
Quanformer Documentation

quanformer

May 08, 2020

1	Installing quanformer	3
2	quanformer API	5
3	Utility functions	7
4	Examples	9

quanformer is a Python-based pipeline for generating conformers, preparing quantum mechanical (QM) calculations, and processing QM results for a set of input molecules.

For each molecule, conformers are generated and optimized using a molecular mechanics force field. Input files for QM calculations are then prepared for geometry optimizations, single point energy (SPE) calculations, or Hessian calculations. The user specifies any QM method and basis set that is supported in the QM software package (Psi4 or Turbomole). With completed calculations, the pipeline extracts final energies and geometries and collects job-related details such as calculation time and number of optimization steps. Analysis scripts are provided for comparing different methods in the lens of conformer energies and calculation times via easily-interpreted plots. This pipeline was tested to robustly process hundreds of conformers per molecule and hundreds of molecules, though it can likely handle more.

This code is open source on [Github](#).

Installing quanformer

1.1 Anaconda

1. Install [Miniconda](#) (or Anaconda) with Python 3.6 (version 3.7 is not yet tested).
2. Create a conda environment for **quanformer**:

```
conda create --name quanformer python=3.6 matplotlib scipy
conda activate quanformer
```

3. Obtain OpenEye, Psi4, and Psi4 dependencies:

```
conda install -c openeye openeye-toolkits
conda install -c psi4 psi4 dftd3 gcp [gpu_dfcc]
```

4. Check installations.

- a. Psi4:

```
psi4 --test
```

- b. OpenEye (in interactive Python session):

```
import openeye.ochem as ochem
mol = ochem.OEMol()
```

1.2 Quanformer

Install **quanformer** from source by the following:

1. Clone the repository from [Github](#):

```
git clone https://github.com/MobleyLab/quantformer
cd quantformer
```

2. Install:

```
python setup.py install
```

or use pip for a local install (editable installation):

```
pip install -e .
```


CHAPTER 2

quanformer API

coming soon

CHAPTER 3

Utility functions

coming soon

CHAPTER 4

Examples

Example files are provided on [Github](#).

Remember to activate your conda environment containing **quanformer**:

```
conda activate quanformer
```

4.1 Basic usage

Set up structures and prepare calculations:

```
import quanformer.pipeline as qp

qp.setup_conformers('two_alkanes.smi')
qp.setup_calculations('two_alkanes-200.sdf', 'mp2', 'def2-sv(p)')
```

Your directory structure should look something like this:

```
.
├── AlkEthOH_c1008
│   ├── 1
│   │   └── input.dat
│   ├── 2
│   │   └── input.dat
│   └── 3
│       └── input.dat
├── AlkEthOH_c312
│   ├── 1
│   │   └── input.dat
│   ├── 2
│   │   └── input.dat
│   └── 3
│       └── input.dat
```

(continues on next page)

(continued from previous page)

```
├─ numConfs.txt
├─ two_alkanes-200.sdf
└─ two_alkanes.sdf
```

After running calculations from provided input files, collect results:

```
qp.process_results('two_alkanes-200.sdf')
```

4.2 Utilities

quanformer comes with a number of utility functions. These can be used as shown here:

```
import quanformer.utils as qu
qu.convert_extension('two_alkanes-200.sdf', 'two_alkanes-200.mol2')
```