation and analysis.

- **rdkit:** A collection of cheminformatics tools for handling chemical data.
  - Chem (from rdkit): Basic RDKit functionality for molecule handling.
  - AllChem: Additional RDKit functionality for molecular modelling and conformer generation.
  - rdDistGeom: Module for distance geometry calculations.
  - rdMolAlign: Module for molecular alignment.
  - Draw (from rdkit): Module for visualising molecules.
  - Descriptors (from rdkit): Provides molecular descriptors (not used in this code snippet).

- **IPython.display:** Allows displaying images in Jupyter notebooks or IPython environments.
- **py3Dmol:** Allows displaying images images in 3-dimensional view

Dataset

The dataset is a subset of the QM9 dataset, which consists of molecular properties for small organic molecules.It has 10k records in it. Here's a detailed description of the dataset's structure and contents.

Columns Description:

1. **mol_id**: Unique identifier for each molecule.
2. **smiles**: SMILES (Simplified Molecular Input Line Entry System) string representing the molecular structure.
3. **A, B, C**: Rotational constants (in GHz).
4. **mu**: Dipole moment (in Debye).
5. **alpha**: Isotropic polarizability (in Bohr^3).
6. **homo**: Energy of the highest occupied molecular orbital (HOMO) (in Hartree).
7. **lumo**: Energy of the lowest unoccupied molecular orbital (LUMO) (in Hartree).
8. **gap**: Gap between HOMO and LUMO (in Hartree).
9. **r2**: Electronic spatial extent (in Bohr^2).
10. **zpve**: Zero point vibrational energy (in Hartree).
11. **u0**: Internal energy at 0K (in Hartree).
12. **u298**: Internal energy at 298.15K (in Hartree).
13. **h298**: Enthalpy at 298.15K (in Hartree).

14. **g298**: Free energy at 298.15K (in Hartree).
15. **cv**: Heat capacity at constant volume (in cal/mol.K).
16. **u0_atom**: Atomization energy at 0K (in Hartree/atom).
17. **u298_atom**: Atomization energy at 298.15K (in Hartree/atom).
18. **h298_atom**: Atomization enthalpy at 298.15K (in Hartree/atom).
19. **g298_atom**: Atomization free energy at 298.15K (in Hartree/atom).

Summary:

- **A, B, C**: These rotational constants have very high variances indicating a wide range of molecular sizes and shapes.
    - **Mean values**: A ≈ 67.52, B ≈ 1.95, C ≈ 1.55
    - **Standard deviations**: A ≈ 6198.63, B ≈ 5.60, C ≈ 3.86
- **mu**: The dipole moment varies with a mean of 2.61 Debye and a standard deviation of 1.54.
- **alpha**: Isotropic polarizability has a mean of 64.34 Bohr^3 and a standard deviation of 9.40.
- **homo and lumo**: Energies of the highest and lowest occupied molecular orbitals.
    - **homo**: Mean ≈ -0.24, Std ≈ 0.03
    - **lumo**: Mean ≈ 0.01, Std ≈ 0.05
- **gap**: Energy gap with a mean of 0.26 and a standard deviation of 0.05.
- **r2**: Electronic spatial extent varies widely with a mean of 943.37 Bohr^2.
- **zpve**: Zero point vibrational energy with a mean of 0.13 Hartree.
- **u0, u298, h298, g298**: Energies and enthalpies at 0K and 298.15K.
    - **Mean values**: u0 ≈ -355.97, u298 ≈ -355.96, h298 ≈ -355.96, g298 ≈ -356.00 (all in Hartree).
- **cv**: Heat capacity at constant volume with a mean of 28.01 cal/mol.K.
- **u0_atom, u298_atom, h298_atom, g298_atom**: Atomization energies at 0K and 298.15K (all in Hartree/atom).

Methodologies Used in Code

1. **Importing Necessary Libraries:**

   The notebook begins by importing necessary libraries, primarily rdkit, pandas, and IPython.display for molecular manipulation, data handling, and visualisation.

2. **Reading the Dataset:**

   The dataset containing molecular information is read into a pandas DataFrame. This data includes SMILES strings, which are a textual representation of molecular structures.

3. **Defining Conversion Functions:**

   A function smiles_to_3d is defined to convert SMILES strings to 3D molecular structures:

   - **SMILES to Molecule**: Uses Chem.MolFromSmiles to create a 2D molecule object from a SMILES string.
   - **Embedding and Optimization**: Employs AllChem.EmbedMolecule and AllChem.UFFOptimizeMolecule to generate and optimise a 3D conformation.
   - **Validation**: Checks if the conversion and optimization are successful.

4. **Converting and Displaying Molecules:**

   Iterates through the first few molecules in the dataset, converting each SMILES string to a 3D molecular structure using the smiles_to_3d function. Successfully converted molecules are visualised using Draw.MolToImage, with the following parameters:

   - **Image Size**: (300, 300) to set the image dimensions.
   - **Kekulize**: Ensures bond orders are correctly depicted.
   - **Wedge Bonds**: Represents stereochemistry in the image.
   - **Legend**: Displays the SMILES string below the molecule.

Code Execution

The code executes in two ways. User has to input the value to determine how many molecules the user wants to visualise or the user has input a smile of a molecule. If the input smile is correct then it will be visualised.

Conclusion

The code provides a clear workflow for reading a dataset of molecular structures, converting SMILES strings to 3D conformations, and visualising these molecules using RDKit. This process is crucial for preparing molecular data for further computational analysis, such as training machine learning models.

References:

- Chatgpt , BlackBox Ai
- RDkit documentation
- Kaggle Dataset Notebook: molecular_representation
- Kaggle Dataset Notebook: Quantum Machine 9 - QM9