
moltools Documentation

Release 1.0.0

Ignat Harczuk

May 11, 2015

1	Requirements	3
2	Installation	5
2.1	Linux	5
3	Modules	7
3.1	molecules.py	7
3.2	use_generator.py	7
3.3	use_calculator.py	7
3.4	template.py	7
3.5	read_dal.py	8
4	Testing	9
5	Bugs	11
	Python Module Index	13

moltools is a package of scripts and modules I use daily in my research as a Ph.D. student to ease setting up and analyzing calculations.

Contents:

Requirements

For bug-free performance, and full compatability, the latest source of the core dependencies are recommended.

The only hard-requirement is that python is the 2.7 version.

- numpy 1.9.1
- scipy 0.15.1
- matplotlib 1.4.3
- python-nose 1.3.4

Optional dependencies include:

- mpl_toolkits
- h5py 2.5.0
- mayavi2 <http://mayavi.sourceforge.net/>
- DALTON any stable version for Alpha properties. <http://www.daltonprogram.org/www/download.html>
- DALTON (Development version, master branch commit 51601c2 and forward is fine) for Beta properties.

Installation

Right now only installation on Linux is supported.

There is currently no plans to port this package to other systems.

2.1 Linux

1. Clone the repository `git@github.com:fishstamp82/dalton_tools.git`
2. set the PYTHONPATH environment variable to the target dalton_tools/src path.

3.1 molecules.py

3.1.1 Atom

3.1.2 Molecule

3.1.3 Water

3.1.4 Rotator

3.1.5 Property

3.1.6 Cluster

3.2 use_generator.py

3.3 use_calculator.py

3.4 template.py

`class template.Template`

This class holds data obtained by the LoProp transformation. Each template depends on 5 variables.

Variable	Choices	Type
Model:	TIP3P / OLAV	string
Method:	HF	string
Basis:	PVDZ/ANOPVDZ/ANOPVTZ	string
LoProp:	True/False	bool
Frequency:	Field ω	string

```
>>> temp = Template().get()
>>> print temp ("01", "charge")
0.0

>>> temp = Template().get( model = "TIP3P", method = "HF",
                           basis = "ANOPVDZ", dist = True, freq = "0.0" )
```

```
>>> print temp ("01", "charge")  
-0.678
```

3.5 read_dal.py

Testing

The code uses continuous testing with [Travis CI](#).

Unit-tests can be launched by the user by running

```
$ nosetests
```

Bugs

Bugs can appear in any shape, form or code.

If you find some bugs and want to fix them, it would be greatly appreciated!

Create a pull-request for the github [repository](#).

t

template, [7](#)

T

Template (class in template), [7](#)
template (module), [7](#)