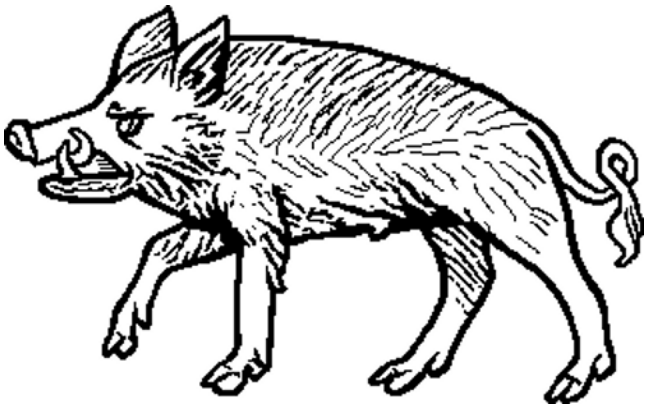
PyWordom User's Guide

w-version 0.23-rc1

July 2014



Developers: Michele Seeber

License: GPL

URL: `http://wordom.sf.net/`

Mantainer: mseeber@unimore.it

Description: pyWordom is a python module derived from the Wordom program, automatically generated with the SWIG tool.

Contents

Chapter 1

Introduction

The basic IO functions of Wordom have been wrapped with the SWIG[1] tool into a python[2] module.

1.1 Installation

Since the pywordom module is not pure python but has a compiled C core it must be recompiled, too. To recompile just go to the sources directory and type:

```
python setup.py build
```

Then, as root, run the command:

```
python setup.py install
```

This will install the wordom python module in system directories. The *import wordom* line in a python script will include the wordom module.

```
#!/bin/env python
import wordom as wrd
```

Chapter 2

Usage

The *wordom* python module allows simple python scripts to access data in the pdb, crd, dcd and xtc formats. Most of the available functionalities are provided through the *molecule*, *trajectory*, *coordinates* and *selection* objects:

2.1 Classes

2.1.1 Molecule

The Molecule class (wordom.Molecule) contains all the informations read from molecule files such as pdb and crd files. These data are further processed and a molecule→ chains→ segments→ residues→ atoms tree is built.

members

nato: number of atoms
nRes: number of residues
nSeg: number of segments
nChain; number of chains

methods

read : read molecule file into instance (mol1.read("mymol.pdb"))
write : write instance content to a file (mol1.write("output.pdb")

iterator

iterating on a molecule will return pointers to chain objects (see below).

2.1.2 Chain

The Chain class describes chains, a subdivision of a molecule.

2.1.3 Segment

The Segment class describes segments, a subdivision of a molecule and collection of residues, usually continuous. Segments can be seen as a subdivision of chains. Not all molecular file formats account for both chains and segments (*eg* crd files do not have chains).

2.1.4 Residue

The Residue class describes residues, usually an aminoacid.

2.1.5 Atom

The Atom class describes atoms, the smallest unit in a description of a molecule in the usual molecular mechanics workframe.

2.1.6 Selection

members

methods

2.1.7 Trajectory

members

methods

2.2 Functions

Create a new instance of the Molecule class:
mol1 = wrd.NewMolecule()
destroy a molecule instance:
wrd.DesMolecule(mol1)
create a selection instance:
sele1 = wrd.NewSele()
create a trajectory instance:
trj1 = wrd.NewTraj()
destroy a trajectory instance:
wrd.DesTrajtrj1)
create a trajectory header instance:
trh1 = wrd.NewTrjh()
create a coordinates set instance:
coor1 = wrd.NewCoor()

destroy a coordinates set instance:
wrd.DesCoor(coor1)

Functions and methods to play with these structure:

Accessing nn^{th} coordinate inside a coordinate set:
coor1.x(nn)
coor1.y(nn)
coor1.z(nn)

Setting nn^{th} coordinate inside a coordinate set:
coor1.setx(nn, xvalue)
coor1.sety(nn, yvalue)
coor1.setz(nn, zvalue)
coor1.setcoor(nn, xvalue, yvalue, zvalue)

Copying values from another coordinate set:
coor1.copycoor(coorset2)

Extract a subset of coordinates:
subset1 = coor.getselecoor(sele1)

-

Example:

```
import wordom as wrd

mymol = wrd.Molecule()
mymol.read( "./structure.pdb" )
mymol.write( "./copy.pdb" )

sele1 = mymol.select( "/CA" )
print sele1.nselatm
mymol2 = mymol.getselemol(sele1)
mymol2.write( "submol.pdb" )
```

Bibliography

[1] D. Beazley, *The Simple Wrapper and Interface Generator*. Available at: `http://www.swig.org`

[2] Guido van Rossum, *The Python Programming Language*. Available at: `http://www.python.org`