
MAGIC AUTOMATED GROMOS LIPID CONSTRUCTOR

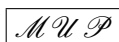
AN AUTOMATED SYSTEM TO PARAMETRISE THE WORLD
DOCUMENTATION

ALBANY, NEW ZEALAND, 2014

EDITED BY

IVAN WELSH

*Massey University
Albany*



2014

MASSEY UNIVERSITY PRESS

Contents

| | | |
|----------|-----------------------------------|----------|
| 1 | Introduction | 1 |
| 2 | Workflow | 2 |
| 3 | Code Documentation | 3 |
| 3.1 | buildlipid.py | 3 |
| 3.2 | config.py | 3 |
| 3.2.1 | Configuration Constants | 3 |
| | STRING_BREAK_CHARACTERS | 3 |
| | RANDOM_NAME_CHARS | 3 |
| | NAME_LENGTH | 4 |
| | NUM_TIERS | 4 |
| | ROOT_DIRECTORY | 4 |
| | SAVED_PICKLED_FILES | 4 |
| | ATOMIC_RADII | 4 |
| 3.2.2 | Global Storage | 5 |
| | MTB_PDB_ROOT_NAMES | 5 |
| | CHECKED_ATOMS | 5 |
| | LOADED_MOLECULES | 5 |
| 3.3 | connect.py | 6 |
| 3.4 | genmethods.py | 6 |
| 3.5 | ifp.py | 6 |
| 3.6 | magic.py | 6 |
| 3.6.1 | main() | 6 |
| 3.7 | molecule.py | 6 |
| 3.8 | rotations.py | 6 |

Chapter 1

Introduction

The Magic Automated GROMOS Lipid Constructor, or MAGIC, is a series of python scripts designed to provide a means to create parameters for any conceivable molecule. The original brief is to construct parameters for any given lipid molecule, but this ability will be extended to being able to parametrise any given molecule.

The working force behind MAGIC is the ATB, an automated topology builder produced and maintained by University of Queensland. ATB is provided with a PDB structure of a given molecule. Using this structure, a number of quantum mechanical calculations are performed and through matching those results to the existing parameters in the GROMOS forcefield, or generating new ones if a suitable one does not already exist, a parameter set for the molecule is generated.

It is from these parameter sets that MAGIC takes over. For various reasons, the ATB can only provide high-level, and thus reliable, results for molecules containing less than 40 atoms. Most biochemical molecules, including lipids, have many more than 40 atoms, leaving the results from the ATB to be unreliable. The aim of MAGIC is to take these ATB generated parameters and join them together in a sensible manner in order to obtain larger molecules with reliable results.

Chapter 2

Workflow

Chapter 3

Code Documentation

Details and usage of all functions present in MAGIC are given here. Each .py file is detailed in a separate section, with a file description followed by a sub-sectional breakdown of each function. In the case of classes being present, the classes are broken into sub-sections, with the functions being broken down into sub-sub-sections.

3.1 `buildlipid.py`

3.2 `config.py`

Contains various configuration options and globally accessible data storage points.

3.2.1 Configuration Constants

`config.STRING_BREAK_CHARACTERS`

A string of the opening characters for various types of molecule definition through strings. See section [ADDSECTIONHERE] for more information. Default string is {(<.

`config.RANDOM_NAME_CHARS`

A string of all the possible characters to be used when generating new file names for the joint molecules. Default string is all upper case alphanumeric characters.

`config.NAME_LENGTH`

The number of characters in the file names of the fragment molecules. Default is 4.

`config.NUM_TIERS`

The number of tiers used in the OVL files to determine overlapping rules. Default is 5. Caution is advised when altering this constant as OVL files will need to be updated and, more importantly, the use of the default 5 is fairly hard coded into MAGIC.

`config.ROOT_DIRECTORY`

The directory from which the magic.py script is run originally. It is set to the current directory using the `os.getcwd()` function. Work directory does not change during a MAGIC process.

`config.SAVED_PICKLED_FILES`

The list of pickled files present in the `config.ROOT_DIRECTORY/PickledFiles/` directory. It is generated each time the constant is called using the `os.listdir(...)` function.

`config.ATOMIC_RADII`

A dictionary of atomic radii used to determine if atoms will overlap in a joint molecule and so if a rotation is needed. Values are measured in angstroms (Å) and sourced from the Wikipedia article. [Change to data book](#). The values present by default are given in Table 3.1 below.

Table 3.1: Atoms and their atomic radius store by default in MAGIC. Radius is in Å.

| Atom Type | Radius |
|------------|--------|
| Hydrogen | 0.53 |
| Carbon | 0.67 |
| Nitrogen | 0.56 |
| Oxygen | 0.48 |
| Fluorine | 0.42 |
| Sodium | 1.90 |
| Magnesium | 1.45 |
| Silicon | 1.11 |
| Phosphorus | 0.98 |
| Sulfur | 0.88 |
| Chlorine | 0.79 |
| Argon | 0.71 |
| Calcium | 1.94 |
| Iron | 1.56 |
| Copper | 1.45 |
| Zinc | 1.42 |
| Bromine | 0.94 |

3.2.2 Global Storage

`config.MTB_PDB_ROOT_NAMES`

A list of the unique root names of molecules present in the various data directories. It is generated by the initial call to `magic`, and added to as files are saved to the output directory.

`config.CHECKED_ATOMS`

A list of the atom indices that have been checked by the `genmethods.graphSearch()` function during the joining process. It is wiped clean at the start of each initial call to `graphSearch()`. Existence is so that infinite loop recursion does not occur.

`config.LOADED_MOLECULES`

A dictionary of all the molecules loaded into memory. Molecules are keyed by name. Each molecule, either created or loaded, is added to the dictionary. If a given key already exists in the dictionary, the molecule isn't loaded again/the random new name is regenerated.

3.3 connect.py

3.4 genmethods.py

3.5 ifp.py

3.6 magic.py

Wrapper script used to call build lipid.MAGIC(...) execution function with the provided command line arguments.

3.6.1 magic.main(argv)

Input *argv*: command line arguments. Usage is:

```
magic.py -s <StringToParse> -i <IFPFile>
```

Functionality Assigns default values to the ifpfile and pString parameters. Generates the list of unique molecule names from all the files in the various data directories, assigning them to the config.MTB_PDB_ROOT_NAMES global data storage point. Parses the command line arguments, setting ifpfile and pString to their requested values. Finally calls buildlipid.MAGIC(ifpfile, pString) to begin the connection process.

3.7 molecule.py

3.8 rotations.py